A Computation Investigation of the Impact of Convex Hull subtour on the Nearest Neighbour Heuristic

Emmanuel O. Asani
Department of Computer Science
Landmark University
Omu-Aran, Nigeria
asani.emmanuel@lmu.edu.ng
https://orcid.com000-0002-6774-8529

Aderemi E. Okeyinka
Department of Computer Science
Ibrahim Badamasi Babangida University
Lapai, Nigeria
aeokeyinka@ibbu.edu.ng
https://orcid.org/0000-0001-6186-314X

Ayodele Ariyo Adebiyi
Department of Computer Science
Landmark University
Omu-Aran, Nigeria
ayo.adebiyi@lmu.edu.ng
https://orcid.org/0000-0002-3114-6315

2023 International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG) | 979-8-3503-2478-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/SEB-SDG57117.2023.10124469

***Abstract*—This study investigated the computational effect of a Convex Hull subtour on the Nearest Neighbour Heuristic. Convex hull subtour has been shown to theoretically degrade the worst-case performances of some insertion heuristics from twice optimal to thrice optimal, although other empirical studies have shown that the introduction of the convex hull as a subtour is expected to minimize the occurrences of outliers, thereby potentially improving the solution quality. This study was therefore conceived to investigate the empirical effect of a convex-hull-based initial tour on the Nearest Neighbour Heuristic vis-a-vis the traditional use of a single node as the initial tour. The resulting hybrid Convex Hull-Nearest Neighbour Heuristic (CH-NN) was used to solve the Travelling Salesman Problem. The technique was experimented using publicly available testbeds from TSPLIB. The performance of CH-NN vis-à-vis that of the traditional Nearest Neighbour solution showed empirically that Convex Hull can potentially improve the solution quality of tour construction techniques.**

Keywords— Travelling Salesman Problem, Nearest Neighbour Heuristic, Convex Hull, Node-based heuristics.

I. Introduction

This paper is an extension of the work originally presented at the 2020 International Conference in Mathematics, Computer Engineering, and Computer Science [1].

The scientific task of finding the optimized solutions to combinatorial problems is often approached using either exact or heuristic techniques. While they do not necessarily produce optimal solution, heuristic techniques drastically cut down the solution space to obtain acceptable solution in polynomial time p [2]. Heuristics have been classified by re-searchers based on divergent criteria in previous studies. For instance, studies by [3], [4], [5], and [6] classified heuristics into three, based on the atomicity of solutions procedure namely, Tour Construction, Improvement / Local Search Heuristics, and Compound Heu-ristics. Tour Construction heuristics iteratively construct based on specified criteria, fol-lowing three steps namely, initialization, selection and insertion. Improvement heuristics iteratively refine solutions until it is impossible to obtain a better solution [7, 8]. Compound heuristics deploy a cocktail of techniques and obtain the best performing combina-tion [9, 10, 11, 12, 13].

Heuristics have also been classified based on their solution paradigms into space-partitioning-based, edge-based, and node-based heuristics [14, 15]. Node-based heuristics build their tour by expanding the nodes one at a time till all the nodes have been inserted. Node-based heuristics must first decide which node to be used as the initial node, then determine the succeeding node to explore in each iteration, and where it will be inserted. heuristics. Node-based heuristics are chiefly Tour Construction techniques as well.

Constructive heuristics can effectively generate suitable heuristic solutions as well as very efficient initial solutions [16]. They are therefore often integrated with popula-tion-based heuristics in order to obtain very efficient initial solutions. Consequently, re-searches that model and generate high performing node-based/ tour construction solu-tions have been on the increase. In this regard, analytical studies have provided the requi-site scientific basis for some empirical findings, while others are being investigated. In particular, this study was inspired by the need to investigate the effect of convex-hull sub-tour on the Nearest Neighbour (NN) Heuristic, against the analytical standpoint that starting some insertion tours (also node-based like the NN) with convex hull theoretically degrades their worst case from twice optimal to thrice optimal [17]. Thus, this study in-vestigated the empirical effect of a convex-hull-based initial tour on the Nearest Neighbour Heuristic vis-a-vis the traditional use of a single node as the initial tour. The resulting hy-brid Convex Hull-Nearest Neighbour Heuristic (CH-NN) was used to solve the Travelling Salesman Problem. The performance of this technique was then comparatively evaluated with respect to two classic node-based techniques, namely Nearest Neighbour (NN) and Nearest Insertion (NI), using parameters such as solution quality and computation speed.

