PalArch's Journal of Archaeology of Egypt / Egyptology

A COMPARATIVE STUDY BETWEEN THE NEAREST-NEIGHBOUR ALGORITHM AND ITS VARIANTS FOR SOLVING THE EUCLIDEAN TRAVELING SALESMAN PROBLEM

Lilysuriazna Raya[1], *Safaa Najah Saud*[2]

[1,2]Faculty of Information Sciences and Engineering, Management & Science University, 40100 Shah Alam, Malaysia,

[1]lilysuriazna@msu.edu.my.[2]safaa_najah@msu.edu.my

Lilysuriazna Raya, Safaa Najah Saud. A Comparative Study Between The Nearest-Neighbour Algorithm And Its Variants For Solving The Euclidean Traveling Salesman Problem-- Palarch's Journal Of Archaeology Of Egypt/Egyptology 17(10), 938-945 ISSN 1567-214x

Key Words: Hybrid Heuristic, Nearest-Neighbour, Traveling Salesman Problem.

ABSTRACT

The Euclidean travelling salesman problem is a subcase of metric travelling salesman problem which also considered as NP-complete and typically solved using numerous heuristics methods. One of the most natural heuristics is the Nearest-Neighbour (NN) algorithm. In this paper, we propose a new hybrid heuristic approach based on the NN and Two Directional Nearest-Neighbour (2NN) algorithms for solving the travelling salesman problem. The underlying idea is to take advantage of these algorithms by fixing a certain number of cities to be inserted in the solution tour either by using the NN or 2NN algorithm. The number of cities is determined by the contribution ratio k . If n cities are selected following the NN algorithm, then the remaining $n-1$ cities will be selected following the 2NN algorithm and vice versa. The performance of this proposed hybrid heuristic is evaluated by using 12 TSP benchmark problems and the experimental results are empirically compared with the NN and 2NN approaches, respectively. The analysis shows that the proposed algorithm has a better performance than the conventional NN and 2NN heuristic with the average percentage of error of less than 13%. Since IoNN has tremendously improved the NN and 2NN solution, it has a great potential to be used as a construction heuristic in other heuristic or metaheuristic approaches.

INTRODUCTION

The Travelling Salesman Problem is a widely studied combinatorial optimisation problem [1]. Its popularity is due to the fact that it is easy to state but difficult to solve and has a larger number of applications. In spite of being very simple to formulate, the TSP is considered as an NP-hard problem [2].

The TSP is a problem to find the shortest tour of a certain number of cities provided that each city is visited once where the end city is also the starting city. Mathematically, the problem may be stated as finding an arrangement $(i_1, i_2, i_3, \dots, i_n)$ of the integers from 1 through n that minimises the value of $c_{i_1 i_2} + c_{i_2 i_3} + \cdots + c_{i_n i_1}$; where c_{ij} is the cost of going from city i to city j, for $(i, j = 1, \dots, n)$. For all i and j, the problem is said to be symmetric if $c_{ij} = c_{ji}$ and asymmetric if $c_{ij} \neq c_{ji}$. Furthermore, the total number of possible tours is equal to $(n-1)!/2$ for symmetric TSP and $(n-1)!$ for asymmetric TSP [2].

In the Euclidean travelling salesman problem (ETSP), cities are points in the Euclidean plane with coordinate (x_i, y_i) and (x_j, y_j) while the cost on each edge is the Euclidean distance between its endpoints given by $c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. The cost matrix for ETSP also satisfies the triangle inequality $c_{ij} + c_{jk} \geq c_{ik}$ for all i, j and k [3] [4].

The major techniques commonly used for solving the TSP are exact and approximate algorithms [5]. The exact algorithms are guaranteed to produces an optimal solution while approximate algorithms generate a near-optimal solution. The approximate algorithms generally can be classified into heuristic and metaheuristic. The approximate approaches can be used in many area such as vehicle routing [6], robot navigation [7], neural network [8] , delivery system [9] and even in software testing phase [10]. The types of heuristic algorithms are construction heuristics and improvement heuristics.

