A Hybrid Algorithm with Modified Inver-Over Operator and Genetic Algorithm Search for Traveling Salesman Problem

Dharm Raj Singh, Manoj Kumar Singh and Tarkeshwar Singh

Abstract In this article, we develop a novel hybrid approach to solve the traveling salesman problem (TSP). In this approach, we first initialize suboptimal solution using Nearest Neighbor (NN) tour construction method, followed modified Inver-over operator and then proposed crossover with 2-opt mutation applied to improve for optimal solution. We use 14 TSP data sets from TSPLIB to evaluate the performance of proposed hybrid method. The proposed hybrid method gives better results in terms of best and average error. In experimental results of the tests we show that the proposed hybrid method is superior to available algorithm in literature.

Keywords Traveling salesman problem · Basic inver-over operator · Modified inver-over operator · 2-opt mutation · Crossover operator

1 Introduction

1.1 Traveling Salesman Problem

Combinatorial optimization problem contains a broad study of the traveling salesman problem (TSP) [1]. Given a set of n cities and distance between each pair of cities, the traveling salesman visit all cities only once and return back to starting city with minimum distances. This type problem is modeled in graph theory with

D.R. Singh (✉) · M.K. Singh
DST-Centre for Interdisciplinary Mathematical Sciences (DST-CIMS), Banaras Hindu University, Varanasi 221005, India
e-mail: dharmrajsingh67@yahoo.com

M.K. Singh
e-mail: manoj.dstcims@bhu.ac.in

T. Singh
Department of Mathematics, Birla Institute of Technology and Science Pilani, K K Birla Goa Campus, Zuarinagar 403726, Goa, India
e-mail: tksingh@goa.bits-pilani.ac.in

© Springer Science+Business Media Singapore 2016

R.K. Choudhary et al. (eds.), *Advanced Computing and Communication Technologies*, Advances in Intelligent Systems and Computing 452,
DOI 10.1007/978-981-10-1023-1_14

help of vertices and edges. Vertices represent cities and edges represent roads between the two cities and the weight of an edge represents distance between two adjacent cities. Thus, we have a weighted graph $G = (V, E, w)$, where $w: E \rightarrow Z$ and Z is a set of nonnegative integer. A closed $C = (u = u_0, u_1, u_2, \ldots, u_n = u)$ tour in which all the vertices are distinct which is known as Hamiltonian cycle. Finding Hamiltonian cycle with minimum travel cost $w(C) = \sum w(e_i)$, where summation is taken over all edges in C in the weighted graph is the desired solution. The matrix representation of weighted graph is known as cost matrix which is denoted by $C = (c_{ij})$, $i, j = 1, 2, \ldots, n$. An entry of cost matrix c_{ij} represents the cost between ith vertex to jth vertex. The total cost of Hamiltonian cycle is the sum of cost of each edge in Hamiltonian cycle. In general; this problem is NP-hard.

The Euclidean distance (cost) between two adjacent vertices is calculated as fallow:

$$c_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}, \tag{1}$$

where (x_i, y_i) and (x_j, y_j) are coordinates of two adjacent vertices.

1.2 Literature Review

According to Arora [2], TSP is NP-hard combinational optimization problem. The most commonly used heuristic and meta-heuristic algorithms such as Greedy algorithms, 2-opt algorithms, Simulated Annealing (cf., [3, 4]), Tabu Search, Ant Colony Optimization [5, 6], Genetic Algorithms (cf., [7–10]), Neural Networks [11], Weed optimization [12] and Memetic Algorithm (MA) (cf., [11, 13]) etc.

TSP algorithms are categorized into two classes such as exact and approximate algorithms. The exact algorithm is always used for optimal solution. Cutting plane or branch and bound method is one of the most effective exact algorithms, which have been solved large TSP instances (see [14]). But the exact algorithms have taken high time complexity. However, an approximate algorithm is used for the solution of TSP instances in reducing computation, while the obtained solution using approximate algorithm is nearest to the optimal solution. In [15], we have used nearest neighbor tour construction method for generating a tour and tour improvement methods used to get better quality of tour by applying the various exchanges. The 2-opt optimal heuristic is generally considered as one of the most effective methods, which is generating approximate optimal solutions for the traveling salesman problem (TSP). Guo Tao proposed an algorithm based on the Inver-over operator and memetic algorithm with improved Inver-over operator, which is presented in [9, 16], respectively.

