The Close Enough Traveling Salesman Problem with Time Window

Soukaina Semami, Hamza Toulni, Abdeltif ElByed

***Abstract*—**The Close Enough Travelling Salesman Problem with Time Window (CETSP-TW) is a new variant of the well-known Traveling Salesman Problem with Time Window (TSPTW), where the salesman does not need to visit the exact location of each customer. The goal of the CETSP-TW problem is to find the minimum distance Hamiltonian cycle through a set of nodes, where the requirement is only to come close enough to the node neighborhood set in a predefined time window. In this paper, we propose a mathematical formulation for the CETSP-TW and we design a fast heuristic to solve the new variant. The proposed heuristic is incorporated with the TSPTW effective optimization algorithms to find a near optimal tour in a short computation time. Computational results are taken on TSPTW Library instances. Performance evaluation of the proposed heuristic is discussed in detail, and the best solutions obtained from a selected set of instances are reported.

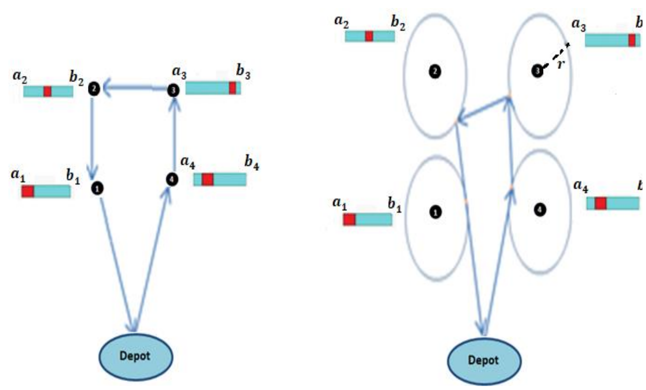
***Keywords*—**Close Enough TSP, Heuristic, RFID, Time Window.

I. Introduction

THE Travelling Salesman Problem with Time Window (TSPTW) [1] is a combinatorial optimization problem that has received substantial attention during the last decades, as it is core to many important real life routing and scheduling applications, it consist in finding a minimum cost tour starting and ending on a given location called depot and visiting a set of customers exactly once. Each customer has a time window or a service time defining its ready time and due date. If a tour visit a customer after its due date, the tour is considered infeasible and also if the salesman or the vehicle service arrives before the customer ready time, it must wait. The cost of a tour is translated by the total distance traveled.

On the other hand, The Close Enough Travelling Salesman Problem with Time Window abbreviated as CETSP-TW is a new variant of the TSPTW, where the salesman does not need to visit the exact location of each customer (node). Instead, a region of the plane containing each node is specified as its neighborhood set, and the goal is to find a shortest tour that starts from a specified depot location and intersects all of these neighborhood sets in order to be visited in a predefined Time Window for an extra travelling freedom. Intuitively speaking, in the CETSP-TW, each of the salesman's customers is willing to travel to any point inside its particular neighborhood to meet with the salesman in a predefined time. This feature allows savings from the standard TSPTW solution, but also adds great complexity to the problem as this creates an infinite number of routes between each customer. For that as the TSPTW, the CETSP-TW is among the class of NP-Hard problems that are computationally intensive to solve [2].

Note that without the time window feature the CETSP-TW variant became the Close Enough Traveling Salesman Problem known as CETSP problem [3] a recent variant of the classical Traveling Salesman Problem (TSP) [4].



(a) Optimal TSPTW tour. (b) Optimal CETSP-TW tour.

Fig. 1 The optimal TSPTW tour compared to the optimal CETSP-TW tour.

The Fig. 1 shows an instance of four customers for the TSPTW and the CETSP-TW respectively. In general the graph represents the planning of the problem, each node representing a customer. Each customer has a time window $[a_i, b_i]$ in which it can be served. The red segment in the time window present the services time of customer i. It also shows the order in which each salesman serves its customers. The CETSP-TW tour in Fig. 1.(b) needs only to get within a distance r (called radius) of each customer *i* in order to be visited in his predefined time, in contrary of the TSPTW tour in Fig. 1.(a) that must visit customer *i* in his own location during his time window. Obviously, the CETSP-TW optimal tour has a shorter total length than the one required by the TSPTW.

S. Semami and A. ElByed authors are with the Laboratory “Informatique et Modélisation de Système d’Aide à la décision,” Faculty of science, University of Hassan II, Casablanca 20000, Morocco (e-mail: soukaina.sema@gmail.com; a.elbyed@gmail.com)

H. Toulni author is with the Laboratory “Génie Industriel, Traitement de l’Information et Logistique,” Faculty of science, University of Hassan II, Casablanca 20000, Morocco (e-mail: hamza.toulni@outlook.com).

The contribution of this paper is to present a Mixed Integer Nonlinear mathematical formulation for the CETSP-TW problem and to develop a three phased heuristic based on Nearest Neighborhood Search (NNS) [5] procedure and TSPTW effective algorithms to solve it. Since this is the first study of the CETSP-TW, our purpose is to define general properties of the CETSP-TW problem and to present its resolution possible strategies. Nevertheless, the computational results indicate that the proposed Perimeter Neighborhood Search Time Window (PNSTW) heuristic is very promising to solve the CETSP-TW efficiently regarding solution quality and running time.

The remainder of the paper is described as follows: Section 2 gives a brief review of the literature related to the CETSP-TW problem and its application field. Section 3, is devoted to the problem definition and mathematical formulation. Description of the proposed resolution approach and the Computational results are presented and discussed in section 4. Finally, Section 5 summarizes our conclusions and future directions.