The article is organized into six sections. Among other things, section one includes a background to the study, motivation for the study, the goal of the study as well as the significance of the study. The second section includes an elucidation of, and a review of relevant literature on the Nearest Neighbour technique. Section three includes a formulation of the classic Travelling Salesman Problem. The fourth section describes the proposed method vis-à-vis the state-of-the-art Nearest Neighbour and Nearest Insertion Heuristics. The fifth section discusses the result and evaluation of the used techniques. The sixth section include the concluding statement.

II. THE NEAREST NEIGHBOUR HEURISTIC

The Nearest Neighbour starts with a city/node and builds the re-maining tour by joining the node closest to the starting node to the tour. This process is iterated for all the nodes that are not yet part of the tour until the tour is fully built and a Hamiltonian circuit is formed. This pro-cess is greedy in nature; thus, the performance is relatively poor [18]. The pseudocode for the Nearest Neighbour Heuristic is as follow:

Algorithm 1: Nearest Neighbour Heuristic

Input: set of nodes $V_{n=1,2,\ldots,n}$
Output: **Path T**

Start
Select **an arbitrary node $k \in V$**
Set $Path \leftarrow k$
while $\{Path\} \neq \{V\}$ **do**
    Find node $k+1$ $Path$ such that
    $dist(Path, k+1)$ is minimal
    set $Path \leftarrow k+1$
end while
$T \leftarrow path$
return T
End

Figure 1. depicts the NN process in a flowchart as follow:

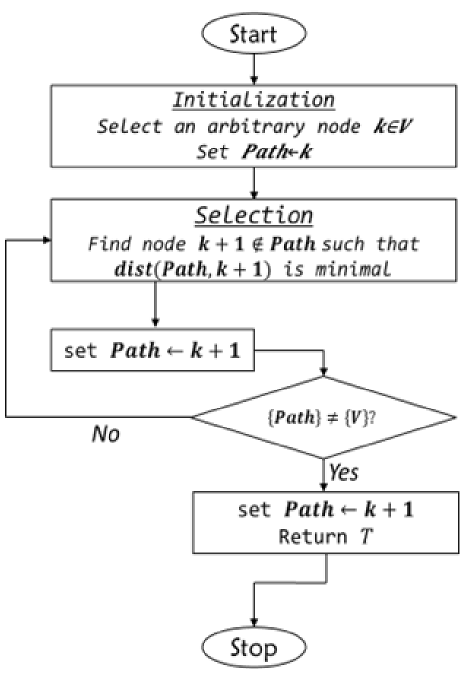

Figure 1. Flowchart of the Nearest Neighbour Heuristic

Analytically, [19] had shown that for a TSP instance of nodes n, the approximation ratio/solution quality of the Nearest Neighbour Heuristic is at most

$f_s/f_{OPT} = \frac{1}{2}[\log(n)] + \frac{1}{2}$ of the optimal length,

where f_s is the length of a tour by the solution and f_{OPT} is the optimal tour length.

The worst-case complexity is $T(n) = O(n^2)$.

However, in practice, the Nearest Neighbor Heuristic can solve the TSP in good time, with much better solution quality. Experimentally, the NN typically yields much better solutions than the worst-case suggests. NN often yields tour quality that is within 25%-30% of the Held-Karp lower bound [20]. The performance of NN greatly suffers from a phenomenon called the "*curse of dimensionally*" resulting from its greedy approach to solving the TSP. NN adds the lowest cost nodes as a priority, and consequently as the search space and nodes increase, more and more outliers are seen. Recent studies focus on how to minimize the incidences of outliers in order to circumvent the "*curse of dimensionally*". Other studies have equally used the Nearest Neighbor either as part of a hybrid method as in [1, 15, 21, 22] or as a seed technique in a metaheuristic for building initial solutions. We review some of these below.