A construction heuristic builds a tour according to some rules without trying to improve them. A tour is sequentially built and parts that already built remain unchanged throughout the algorithm [11]. In contrast, improvement heuristics systematically tries to improve a tour even after a completed initial tour is found. The most commonly used improvement heuristics is 2-opt algorithm [12].

In recent years, many hybrid heuristics approaches have been proposed in the literature for the TSP that combines various construction heuristics such as nearest-neighbour with insertion algorithm [13], nearest-neighbour with greedy algorithms [14], the farthest vertex with greedy algorithm [15], nearest-neighbour with a construction priority based on the nearest-neighbours [12] and double-ended nearest-neighbour with the loneliest concept [16].

Here, a new approach is proposed for solving the Euclidean TSP that combines the conventional nearest-neighbour algorithm with its variant which is the Two Directional nearest-neighbour (2NN). The motivation behind this study is to prevent the “forgotten” cities that need to be inserted at high cost at the end of the nearest-neighbour algorithm.

CONSTRUCTION HEURISTICS

A construction heuristic starts with an empty tour and repeatedly extends the current tour until a complete tour is obtained. Such algorithms are greedy, nearest-neighbour and two directional nearest-neighbour.

Greedy Algorithm

A greedy algorithm is a simple and straightforward approach which makes a locally optimal decision at each step but not necessarily for all the future steps. The general framework of the greedy algorithm is:

Step1: Sort all the N edges.

Step2: Select the shortest tour and add to the current tour if it does not violate any of the subtour constraints.

Step3: Is the cardinality of the current tour is N? If no, go to Step2, otherwise, go to Step4.

Step4: Terminate the algorithm.

The greedy algorithm gives feasible solutions although they are not always good.

Nearest-Neighbour Algorithm

This algorithm was the first strategy that has been introduced and used for solving the TSP problem [15]. It starts with a randomly chosen city and repeatedly adds the closest unvisited city to the last city in the tour until all the cities have been visited [16]. The steps of the nearest-neighbour algorithm are given as:

Step1: Randomly pick the initial city.

Step2: Find the closest unvisited city and add to the current tour.

Step3: Is the cardinality of the unvisited cities is $\emptyset$? If not, repeat Step2, otherwise go to Step4.

Step4: Terminate the algorithm.

Since the tours quality might depend on the starting city chooses, a better result can be obtained by repeating the procedures for n different starting city

Two Directional Nearest-Neighbour Algorithm

The basic idea behind this variation is to consider nodes that are closer to the route’s both ends. A node with a minimum cost will be added to the current tour. The direction chosen in each step is guided by the minimum cost of a neighbour of all new vertices in both ends. By doing this, the current path can be extended from both of its end nodes. This algorithm is stated as follows:

Step1: Randomly pick the initial city.

Step2: Choose the nearest city from the initial city and add to the current tour.

Step3: Find the nearest unvisited city to these two cities, choose the city with the shortest distance and update the tour. Is the cardinality of the tour is N? If yes, repeat Step 2, otherwise go to Step 4.

Step4: Terminate the algorithm.

THE PROPOSED ALGORTIHM

This proposed algorithm begins with a randomly chosen node. Then, the next nodes to be chosen from the list of the unvisited nodes are based on the NN or 2NN algorithm. If $k * n$ numbers of nodes are chosen following the NN algorithm, then the remaining $(1 - k) * n$ unvisited nodes will be chosen following the 2NN algorithm and vice versa.

The contribution ratios of the algorithms are determined by the parameter k. If $k = 0$, the proposed algorithm is the 2NN algorithm; likewise, if $k = n$, the proposed algorithm is the NN algorithm. When k is in the interval $(0, n)$, the first k nodes is performed using the NN and the rest of the nodes using the 2NN algorithm. This hybrid algorithm is named as an *IorNN* algorithm. The procedures of this proposed algorithm are:

Step1: Identify the k parameter.

Step2: Choose a city using the NN algorithm and add to the current tour.