The remaining part of the paper is constructed as follows. Section 2 describes the overview of Basic Inver-over operator. Proposed hybrid method is described in Sect. 3. Experimental results are presented in Sect. 4. In the last Sect. 5 provides the concluding part of the article.

$$\underset{c=6}{(2,1,4,6,8,7,3,5)} \xrightarrow[\mathit{inverse}(8,7,3)]{c'=3} \underset{c=3}{(2,1,4,6,3,7,8,5)} \xrightarrow[\mathit{inverse}(2,1,4,6)]{c'=2} \underset{c=2}{(6,4,1,2,3,7,8,5)} \xrightarrow{c'=3} \mathit{end}$$

Fig. 1 Single iteration of basic inver-over operator

2 Basic Inver-Over Operator

The Inver-over operator [9] performs both operation crossover and mutation. In this method, first we randomly select a city from the individuals, after then we generate a random number (rnd). If the random number (rnd) < certain threshold value (prd) then select the second city from the rest of individuals for inversion. In this case, inversion performs mutation. If the random number (rnd) > prd then randomly select another new individual from the population. After then select the second city just after to the selected first city from the new individual. In this case, the inversion operation performs crossover. The whole procedure of Basic Inver-over operator is given in [16, 17].

Let us assume that starts with selected chromosome S' is (2, 1, 4, 6, 8, 7, 3, 5). Figure 1 shows a single iteration of this operator and the inversion function which is responsible for the inversion.

3 Proposed Hybrid Method

Proposed hybrid method provides the approximate solution which depend heuristics with exploration and exploitation. The overall procedures of our algorithm that combines the nearest-neighbor tour construction, modified Inver-over operator and genetic algorithms with 2-opt mutation heuristics.

In the proposed hybrid method, the first step in algorithm is using Nearest Neighbor tour construction heuristic (see, [18]) to generate initial population of population size. In the second step, we find fitness value of each chromosome in population. In third step, we select best chromosome S' from population. In the fourth step, we applied the modified Inver-over operator on the selected chromosome S'. After the fourth step, we generate new chromosome. If cost of new generated chromosome is less than old chromosome then replace the old chromosome by new chromosome. In the fifth step randomly select two chromosomes from population and then we applied the proposed crossover operator on selected chromosomes with crossover probability rate (pc). In the last step, we applied 2-opt optimal mutation operator on selected two parents or new individuals that generated after crossover, with mutation probability rate (pm) and update the population. The whole process repeated until termination condition is satisfied. Figure 2 describes a detail description of the proposed hybrid algorithm.

```
Using Nearest Neighbor tour construction heuristic to generate initial population of size $P$;
while (terminate condition are not satisfied) do
  Find the fitness value $f$ for each chromosome in $P$;
  Select best chromosome $S^{'} = S_i$ from $P$;
  Randomly select a city $c$ from $S^{'}$;
  Generate a random number (rnd1) between (0, 1);
  while (true)
    Generate a random number (Rnd2) between (0, 1);
     if (Rnd2< prd) then
       Apply 2-opt optimal mutation to best selected chromosome $S^{'}$;
       Randomly select city $c$ from $S^{'}$;
       Continue;
    else if (Rand2<prd + pcs)
       Randomly select city $c^{'}$ from a set of kα- nearest neighbors of the city $c$;
    else
       Randomly select a new chromosome of $P$;
      Assign to city $c^{'}$ the subsequently to the city $c$ in the newly selected chromosome;
    end if
    if ( city $c^{'}$ is  the previous or  next  city of $c$ in $S^{'}$ ) then
      Break;
    else
      Apply Inverse operation from city $c$ to the city $c^{'}$ in $S^{'}$ (excluding $c$);
    end
    if (rnd1< puc (gn)) then
      Remain city $c$ unchanged;
    else
      $c = c^{'}$
   end
  end while
    if (cost of new generated chromosome is lesser than old chromosome) then
       Replace the old chromosome by new chromosome;
   end
   Random select two parents for crossover ($S_1$, $S_2 \in P$ );
   Apply proposed crossover on selected parents with probability rate pc;
   Apply 2-opt optimal mutation on selected two parents or new child with probability rate pm;
   Update population;
end while
```