Reference [23] used the Nearest Neighbour heuristic to build an initial tour in their experimental survey of some leading techniques. They identified important implementation success factors and experimented a total of nine high performing heuristics on different instances of both symmetric and asymmetric TSPs. These methods included four derivatives of the Lin-Kernighan technique and two stem and cycle (S&C) variants technique for the implementation of the Symmetric TSP; while three generalized LK and S&C methods were used for implementation on the Asymmetric TSPs. All the generalized methods' implementation used the Nearest Neighbour Heuristic to build their initial tour. Their findings revealed that S&C methods had better solution quality than the generic LK methods, while the LK performed better in terms of time. Reference [24] developed a hybrid metaheuristic algorithm for solving large-scale vehicle routing problem, the algorithm was a combination of Nearest Neighbour and TABU algorithms. The Nearest Neighbour was used to generate the initial routes while the TABU was used for the intra and cross-exchange routes. The testbed used in the experiments carried out was from a dataset of 6772 customers in the central and suburb of Suizhou city and from the evaluation, it was seen that the proposed algorithm was efficient in providing minimum cost for delivery. Reference [25] introduced an extension of the Traveling Salesman Problem (TSP), called problem quadratic TSP (QTSP). Three Exact algorithms (an exact approach based on a polynomial transformation to a TSP, branch-and-bound algorithm and branch-and-cut) and seven approximate algorithms (Cheapest-Insertion Heuristic (CI), Nearest-Neighbour Heuristic(NN), Two-Directional Nearest-Neighbour Heuristic (2NN), Assignment-Patching Heuristic (AP), Nearest-Neighbour-Patching Heuristic (NNP), Two-Directional Nearest-Neighbour-Patching Heuristic (2NNP) and Greedy Heuristic (GR)) were used to solve the QTSP. From computational evaluation, the branch-and-cut approach was seen to be capable of solving large real-world instances with up to 100 nodes and provided optimality in a reasonable time of about ten minutes. Although the running times of exact algorithms were reasonable, they were not as fast as heuristics which took less than or equal to ten seconds to solve the largest instances. The variants of the Nearest Neighbour presented did well in terms of computational speed but fell short in comparison to the exact methods in terms of solution quality. Authors [26] modified the Nearest Neighbour heuristic and hybridized it with the Greedy technique. The resulting hybridized method (NNDG) was implemented using some benchmark instances. NNDG generated a better solution. Authors [27] developed a hybrid technique comprising both the Nearest Neighbour and Nearest Insertion heuristics. method to form the NN-IN. They showed analytically that the hybridized NN-IN will produce a better solution. They then experimented their method on several TSP benchmarks. Their technique performed better than both the Nearest Neighbour and Insertion technique in more than 89% of cases. Authors [28] developed a hybridized 'adaptive-type' neural network (Convex-Elastic Net - CEN) and a non-deterministic iterative (NII) algorithm. The CEN initial solution was iteratively refined by the NII to obtain a 'near-optimal' result which outperformed similar techniques.

Theoretically, the introduction of the convex hull as a subtour is expected to minimize occurrences of outliers, thus circumventing the problem of "curse of dimensionality". To the best of our knowledge, the empirical possibility of this has not been explored in any study.

III. PROBLEM FORMULATION

First formulated in the nineteenth century and enhanced in the 1930s by M.M. Flood, the Travelling Salesman Problem has become the benchmark for several other techniques of optimization [1]. The TSP is a shortest tour (or path) problem to find the optimal route while visiting a set of cities (or nodes), ensuring each city (or node) is visited exactly once and regarding the Hamiltonian circuit, return to the start node or city [18]. The Travelling Salesman must traverse cities $1\ to\ n$ in a Hamiltonian cycle that is; Start from city 1 and traverse the remaining $n-1$ cities in arbitrary order, and return to the starting point with the objective of touching the cities once at a minimal cost. The distance $d(i,\ j)$ depicts the distance from city $i\ to\ j$. Thus TSP is formally defined as below;

$$F = \min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \quad (1)$$

$$\sum_{j=1}^{n} x_{ij} = 1; i = 1,\ \ldots,\ n \quad (2)$$

$$\sum_{i=1}^{n} x_{ij} = 1; j = 1,\ \ldots,\ n \quad (3)$$

The objective function is marked with F. With a limitation,

$$x_{i_1 i_2} + x_{i_2 i_3} + \ \ldots + \ x_{i_r i_1} \leq r - 1$$

$x_{ij}\, x_{ij}$ are the binary variables

$$x_{ij} = \begin{cases} 1 & if\ the\ salesman\ travels\ from\ city\ i\ to\ city\ j \\ 0 & if\ the\ salesman\ is\ not\ travelling\ from\ city\ i\ to\ city\ j \end{cases} \quad (4)$$

d_{ij} is the distance from city i to city j.