Step3: Is the cardinality of the current tour is $k * n$? If no, go to step 2, otherwise go to Step 4.

Step4: Choose a city using the 2NN algorithm and add to the current tour.

Step5: Is the cardinality of the current tour is $(1 - k) * n$? If no, go to Step 4, otherwise terminate the algorithm.

The computational experiments of this proposed algorithm were executed on Intel (R) Core (TM) i5-3470 CPU @3.20GHz with 8.00 GB of RAM and the algorithm was written in the *AMPL* language with CPLEX 12.5.1.0 solver.

EXPERIMENTAL RESULTS

To test the efficiency of the new proposed algorithm, some computational experiments have been carried out. The benchmark problems used in these experiments are from [19] and the optimum solutions for each of these problems are from [20]. Note that the numerical suffix in the dataset name shows the number of cities in that instance.

The comparative results of these experiments are provided in Table 1 with the best results being displayed in bold. Column 'best-known solution' denotes the optimal tour-length as reported in the TSPLIB standard library, column 'solution' represents the best result found for each algorithm and column '% error' expresses the percentage difference between the solution and the best-known solution.

The percentage error for each of the benchmark problems and algorithms, respectively, are calculated as:

$$\% \text{ Error} = \frac{\text{Solution} - \text{Optimal Solution}}{\text{Optimal Solution}} \times 100$$

The value of the k parameter use in this experiment is $k = 0.5$.

Table 1 presents that the proposed algorithm finds the best solution for all the benchmark problems with an average percentage of error less than 13% as depicted in Figure 1.

Table 1. Performance Comparison of the NN, 2NN and IorNN Algorithms

Problems	Best-known solution	Solution			% Error		
		NN	2NN	IorNN	NN	2NN	IorNN
ulysses16	6859	7943	7943	**6972**	15.80	15.80	1.65
ulysses22	7013	8180	8180	**7363**	16.64	16.64	4.99
eil51	426	514	518	**498**	20.66	21.60	16.90
eil76	538	620	600	**591**	15.24	11.52	9.85
kroB100	22141	2588	2571	**25621**	16.90	16.14	15.72
kroC100	20749	2356	2340	**23402**	13.58	12.79	12.79
kroD100	21294	2485	2559	**24622**	16.73	20.20	15.63
eil101	629	776	877	**772**	23.37	39.43	22.73
pr124	59030	6730	6752	**63132**	14.02	14.39	6.95
ch130	6110	7197	7018	**6977**	17.79	14.86	14.19
pr144	58537	6096	6254	**60963**	4.14	6.85	4.14
kroB150	26130	3132	3174	**31215**	19.86	21.50	19.46

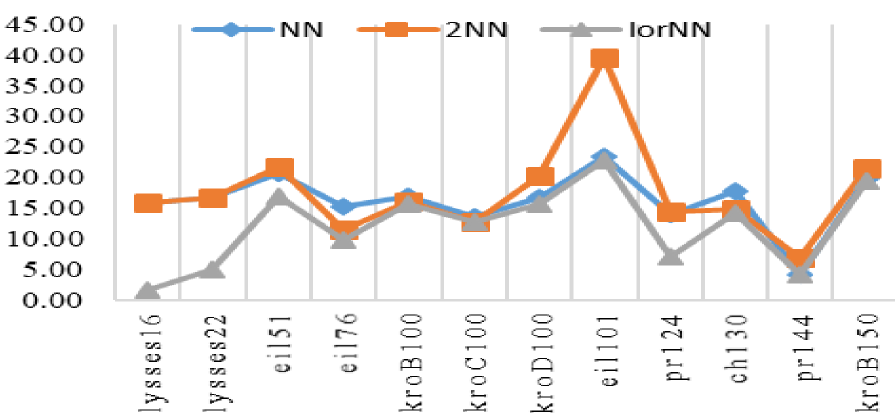

Figure 1. Average percentage error for all the algorithms

Figure 2 shows that the IorNN algorithm has outperformed the other algorithms with percentage error between 1.65% and 22.73% for all the benchmark problems. This further illustrates in Figure 3 and Figure 4 which specifies the gap difference between the proposed algorithm and the NN and the 2NN algorithms, respectively. However, there was no difference observed in pr144 for the NN and kroC100 for the 2NN algorithm.