Fig. 2 Proposed hybrid algorithm with modified inver-over operator

3.1 Modified Inver-Over Operator

Figure 1 depicts Basic Inver-over operator process, in which first inversion operator to city (8, 7, 3), then the new age (6, 3) is added to the current solution and again apply inversion operator to city (2, 1, 4, 6), then the edge (3, 2) is added. After the second inversion edge (6, 3) is deleted from solution. Therefore, Basic Inver-over operator ignores direction of path. The direction of path involved in Modified Inver-over operator. In this case, the newly added edge, by applying first inversion operator will not be removed till the next inversion. Figure 3 shows Modified

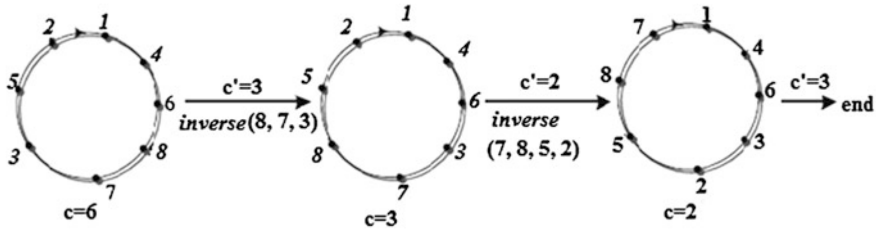

Fig. 3 A single iteration of modified inver-over operator (involving direction of path)

Inver-over operator that considers direction of path. Therefore, effect of each inversion will be sure.

It has verified by experimental result. The speed of convergence will be increased by receiving instructor of candidate set to select the city c'. In Modified Inver-over operator, randomly we select city c' from a set of α-nearest neighbors of the city c with certain threshold value [19].

Moreover, if the city c is unchanged with city c' then the population diversity will be kept until the possibility of traffic at local optimum will be reduced. Then, both hybrid methodologies have developed to kept population changes with effectiveness. If the random number (rnd1) < puc then the city c unchanged, otherwise city c replace by c'. In the algorithm, the possible adaptive value of puc increases with iteration of the algorithm, the city c will be unchanged with the higher possibility in a later period of the algorithm, while the population changes will be managed suitably. The relationship between iteration of algorithm and puc is given as follows:

$$puc(gn) = \exp\left(\log\left(\frac{puc\_\max - puc\_\min}{N}\right)\right) \times gn, \tag{2}$$

where, *puc* denotes the possibility unchanged city c, *puc_min* and *puc_max* are minimum and maximum possibilities, gn indicates iteration number while N is total number of iteration.

3.2 2-Opt Mutation

Croes [20] proposed the 2-Opt method. Basically, 2-opt mutation deletes two edges from path, and add two new edges that are not in solution in such way that the cost of new path is lesser. The process is continued till no further improvements are possible [19]. The resulting path is referred to as a 2-opt optimal (see [21]). For example, inverting the subsequence $(x_{i+1}, \ldots, x_j)$ of sub path $(x_i, x_{i+1}, \ldots, x_j, x_{j+1})$ is replaced with $(x_i, x_j, \ldots, x_{i+1}, x_{j+1})$.

Finally, performing the 2-exchange mutation, the resulting cost is expressed as follow:

$$\Delta_{ij} = c(x_i, x_j) + c(x_{i+1}, x_{j+1}) - c(x_i, x_{i+1}) - c(x_j, x_{j+1}). \qquad (3)$$

The resulting cost is obtained by iteratively applying 2–exchange mutation till impossible move yields a negative Δ value.

3.3 Proposed Crossover Operator

In the proposed crossover, the first city of chromosome s_1 is copied to first position in the child c_1 and first city of chromosome s_2 is copied to first position in the child c_2. The remaining cities changed accordingly in Fig. 5. For example, the first city of parent s_1, s_2 are copied at the first position in the child c_1 and c_2 respectively in Fig. 4, remaining positions 2, 3, 4, 5, 6, 7, 8 and 9 cities are swapped as procedure given in Fig. 5.