$$total\ distance = \sum_{(i,j)\in E} d_{ij} x_{ij} \quad (5)$$

The objective is to obtain a minimal value in equation (4), given that the salesman traverses all the nodes at most once and the tour must be a complete graph.

Thus, TSP is formulated as follows:

$$minimize \sum_{(a,b)\in E} d_{ab} x_{ab} \quad (6)$$

$$subject\ to \sum_{b\in V} x_{ab} = 2 \quad \forall i \in \boldsymbol{V} \quad (7)$$

$$\sum_{a.b\in S, a\neq b} x_{ab} \leq |S| - 1 \ \forall S \subset V, S \neq \emptyset \quad (8)$$

$$x_{ab} \in \{0,1\}$$

IV. THE PROPOSED TECHNIQUES

A hybrid of Convex Hull and Nearest Neighbour heuristic (CH-NN) is proposed in this study. This was motivated by studies by conducted by authors [14, 15, 20] suggesting that node-based techniques which are based on the use of polygons as their initial tour produce superior tour. The Nearest Neighbour Heuristic is a node-based technique which builds circuits on a node-by-node basis. The Nearest Neighbour tour begins with a single, fixed or random city/node and iteratively adds unvisited nodes closest to the tour, terminating only when there are no more unvisited nodes.

The hybrid technique starts by constructing a convex initial tour and completing the cycle by applying the Nearest Neighbour principle. Thus, given a set of specified vertices, the method begins by constructing the convex hull associated with the nodes, start the tour from a point in the edge, then inserts the nearest neighbor of subsequent nodes not already on the tour. The Convex Hull P of a set of points (S) in the Euclidean plane is the minimum convex polygon that encompasses all the points in S. It is generally used to build initial points of an approximation technique [29].

Thus, given the Convex Hull subtour, the Convex Hull Nearest Neighbour (CH-NN) technique is depicted in the pseudocode below:

Algorithm 2: The Convex Hull-Nearest Neighbour (CH-NN) Algorithm

1. **Start** with a partial tour formed by the convex hull of all cities
2. **For** each city not yet inserted in the partial tour find the edge (i, j), belonging to the partial tour, that minimizes $C_{ik} + C_{kj} - C_{ij}$ find the city $k + 1$ that is not yet in the tour and that is closer to **k**
3. **Insert** $k + 1$ at the end of the partial tour
4. **If** *all cities are inserted* **then**
   STOP
   else
   go back to 2
   end
5. End

V. Performance Evaluation and Discussion

The hybrid algorithm (CH-NN), as well as the Nearest Neighbour (NN) and Nearest Insertion Heuristics (NI), were experimented on ten (10) benchmark instances from TSPLIB, made available by Heidelberg University on http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/. The algorithms were implemented using Python programming language running on Intel Pentium Core i7 3GHz, Windows 10 (64bit). The experimental evaluation was undertaken using two criteria namely computational speed and solution quality. Table 1 shows the computational speed evaluation of the CH-NN vis-à-vis the NN and NI.

Table I. Computational speed evaluation of CH-NN, NN, and NI

S/N	Instances	$T_{Average}(S)$		
		NN	NI	CH-NN
1	*berlin52*	0.0059	0.0054	0.0034
2	*bier127*	0.0306	0.0279	0.0166
3	*ch130**	0.0427	0.0401	0.028
4	*ch150*	0.043	0.0400	0.0230
5	*d493**	0.4688	0.435	0.2508
6	*d657**	0.8425	0.7824	0.4454
7	*d1291*	3.4465	3.2159	1.8268
8	*pcb442*	0.3647	0.3387	0.1983
9	*pr76**	0.0185	0.0178	0.0135
10	*pr144*	0.0461	0.0433	0.0277

For all the ten benchmark instances considered, the CH-NN, outperformed both the NN and the NI in terms of computational speed. Thus, it can be inferred that the introduction of Convex Hull as subtour for the Nearest Neighbour Heuristic significantly reduces the time of computational.

