The modification of hybrid method of ant colony optimization, particle swarm optimization and 3-OPT algorithm in traveling salesman problem

Hertono. G.F.1, Ubadah1, Handari, B.D.1
1 Department of Mathematics, Universitas Indonesia, Depok, 16424, Indonesia

gatot-fl@sci.ui.ac.id, ubadah@sci.ui.ac.id, bevina@sci.ui.ac.id

Abstract. The traveling salesman problem (TSP) is a famous problem in finding the shortest tour to visit every vertex exactly once, except the first vertex, given a set of vertices. This paper discusses three modification methods to solve TSP by combining Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and 3-OPT Algorithm. The ACO is used to find the solution of TSP, in which the PSO is implemented to find the best value of parameters $\alpha$ and $\beta$ that are used in ACO. In order to reduce the total of tour length from the feasible solution obtained by ACO, then the 3-Opt will be used. In the first modification, the 3-Opt is used to reduce the total tour length from the feasible solutions obtained at each iteration, meanwhile, as the second modification, 3-Opt is used to reduce the total tour length from the entire solution obtained at every iteration. In the third modification, 3-Opt is used to reduce the total tour length from different solutions obtained at each iteration. Results are tested using 6 benchmark problems taken from TSPLIB by calculating the relative error to the best known solution as well as the running time. Among those modifications, only the second and third modification give satisfactory results except the second one needs more execution time compare to the third modifications.

1. Introduction
Traveling salesman problem (TSP) is known as a problem to determine the salesman’s route visiting some places (cities) that start and end in a certain city. Furthermore, a condition is applied that every city must be visited once (except the initial city). The chosen route must have the minimal distance (Gupta, 2013).

Some methods have been developed to solve the TSP, and one of them is the hybrid method proposed by Mahi et al. (2015), that is based on Ant Colony Optimization (ACO) (Dorigo, 1996, Deneubourg and Goss, 1989), Particle Swarm Optimization (PSO) (Kennedy, 1995, Reynold, 1987, Heppner and Grenander, 1990) and 3-Opt Algorithm.

In this paper, we modified the method proposed by Mahi et al with three different strategies. The difference among those modifications is lay on the usage of 3-Opt algorithm during the process. In the first modification method, the 3-Opt algorithm is used to reduce the total distance of the feasible solution that is obtained from each iteration. Meanwhile, in the second modification, the 3-Opt algorithm is used to reduce the total distance of all solutions that are obtained in each iteration. In the third modification, the 3-Opt algorithm is used to reduce the total distance of different solutions that are obtained in each iteration. In order to see the performance of those modifications, 6 benchmark
problems from TSPLIB are used. The relative error of the mean solution to the best known solution will be compared as well as the related running time.

2. Preliminaries
In [6], Mahi’s et al proposed the hybrid method that is described in the scheme as shown in Figure 1.

![Figure 1. Mahi’s Hybrid Method.](image)

The hybrid method starts by implementing the ACO steps, which consist of: the initial agent distribution process, pheromone initialization and heuristic value calculation. In order to create solutions, the second step of ACO will calculate a probability by using the parameter values of $\alpha$ and $\beta$. An agent will visit the next vertex based on this probability value. This step will be repeated until every agent has visited every vertex. The feasible solution, which is the route with the minimum total distance will be determined from all agent’s route. The third step is used to update the pheromone of the feasible solution. The ACO process will be ended by termination criteria. If the criteria do not meet the termination condition, the process will be continued by redistributing agents to the vertices. In this phase, the PSO will be used to determine the new parameter values $\alpha$ and $\beta$ that will be used in the next ACO processes. The whole process will be repeated until the termination criteria is met and the candidate solution can be obtained from all feasible solutions. This candidate solutions will be optimized by 3-Opt algorithm to obtain the best solution.