Fig. 4 Proposed crossover

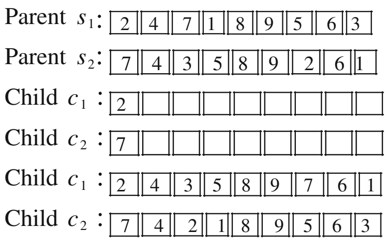
Parent s_1: 2 4 7 1 8 9 5 6 3
Parent s_2: 7 4 3 5 8 9 2 6 1
Child c_1 : 2
Child c_2 : 7
Child c_1 : 2 4 3 5 8 9 7 6 1
Child c_2 : 7 4 2 1 8 9 5 6 3

Fig. 5 Algorithm for proposed crossover

```
for i = 1: n
    for j = 2:n
      if ($s_2(i) == s_1(j)$)
        $c_1(j) = s_2(j)$;
      end
      if ($s_1(i) == s_2(j)$)
        $c_2(j) = s_1(j)$;
      end
    end
end
for i = 2:n
    if ($c_1(1) == c_1(i)$)
      $c_1(i) = s_2(1)$;
    end
    if ($c_2(1) == c_2(i)$)
      $c_2(i) = s_1(1)$;
    end
end
```

4 Experimental Results

4.1 Experimental Setup

For evaluating the performance of experimental results, Intel (R) Core (i5) 3.20 GHz processor, 2GB RAM on MATLAB is used. In this experiment, we used 14 different TSP benchmark instances taken from the TSPLIB. The value of parameters used in experiment are population size (P) = 40, minimum probability of city (c) unchanged is *puc_min* = 0.2, maximum probability of city (c) unchanged is *puc_max* = 0.5 and 5α-nearest neighbour. The value of prd = 0.02, pcs = 0.05, pc = 0.8, pm = 0.25 with N = 2000 iteration.

The efficiency of proposed hybrid method is based on Percentage Best Error (% Best Err.) and Percentage Average Error (% Ave. Err.). The Percentage Best Error and Percentage Average Error are given as follows:

$$\%\text{ Best Err.} = \frac{(\text{best solution from } n-\text{trail}) - (\text{best known solution from TSPLIB})}{\text{best known solution from TSPLIB}} * 100$$

$$\%\text{ Ave. Err.} = \frac{(\text{average solution from } n-\text{trail}) - (\text{best known solution from TSPLIB})}{\text{best known solution from TSPLIB}} * 100$$

4.2 Experimental Results and Analysis

In this section, we compared the proposed hybrid algorithm with the recent algorithms as presented in [7, 15, 16]. We performed n = 10 trails for our present method. The comparative results are presented in Table 1, the best results of the method for particular instances are in bold. From last row of the table, it is clear that our proposed method gives better result for all instances with respect to all parameters, except berlin52 for % Best Err. and % Ave. Err., kroA100 for % Best Err. and % Ave. Err., pr144 for % Ave. Err., ch150 for % Best Err. and % Ave. Err., pr152 for % Best, rat195 Best Err. and % Ave. Err. and ts225 for % Best Err. Moreover, obtained result by proposed method is better for TSP instances pr144 and kroB150, the best known solutions 58,537 and 26,130 reported by TSPLIB are replaced by the new best obtained solutions 58535.2 and 26127.36.

The last row of Table 1 has shown the average performance of each method over the data set. Figure 6 indicates % Ave. Err. with respect to instances size. From Fig. 6, we conclude that the proposed hybrid method is best among Inver-over and LK MA without local search, NN+2-opt mutation and GSTM algorithm for the traveling salesman problem.