In evaluating the solution quality of the heuristics, the following parameters were deployed:

Percentage Error (δ): the percentage error of the heuristics' solution quality is the percentage deviation of the solution from the optimal tour solution. This is computed as:

$$\frac{soln - opt}{opt} \times 100\% \qquad (6)$$

where $soln$ is the solution cost obtained by each heuristic, and opt is the optimal solution cost. This is the same thing as the performance ratio for non-optimal heuristics

Quality impr. (Σ): this the improvement of the CH-NN method's solution quality with respect to NN and NI. This is computed by:

$$E_{NN} - E_{CH-NN} \qquad (7)$$

where E_{NNH} is the percentage error of the NN and E_{CH-NN} is the percentage error of the CH-NN.

Goodness Value (ℊ): this is also referred to as accuracy. This is the inverse of error and is computed as

$$\left(1 - \frac{soln - opt}{opt}\right) 100\% \qquad (8)$$

Table 2 displays the benchmark instances with their optimal tour cost and the maximum number of solutions while Table 3 displays the *percentage error, quality impr and goodness value* of CH-NN, NN, and NI on the ten benchmark instances.

Table II. Benchmark instances with their optimal tour cost and maximum number of solutions

S/N	Instances	Solutions	Optimal Tour Length
1	*berlin52*	52	7542
2	*bier127*	127	118282
3	*ch130**	130	6116
4	*ch150*	150	6528
5	*d493**	493	35002
6	*d657**	200	48912
7	*d1291*	100	50801
8	*pcb442*	442	50778
9	*pr76**	76	108159
10	*pr144*	144	58537

Table III. The *percentage error, quality impr and goodness value* of CH-NN, NN, and NI on the ten benchmark instances.

Instance	Σ	δ(%)			ℊ (%)		
		CH-NN	NN	NI	CH-NN	NN	NI
berlin52	12.7	4.9	17.6	16.5	95.1	82.4	83.5
*pr76**	33.4	9.8	43.2	5.3	90.2	56.8	94.7
bier127	25.8	0.1	25.9	14.3	99.9	74.1	85.7
*ch130**	14.2	9.1	23.3	6.9	90.9	76.7	93.1
pr144	8	2.9	10.9	7.7	97.1	89.1	92.3
ch150	5.2	9.4	14.6	10	90.6	85.4	90
pcb442	8.6	16.3	24.9	15.4	83.7	75.1	85.6
*d493**	21	3.2	24.2	11	96.8	75.8	89
*d657**	21.9	11.3	33.2	12.7	88.7	66.8	87.3
d1291	1.2	21.1	22.3	21.4	78.9	77.7	78.6

The result of the CH-NN is further compared with similar study on hybridized Nearest Neighbour and Nearest Insertion TSP solution by [21]. Table 4 shows the ***Percentage Error (δ)*** of the CH-NN and the NN-NI on some benchmark instances.

Table IV. The *percentage error* of CH-NN and NN-NI [21] on some benchmark instances.

Instance	δ(%)	
	CH-NN	NN-NI
berlin52	4.9	13.38
*pr76**	9.8	9.7
bier127	0.1	4.58
*ch130**	9.1	11.60
pr144	2.9	1.66 11
ch150	9.4	14.10
pcb442	16.3	18.72
*d493**	3.2	8.76
*d657**	11.3	11.68
d1291	21.1	21.42

The results show that CH-NN outperformed the Nearest Neighbour (NN) and the Nearest Insertion (NI) heuristics in terms of closeness to optimality (as depicted by their respective ***Percentage Error (***$\boldsymbol{\delta}$)) except in three (3) instances (ch130, pcb442, and pr76) where NI performed better in terms of closeness to optimal solution only. The CH-NN equally performed better in comparison to similar study on hybridized Nearest Neighbour and Nearest Insertion TSP solution by [21] as shown in Table 4.0. On the average, the NN tour quality was 24.01% worse than the optimal tour. Equally, the average performance of NI for the instances considered was 12.12% of the Held-Karp lower bound. The Nearest Neighbour Heuristic reached a peak of 43.2% and a base value of 10.9%. The Nearest Insertion Heuristic reached a peak of 21.4% and a base value of 5.3%. These performances are consistent with documented findings about NN and NI in literature [30, 31]. On the other hand, the performance of CH-NN was on average, 8.81% worse than the optimal tour length. On average, the hybrid CH-NN has a significant quality improvement average of 15.2% over NN. Figure 2 shows a chart of the percentage deviation of NN, NI, and CH-NN from the optimal tour length.