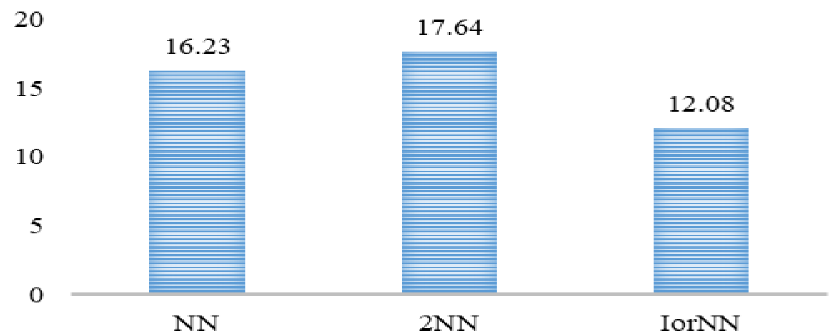

Figure 2. Percentage error of the NN, 2NN and IorNN algorithms

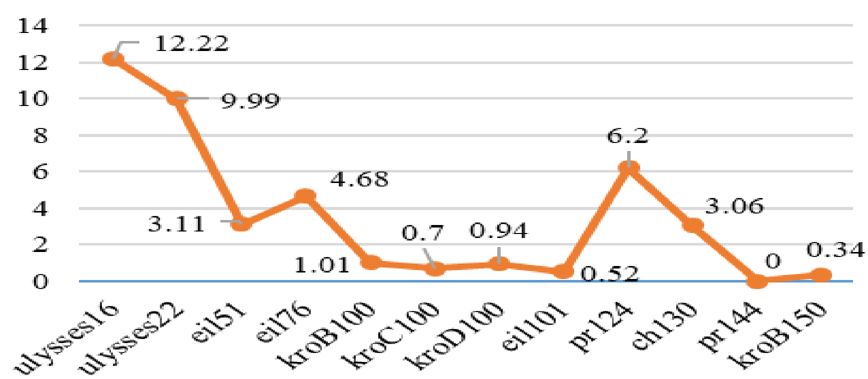

Figure 3. Percentage gap between the NN algorithm and the IorNN algorithm

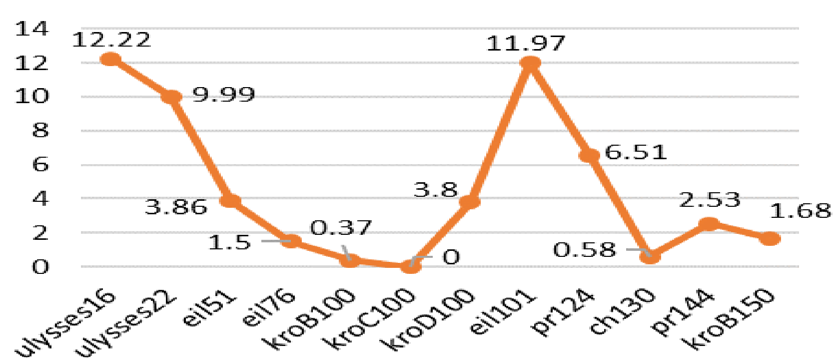

Figure 4. Percentage gap between the 2NN algorithm and the IorNN algorithm

DISCUSSION AND CONCLUSION

This paper presents a new hybrid algorithm named IorNN that is based on the NN and 2NN algorithms. The IoNN take advantage of the NN and 2NN algorithms by integrating both methods using a contribution ratio k.