3. Discussion
In Mahi’s, 3-Opt algorithm is applied to the candidate solution. In our first modification method, 3-Opt algorithm is applied to the feasible solution, as described in Figure 2. The differences can be seen after every agent has visited every vertex (visitation complete). The feasible solution will be chosen from all solutions generated by all agents. The 3-Opt algorithm will be applied to these feasible solutions. The next step is the third ACO process that updates the pheromone and followed by the same steps as Mahi’s until the process satisfies the terminal criteria. When the criteria is fulfilled, the best of the feasible solution will be the best solution of the first modification hybrid method.
In the second hybrid modification method, we implement the 3-Opt algorithm to all agent’s solution in each iteration as shown in Figure 3. In the first modification method, the 3-Opt algorithm is only applied to the feasible solution in each ACO iteration, while in the second modification method, the 3-Opt algorithm is applied to all agent’s solution in each iteration. After all solutions have been optimized by 3-Opt, the solution with the minimum total distance is selected as a feasible solution.
The next step is the third step of ACO (update the pheromone) and testing criteria. The process will continue until the process satisfies the terminal criteria and the best solution is obtained.

The third hybrid modification method is described as shown in Figure 4:

In this modification, we only implement the 3-Opt algorithm to solutions which are different from solutions obtained in the previous iterations. The idea of this strategy is since some agents may produce the same solutions which represent the same route even though they have different vertex order. The checking step can be done by setting one vertex as the initial vertex and put the other vertices in a corresponding order. Hence, the route of each agent will be obtained with the same initial vertex. When one route has been applied the 3-Opt algorithm, and if there is other route with the same vertex’s order exists then the corresponding route will be skipped from the 3-Opt algorithm. In other words, the 3-Opt algorithm only applied to the routes with the different vertex’s order. Hence, this strategy will reduce the running time. After all different routes have been optimized, the solution is chosen to be the feasible solution and the process continue with the third ACO process. The whole process will be repeated until the termination criteria is held and the best solution is obtained.

The implementation uses 6 benchmarks problems from TSPLIB (Reinelt, 1991): Eil51, Berlin52, St70, Eil76, Rat99, and kroA200. Every problem is tested in 10 runs with 50 iterations each. Mean solution (MS) is calculated by using the following equation:

\[ MS = \frac{\sum_{i=1}^{n} D_i}{n}, \]  

where \( D_i \) is the solution of problem \( i, i = 1, 2, ..., n \). The percentage of error relative (ER) is used to determine how good the method solving the TSP and calculated by equation:

\[ ER = \frac{MS - BKS}{BKS} \times 100, \]  

**Figure 4. Third Modification of Hybrid Method.**
where BKS is the best known solution.

In the implementation, we also use the parameters value as suggested in the previous papers, as follows: \( \rho = 0.1, \ c_1 = 2, \ c_2 = 2 \) (Mahi, 2015) and \( Q = 100 \) as the best constant value for the ACO method (Dorigo, 1996). We also use \( 0 \leq \alpha \leq 2 \) and \( 0 \leq \beta \leq 2 \) as suggested by Mahi (2015) to get the best result for the ACO method. Mahi (2015) also suggests using 10 agents when we use the hybrid method using ACO, PSO and 3-Opt algorithm.

Table 1. The results of the first hybrid modification method.

| Problems | BKS | Best Solution | Worst Solution | Mean Solution | ER (%) | Time (second) |
|----------|-----|---------------|----------------|---------------|--------|---------------|
| Eil51    | 426 | 437           | 514            | 480.8         | 12.86  | 325.55        |
| Berlin52 | 7542| 7930          | 8832           | 8351.6        | 10.73  | 328.27        |
| St70     | 675 | 683           | 777            | 732.8         | 8.56   | 427.28        |
| Eil76    | 538 | 605           | 663            | 634.4         | 17.92  | 488.64        |
| Rat99    | 1211| 1311          | 1499           | 1439.6        | 18.88  | 818.51        |
| KroA200  | 29368| 29957         | 36761          | 34039.4       | 15.91  | 1042.13       |

Table 1 shows that the first hybrid modification method does not give good results for all problems. It shows the corresponding best solution is larger than the best known solution (BKS) in Mahi (2015). Furthermore, the corresponding ER is also greater than 8%. The results also show high running time.