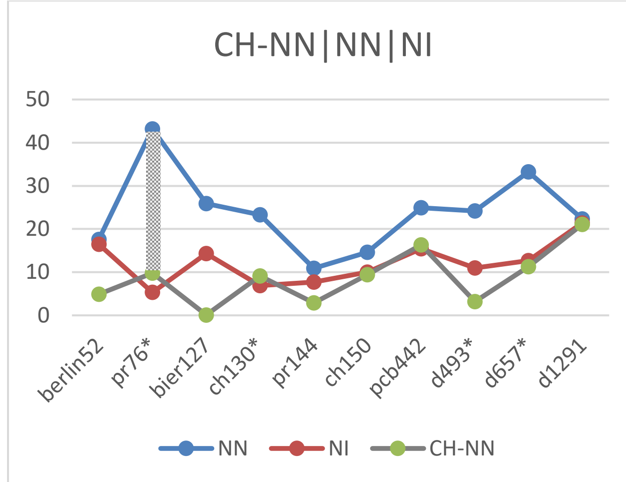

Fig. 2. Percentage error of NN, NI and CH-NN on ten benchmark instances

The hybrid CH-NN consistently outperformed the Nearest Neighbour and Nearest Insertion Heuristic across a wide spectrum of benchmark instances with statistical significance of as much as 33% at some point as highlighted by the shaded area of quality improvement in Figure 2.0. The average goodness value of the hybrid CH-NN was 91.19% compared to 87.88% for the Nearest Insertion and 75.99% for the Nearest Neighbour Heuristic. This is displayed in Figure 3.

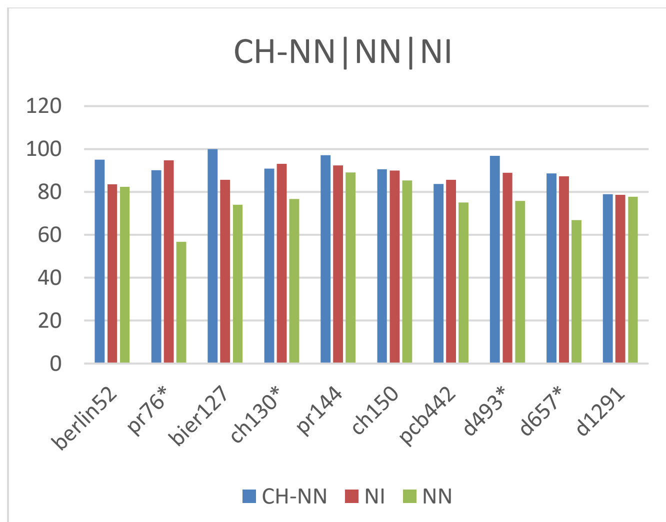

Figure 3. Measure of goodness value of CH-NN, NN and NI

This means that the introduction of the convex hull as a subtour for the Nearest Neighbour Heuristic is computationally more efficient and accurate than the traditional use of a single node as an initial tour for the Nearest Neighbour Heuristic.

VI. CONCLUSION

In this study, the computational effect of a convex hull subtour on the Nearest Neighbour Heuristic was investigated. An earlier study has posited that convex hull subtour theoretically degrades the worst-case performances of some insertion heuristics from twice optimal to thrice optimal. Other studies showed empirical promises that the introduction of the convex hull as a subtour is expected to minimize the occurrences of outliers, thereby circumventing the problem of "curse of dimensionality". To the best of our knowledge, the empirical possibility of this, as it pertains to the Nearest Neighbour Heuristic has not been explored in any study. Thus, the hybrid CH-NN was experimented on some benchmark instances from TSPLIB vis-à-vis the Nearest Neighbour and the Nearest Insertion Heuristics. The result showed that the hybrid heuristic outperformed NN and NI in terms of quality of results and computational speed. Thus, it can be inferred that the introduction of Convex Hull as subtour for the Nearest Neighbour Heuristic significantly improves the complexity and performance of the Nearest Neighbour Heuristic. Future research may include the integration of the polygon subtour to new improved construction heuristics to generate tours with better solution quality. Future works may also include the introduction of complexities curtailing techniques to further improve the computation speed.