Table 1 Performance comparison of different algorithm

Instance	Modified inver-over operator with genetic algorithm			Inver-over and LK MA (remove local search)			NN+2-opt mutation			GSTM algorithm		
	Best Err. (%)	Ave. Err. (%)	Ave. time(s)	Best Err. (%)	Ave. Err. (%)	Ave. time (s)	Best Err. (%)	Ave. Err. (%)	Ave. time (s)	Best Err. (%)	Ave. Err. (%)	Ave. time (s)
berlin52	0.0314	0.0314	1.2775	**0.0000**	**0.0000**	**0.4867**	0.0318	0.0318	3.042	**0.0000**	**0.0000**	0.836
kroA100	0.0162	0.0204	2.9336	**0.0000**	**0.0000**	**0.6193**	0.0141	0.1142	7.394	**0.0000**	1.1836	6.987
pr144	**−0.003**	0.0439	3.9659	0.0564	0.1350	**0.6878**	**−0.003**	**0.0174**	14.780	0.0000	1.0809	13.598
ch150	0.2778	0.6062	5.1288	**0.0000**	**0.3585**	**0.8566**	0.7077	1.2467	13.213	0.4596	0.6357	11.240
kroB150	**−0.0101**	**0.6271**	5.7666	0.0421	0.6456	**0.7829**	0.0421	0.9885	14.148	0.9644	1.7616	11.684
pr152	0.0022	**0.0769**	5.8002	**0.0000**	0.1259	**0.7144**	0.1886	0.4072	18.364	0.7695	1.6202	7.937
rat195	0.6518	0.8275	6.9373	**0.4305**	**0.6586**	**0.8503**	1.2613	1.9513	20.248	0.6027	1.8425	15.050
d198	**0.2050**	**0.4456**	9.3652	0.3359	0.6800	**0.9391**	0.5830	0.9157	22.302	0.3866	1.2193	12.096
kroA200	**0.0889**	**0.4293**	7.9241	0.4052	0.5816	**0.9047**	0.5925	1.0440	21.719	0.8683	1.5432	13.292
ts225	0.0023	**0.0209**	8.7002	**0.0000**	0.4850	**0.9815**	0.0055	0.3040	37.201	0.2527	0.4994	11.559
pr226	**0.0016**	**0.3257**	8.9554	0.1344	0.4270	**0.9206**	0.0535	0.3831	31.887	0.7242	1.5287	13.843
pr299	**0.2103**	**0.6718**	15.9916	0.6661	2.3401	**1.1949**	1.2139	2.1958	51.794	1.2326	2.9169	17.424
lin318	**0.5389**	**0.8847**	20.8173	1.3610	2.3058	**1.2731**	1.0027	2.2865	77.049	0.9827	3.3099	14.643
pcb442	**0.9038**	**1.3331**	37.5992	1.4494	2.1127	**1.6723**	1.8177	2.9822	122.998	2.0501	2.7758	19.132
Average	**0.2084**	**0.4532**	10.0831	0.3486	0.7754	**0.9203**	0.5368	1.0621	31.637	4.0423	1.5655	12.094

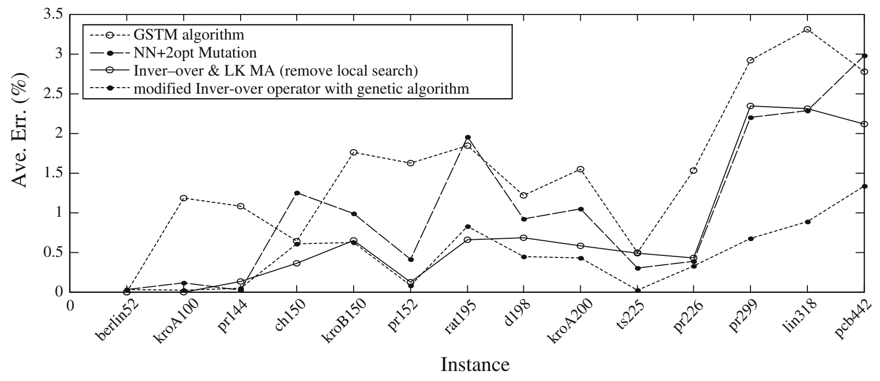

Fig. 6 % Ave. Err. (over 10 runs) for the 14 TSPLIB instances ordered by size

Our proposed method gets speed up the convergence of optimal solution because in this hybrid proposed method. Firstly, we generate initial global solution by using NN tour construction algorithm, then modified Inver-over operator is being used, after then proposed crossover with a powerful local improvement method (2-opt mutation) that refines the solution for global optimality.