Table 2. The results of the second hybrid modification method.

| Problems | BKS | Best Solution | Worst Solution | Mean Solution | ER (%) | Time (second) |
|----------|-----|---------------|----------------|---------------|--------|---------------|
| Eil51    | 426 | 426           | 426            | 426           | 0.00   | 658.43        |
| Berlin52 | 7542| 7542          | 7542           | 7542          | 0.00   | 547.11        |
| St70     | 675 | 675           | 675            | 675           | 0.00   | 1138.9        |
| Eil76    | 538 | 538           | 538            | 538           | 0.00   | 1413.8        |
| Rat99    | 1211| 1211          | 1211           | 1211          | 0.00   | 3941.3        |
| KroA200  | 29368| 29368         | 29368          | 29368         | 0.00   | 7261.4        |

The second hybrid modification method gives better results for every problem used. Table 2 shows that the worst solutions exactly match the BKS for every problem. However, the running time that represents the mean time needed to solve problem in one trial is greater than the one of the corresponding problems in the first hybrid modification method. The hypothesis is that it causes too many executions of 3-Opt algorithm in the method. This 3-Opt algorithm works for every solution generated by every agent in each iteration. Since there are 10 agents and each runs in 50 iterations there are 500 executions 3-Opt algorithm.

In Table 3, the third modification method relatively gives a better result than the previous methods. The difference of the relative error is not more than 0.1% compared to the second one. Fortunately, the mean time needed to execute one trial is also less than the corresponding time that needed to solve one trial of the second modification problems, eq. problem Rat99 needs 3941.3 seconds in second hybrid modification and it needs 1103.3 seconds in the third modification.
Table 3. The result of the third hybrid modification method.

| Problems | BKS | Best Solution | Worst Solution | Mean Solution | ER(%) | Time (second) |
|----------|-----|---------------|----------------|---------------|-------|---------------|
| Eil51    | 426 | 426           | 428            | 426.4         | 0.09  | 408.82        |
| Berlin52 | 7542| 7542          | 7542           | 7542          | 0.00  | 450.72        |
| St70     | 675 | 675           | 676            | 675.2         | 0.03  | 526.51        |
| Eil76    | 538 | 538           | 539            | 538.2         | 0.04  | 570.46        |
| Rat99    | 1211| 1211          | 1212           | 1211.8        | 0.07  | 1103.3        |
| KroA200  | 29368| 29368         | 29648          | 29424         | 0.19  | 1405.71       |

In Table 4, we try with a larger number of iterations with two TSPLIB problems, Eil51 (problem with 51 vertices) and KroA200 (problem with 200 vertices). With 5 runs and 100 iterations each, the first modification of hybrid method has a significant reduction in percentage of relative error by increasing the number of iterations.

Table 4. The Comparison Results of Three Methods on Eil51 and KroA200 with 50 and 100 iterations.

| Problems | Iterations | ER(%)     | Time(Second) |
|----------|------------|-----------|--------------|
|          | 1     | 2   | 3     | 1      | 2     | 3     |
| Eil51    | 50    | 12,28| 0.00 | 0.13 | 304.04| 655.01| 408.00|
|          | 100   | 7,04 | 0.00 | 0.00 | 533.54| 1004.98| 690.95|
| KroA200  | 50    | 20,17| 0.00 | 0.22 | 1338.13| 7371.02| 1801.71|
|          | 100   | 12,01| 0.00 | 0.09 | 1985.14| 13292.01| 2915.16|

4. Conclusion
Based on six TSPLIB problems used with 10 agents, with 10 runs in 50 iterations each, we can see that the third hybrid modification gives the best result in terms of relative error and running time. The second modification gives a good relative error but needs more time for execution. The performance of the first hybrid modification method can be improved by increasing the number of iterations.

Acknowledgments
This research is partially funded by Cluster research UI 2015 Grant.

References
[1] Deneubourg J L and Goss S 1989 Collective patterns and decision-making Ethology Ecology & Evolution 1 pp 295-311.
[2] Dorigo M, Maniezzo V and Colomi A 1996 The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics–Part B 26 1 pp 1-13.
[3] Gupta S and Panwar P 2013 Solving Travelling Salesman Problem Using Genetic Algorithm International Journal of Advanced Research in Computer Science and Software Engineering 3 6 pp 376-380.
[4] Heppner F and Grenander U 1990 A stochastic nonlinear model for coordinated bird flocks The Ubiquity of Chaos (Washington DC: AAAS Publications).
[5] Kennedy J and Eberhart R 1995 Particle swarm optimization 1995 IEEE International Conference on Neural Networks Proceedings 4 pp 1942–1948.
[6] Mahi M, Baykan Ö K and Kodaz H 2015 A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem *Applied Soft Computing* **30** pp 484–490.

[7] Reinelt G 1991 TSPLIB—A Traveling Salesman Problem *ORSA Journal on Computing* **3** pp 376-384.