ACKNOWLEDGMENT

We gratefully acknowledge the support of the Landmark University Center for Research Innovation and Development (LUCRID) for access to research repositories, literary materials, useful insights from affiliate researchers and funding.

REFERENCES

[1] E. O. Asani, A. E. Okeyinka and A. A. Adebiyi, "A Construction Tour Technique For Solving The Travelling Salesman Problem Based On Convex Hull And Nearest Neighbour Heuristics" Proceedings of the 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), 1-4, 2020. doi: 10.1109/ICMCECS47690.2020.240847.

[2] P.M. Todd, "Heuristics for Decision and Choice" International Encyclopedia of the Social & Behavioral Sciences, 6676-6679, 2001

[3] G. Manfred, "An Introduction to Optimization Heuristics" 2004 Available online: http://www.unige.ch/ses/dsec/static/gilli/CyprusLecture2004.pdf. Accessed 13 Feb. 2019

[4] J.F. Oliveira and M.A. Carravilla, "Heuristics and Local search" 2009 Available online: https://paginas.fe.up.pt/~mac/ensino/docs/OR/CombinatorialOptimizationHeuristicsLocalSearch.pdf. Accessed 13 Feb. 2019.

[5] R. Marti and G. Reinelt "Heuristic Methods. The linear Ordering Problem Exact and Heuristic Methods in Combinatorial Optimisation" Springer, -Verlag Berlin Heidelberg. ISBN: 978-3-64-16728-7, 17-40, 2011. DOI: 10.1007/978-3-642-16729-4 2

[6] M. Kyritsis, S.R. Gulliver, E. Feredoes and S. Ud Din "Human behaviour in the Euclidean Travelling Salesperson Problem: Computational modelling of heuristics and figural effects" Cognitive Systems Research 52, 387-399, 2018.

[7] F. Mtenzi, "Improvement heuristics for the Sparse Travelling Salesman Problem" Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, 101-108, 2006.

[8] L.G. Tavares, H.S. Lopes and C.R.E. Lima, "Construction and improvement heuristics applied to the capacitated vehicle routing problem" Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, 690-695, 2009. doi: 10.1109/NABIC.2009.5393467

[9] G. Frederickson, M. Hecht and C. Kim "Approximation Algorithms for Some Routing Problems," SIAM J. Computing T, 178—193, 1978.

[10] M.A. Langston, "A Study of Composite Heuristic Algorithms" Journal of Operational Research Society. 38, 539-544, 1987.

[11] A.C. Yao, "New Algorithms for Bin Packing," J. Ass. Comput. Mach. 27,207-227, 1980.

[12] B.L. Golden and W.R. Stewart, "Empirical Analysis of Heuristics in The Travelling Salesman Problem: A guided tour of Combinatorial Optimization" E. W. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan and D. B. Smoys (eds), Wiley, Chichester, 207-249, 1985.

[13] M. Gendreau, A. Hertz and G. Laporte, "New Insertion and Postoptimization Procedures for the Travelling Salesman Problem" Operations Research, 40, 1086-1094, 1992.

[14] W. Huang, J.X. Yu, Z. Shang, "A Sketch-First Approach for Finding TSP", In: Cheema M., Zhang W., Chang L. (eds) Databases Theory and Applications. ADC 2016. Lecture Notes in Computer Science, vol 9877, (2016), Springer, Cham. https://doi.org/10.1007/978-3-319-46922-5_10

[15] W. Huang and J.X. Yu, "Investigating TSP Heuristics for Location-Based Services", Data Sci. Eng. 2, 71–93, 2017. DOI 10.1007/s41019-016-0030-0.