The performance of the IoNN algorithm is evaluated using 12 symmetric TSP benchmark problems and the experimental results are then compared with the optimal solution, NN and 2NN algorithm, respectively. From this analysis, the new proposed algorithm has shown a better or equal performance to the NN and the 2NN algorithms in terms of accuracy and ability to find better solutions. Since the IorNN are able to minimize the percentage error of the NN and 2NN algorithms, it has a good potential to be use as a construction heuristic in other approaches. Besides, further research need to be carried out to verify its performance for other types of TSP including asymmetric TSP and multi TSP.

REFERENCES

M. S. Bazaraa, J. J. Jarvis and H. D. Sherali, ***Linear Programming and Networkk Flows***, vol. Fourth Edition, USA: A John Wiley & Sons, Inc., Publication, 2010.

M. R. Garey and D. S. Johnson, ***Computers and Interactability. A Guide to the Theory of NP-Completeness,*** USA: W. H. Freeman and Company, 1979.

K. Helsgaun, **An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic**, *European Journal of Operational Research,* vol. 126, pp. 106-130, 2000.

J. L. Gross, J. Yellen and P. Zhang, **Handbook of Graph Theory**, USA: CRC Press, 2013.

G. Laporte, **The Traveling Salesman Problem:An overview of exact and approximate algorithms**, *European Journal of Operational Research,* vol. 59, pp. 231-247, 1992.

T. Bektas, **The multiple traveling salesman problem : an overview of formulations and solution procedures**, *Omega,* vol. 34, no. 3, pp. 209-219, 2006.

D. Barral, J.-P. Perrin, E. Dombre and A. Liegeois, **An Evolutionary Simulated Annealing Algorithm for Optimizing Robotic Task Point Ordering**, in *Proceeding of the 1999 IEEE lnternational Symposium on Assembly and Task Planning*, Porto,Portugal, 1999.

L. Zajmi, F. Y. H. Ahmed and A. A. Jaharadak, **Concepts, Methods, and Performances of Particle Swarm Optimization, Backpropagation, and Neural Networks**, *Applied Computational Intelligence and So Computing,* vol. 2018, 2018.

Africa, Aaron.. **Design of an Aerial Drone Delivery System**. *International Journal of Emerging Trends in Engineering Research*. Vol 8. pp. 3116-3121, 2020.

Solanki, Dr & Dahiya, Omdev & Dhankhar, Amita. **An Exploratory Retrospective Assessment on the Usage of Bio-Inspired Computing Algorithms for Optimization**. *International Journal of Emerging Trends in Engineering Research,* vol 8, 2020.

E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan and D. B. Shmoys, ***The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization***, John Wiley & Sons, 1985.

G. Reinelt, ***The Traveling Salesman: Computational Solutions for TSP Applications***, Springer, 1994.

C.-P. Hwang, B. Alidaee and J. D. Johnson, **A Tour Construction Heuristic for the Travelling Salesman Problem**, *The Journal of the Operational Research Society*, pp. 797-809, 1999.

G. Kızılateş, F. Nuriyeva and H. Kutucu, **A Tour Extending Hyper-Heuristic Algorithm for the Traveling Salesman Problem**, in *Proceedings of IAM*, 2015.

F. Nuriyeva, G. Kızılateş and M. E. Berberler, **Improvements on Heuristic Algorithms for Solving Traveling Salesman Problem**, *International Journal of Advanced Research in Computer Science*, pp. 1-8, 2013.

G. Kızılates and F. Nuriyeva, **A Parametric Hybrid Method for the Traveling Salesman Problem**, *Mathematical and Computational Applications*, pp. 459-466, 2013.

F. G. S. L. Pimentel, **Double-ended nearest and loneliest neighbour - a nearest neighbour heuristic variation for the travelling salesman problem**, in *Proceedings of Trends in the Development of Machinery & Associat*, 2011.

C. Chauhan, R. Gupta and K. Pathak, **Survey of Methods of Solving TSP along with its Implementation using Dynamic Programming Approach**, *International Journal of Computer Applications*, pp. 12-19, 2012.

http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html.

http://elib.zib.de/pub/mp-testdata/tsp/tsplib/stsp-sol.html.