5 Conclusion

In this article, we have discussed a novel proposed hybrid method for the Euclidean traveling salesman problem. It is combination of NN tour construction, Modified Improved Inver-over operator and genetic algorithm. In proposed method we considered the direction of path, randomly select a city from a set of α-nearest neighbors and proposed crossover with a powerful local improvement method (2-opt mutation). The combination affects the quality of solution. Therefore, our proposed hybrid method gives better performance than Greedy Sub Tour Mutation (GSTM), NN+2-opt mutation and Inver-over and LK MA (remove local search). Thus, the proposed hybrid method i.e. Modified Inver-over operator with genetic algorithm remarkably increases the performance of genetic algorithm in TSP solutions.

References

1. Lawler, E.L.: The traveling salesman problem: a guided tour of combinatorial optimization. Wiley-Interscience Series Discrete Mathematics (1985)
2. Arora, S.: Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. J. ACM (JACM) **45**(5), 753–782 (1998)

3. Chen, Y., Zhang, P.: Optimized annealing of traveling salesman problem from the nth-nearest-neighbor distribution. Physica A **371**(2), 627–632 (2006)
4. Jeong, C.S., Kim, M.H.: Fast parallel simulated annealing for traveling salesman problem on SIMD machines with linear interconnections. Parallel Comput. **17**(2), 221–228 (1991)
5. Deng, W., Chen, R., He, B., Liu, Y., Yin, L., Guo, J.: A novel two-stage hybrid swarm intelligence optimization algorithm and application. Soft. Comput. **16**(10), 1707–1722 (2012)
6. Yun, H.Y., Jeong, S.J., Kim, K.S.: Advanced harmony search with ant colony optimization for solving the traveling salesman problem. J. Appl. Math. (2013)
7. Albayrak, M., Allahverdi, N.: Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. Expert Syst. Appl. **38**(3), 1313–1320 (2011)
8. Louis, S.J., Li, G.: Case injected genetic algorithms for traveling salesman problems. Inf. Sci. **122**(2), 201–225 (2000)
9. Tao, G., Michalewicz, Z.: Inver-over operator for the TSP. In: Parallel Problem Solving from Nature—PPSN, pp. 803–812. Springer, Berlin (1998)
10. Ahmed, Z.H.: An experimental study of a hybrid genetic algorithm for the maximum traveling salesman problem. Math. Sci. **7**(1), 1–7 (2013)
11. Créput, J.C., Koukam, A.: A memetic neural network for the Euclidean traveling salesman problem. Neurocomputing **72**(4), 1250–1264 (2009)
12. Zhou, Y., Luo, Q., Chen, H., He, A., Wu, J.: A discrete invasive weed optimization algorithm for solving traveling salesman problem. Neurocomputing **151**, 1227–1236 (2015)
13. Merz, P., Freisleben, B.: Memetic algorithms for the traveling salesman problem. Complex Syst. **13**(4), 297–346 (2001)
14. Laporte, G.: La petite et la grande histoire du probleme du voyageur de commerce (2006)
15. Singh, D.R., Singh, M.K., Singh, T.: A hybrid heuristic algorithm for the Euclidean traveling salesman problem. In: 2015 International Conference on Computing, Communication and Automation (ICCCA), pp. 773–778. IEEE (2015)
16. Wang, Y.T., Li, J.Q., Gao, K.Z., Pan, Q.K.: Memetic algorithm based on improved inver–over operator and Lin-Kernighan local search for the Euclidean traveling salesman problem. Comput. Math Appl. **62**(7), 2743–2754 (2011)
17. Wang, Y., Sun, J., Li, J., Gao, K.: A modified inver-over operator for the traveling salesman problem. In: Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, pp. 17–23. Springer, Berlin (2012)
18. Johnson, D.S., McGeoch, L.A.: The traveling salesman problem: A case study in local optimization. Local Search Comb. Optim. **1**, 215–310 (1997)
19. Helsgaun, K.: An effective implementation of the Lin-Kernighan traveling salesman heuristic. Eur. J. Oper. Res. **126**(1), 106–130 (2000)
20. Croes, G.A.: A method for solving traveling-salesman problems. Oper. Res. **6**(6), 791–812 (1958)
21. Gutin, G., Punnen, A.P. (eds.): The traveling salesman problem and its variations, vol. 12. Springer Science & Business Media (2002)