[16] A.H. Ismail "Domino algorithm: a novel constructive heuristics for traveling salesman problem" IOP Conf. Series: Materials Science and Engineering 528 012043, 2019. doi:10.1088/1757-899X/528/1/012043

[17] A.R. Warburton, "Worst-case analysis of some convex hull heuristics for the Euclidean travelling salesman problem", Operations Research Letters 13, 37-42, 1993.

[18] E.O. Asani, P.O. Ayegba, J.A. Ayoola, A.E. Okeyinka and A.A. Adebiyi (2019). A Preliminary Study on the Complexity of Some Heuristics for Solving Combinatorial Optimization Problems. International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 10 (2019), pp. 1615-1620

[19] D.J. Rosenkrantz, R.E. Stearns, P.M. Lewis, "An analysis of several heuristics for the traveling salesman problem." SIAM J. Comput. 6(3) 563–581, 1977.

[20] Víctor Pacheco-Valencia, José Alberto Hernández, José María Sigarreta and Nodari Vakhania, "Simple Constructive, Insertion, and Improvement Heuristics Based on the Girding Polygon for the Euclidean Traveling Salesman Problem." Algorithms, 13 (5), 2020. doi:10.3390/a13010005.

[21] W. Rao and C. Jin, "A New Hybrid Algorithm for Solving TSP" Proceedings of ICLEM 2010: Logistics for Sustained Econimic Development, 3327-3334, 2010

[22] S. Lity, M. Al-Hajjaji, T. Thüm and I. Schaefer, "Optimizing product orders using graph algorithms for improving incremental product-line analysis", VAMOS '17: Proceedings of the Eleventh International Workshop on Variability Modelling of Software-intensive Systems, 60-67, 2017. https://doi.org/10.1145/3023956.3023961.

[23] César Rego, Dorabela Gamboa, Fred Glover and Colin Osterman "Traveling salesman problem heuristics: Leading methods, implementations and latest advances" European Journal of Operational Research 211, 427–441, 2011. https://doi.org/10.1016/j.ejor.2010.09.010

[24] D. Lingling and H. Ruhan "Combining Nearest Neighbor Search with Tabu Search for Large-Scale Vehicle Routing Problem." Physics Procedia 25, 1536 – 1546, 2012. doi: 10.1016/j.phpro.2012.03.273

[25] A. Fischer, F. Fischer, G. Jäger, J. Keilwagend, P. Molitor and I. Grosse "Exact algorithms and heuristics for the Quadratic Traveling Salesman Problem with an application in bioinformatic." Discrete Applied Mathematics 166, 97–114, 2014. http://dx.doi.org/10.1016/j.dam.2013.09.011.

[26] G. Kizilateş and F. Nuriyeva, "On the Nearest Neighbor Algorithms for the Traveling Salesman Problem" In Adv. in Comput. Sci., Eng. & Inf. Technol., AISC 225, 111–118, 2013. DOI: 10.1007/978-3-319-00951-3_11.

[27]M. Al-Mulhem and T. Al-Maghrabi, “Efficient Convex-Elastic Net Algorithm to Solve the Euclidean Traveling Salesman Problem” IEEE transactions on Systems, Man, and Cybernetics—part b: Cybernetics, 28(4), 1998.

[28]N.L.F. García, M.D. Martínez, A.C. Poyato, F.J.M. Cuevas and R.M. Carnicer, “Unsupervised generation of polygonal approximations based on the convex hull”, Pattern Recognition Letters, 135, 138-145, 2020. https://doi.org/10.1016/j.patrec.2020.04.014.

[29]G. Reinelt, “The Traveling Salesman: Computational Solutions for TSP Applications.” Springer-Verlag Berlin Heidelberg, 1994. DOI 10.1007/3-540-48661-5

[30] D.S. Johnson and L.A. McGeoch, “Experimental analysis of heuristics for the STSP.” In: Gutin, G., Punnen, A.P. (eds.) The Traveling Salesman Problem and Its Variants, 369–443, Kluwer, Dordrecht (2002).

[31]L. Babel, “New heuristic algorithms for the Dubins traveling salesman problem.” Journal of Heuristics 26: 503–530, 2020. https://doi.org/10.1007/s10732-020-09440-2