II. Problem Background

Since the problem studied in this paper is related to the Close Enough Traveling Salesman Problem (CETSP), we briefly cite the related literature and contrast our variant with the variants in the literature of this area.

The CETSP was first introduced by [3]. The authors proposed six heuristics developed by a team of graduate students as part of a class project to solve the CETSP. All the six heuristics followed three common steps: First, a set of supernodes was selected (a supernode is a point in the plane that represents a customer neighborhood) such that if each supernode is visited, the tour will have come within the appropriate distance of each node (customer). Second, a TSP tour through the selected supernodes and the depot was generated. Third, the obtained tour was improved while maintaining feasibility. Based on [3] conclusions, the authors in [6] and in [7] proposed a generic three phased approach called Steiner Zone heuristic (SZH). First, the goal of the SZH is to identify intersection between disks called Steiner Zone in which the tour comes close enough to multiple nodes. Second, each identified Steiner zone or intersection area is presented by a point (supernode known as well as representative or hitting point), and then a TSP tour is constructed over these points. Third, the TSP sequence found earlier is improved using a TPP (Touring a sequence of Polygon Problem). The authors also ameliorated the heuristic presented in [8] and modified the CETSP to be solved as a Generalized Traveling Salesman Problem (GTSP) based on [9] works. With constant radius and disks-shaped-neighborhoods, the heuristics were tested on 48 instances and on 14 instances with different radius but in both cases the SZH produced good results regarding solution quality and running time.

From the perspective of this paper, many of the techniques and heuristics developed to solve the CETSP problem can be utilized in solving the CETSP-TW since both the CETSP and the CETSP-TW need to deal with a continuous problem (the optimization of the hitting points locations) in first place as we show in Section 3.

Recently, numerous studies of the CETSP problem used discretization methods to discretize a set of continuous covering neighborhoods into so called cluster and requiring the salesman to visit each cluster exactly once. The authors in [10] used only 4 points to discretize each neighborhood and used Benders decomposition to find tight lower and upper bounds as well. The proposed method was very expensive in term of running time due to the high partitioning levels and no solution was proven to be optimal. All of the instances created by Behdani were solved to optimality by [11] later on using an exact Branch and Bound algorithm. The authors in [11] also solved instances with as many as 1000 customers from [6] with a large customer covering radius. In 2017, authors in [12] improved the discretization scheme already proposed in [10] using Perimetric Discretization scheme and Internal Point discretization scheme which provided better and tighter upper and lower bounds. Also, they transformed the CETSP to a Generalized TSP (GTSP) by using an arc discretization scheme and then reduced the problem size by applying a graph reduction algorithm. But, the proposed discretization cannot guarantee a globally optimal or nearly optimal solution in a given time as it was discussed in [13] who encoded scheme for neighborhoods with different shapes joint or disjoint, regular or irregular. The authors were able to reduce the search space without degrading the quality of the generated solution.

More recently in 2019, the authors in [14] developed a three phased heuristic called Steiner Zone Variable Neighborhood Search heuristic (SZVNS) to solve the CETSP problem. In phase I, the SZVNS trims the problem size to reduce running time. In phase II, the sweep line algorithm identifies the Steiner zones, then a minimum number of those Steiner zone are selected by solving a set covering problem to cover all the customers. Steiner points are chosen from these selected Steiner zones based on three rules giving rise to three feasible tours, the best of these tours was kept by SZVNS. In phase III, a Variable Neighborhood Search procedure (VNS) improves the retained feasible tour. The heuristic produced optimal solutions on state of art instances in shorter times but do not assume any information on customer radius.

On the other hand, the CETSP-TW variant underlies several interesting applications in real world problems. In the Automated Meter Reading (AMR) context for example, utility companies used to send their technicians to read the meters at every customer location. Nowadays, using Radio Frequency Identification (RFID) technology meters can be read from a specific distance instead of visiting every customer, technicians only need to get close enough to a customer residential location to read the meter according to his availability time. Another application of the CETSP-TW arises in the Unnamed Aerial Vehicle (UAV) operations context, when a pilot in an airplane or an unmanned drone survey several ground targets. The aircraft does not have to fly directly above the targets but only has to get close enough in a predefined time to survey them. The UAV mission planning problem in military operations context where each target location has its own visit priority can also be modeled as a CETSP-TW problem. In fact, other similar applications exist in submarine reconnaissance of coastlines, delivering munitions to targets, ship tracking, aerial forest fire detection, robot monitoring of wireless sensor networks described in [13].

III. Problem Formulation

We are given an undirected graph $G = (V, E)$ where $V = \{v_0 \dots v_{n-1}\}$ is the set of nodes and $\mathrm{E} = \{v_i, v_j / v_i, v_j \in V\}$ is the set of edges. Node v_0 represents the depot. Each node i ∈V other than the depot is surrounded by a disk D_i with radius r assumed to be the same for all disks. The requirement to visit node i is only to hit or to pass through disk D_i exception made for the depot. Let (m_i, n_i) be the Cartesian coordinates of node i in the plane and let (x_i, y_i) be the Cartesian coordinates of the hitting points on the tour that is close enough to node i known as node's i representative point in [6] and referred to in this paper as p_i where set P is the set of the candidates hitting points while $p_0 = v_0$ is the depot. The service at node i should begin within a time window $[a_i, b_i]$. A time window is associated to each node other than the depot. Early arrivals are allowed, in the sense that the salesman can arrive before the time window lower bound. However, in this case the salesman has to wait until the node i became ready for the beginning of the service or information change session in case of using RFID meter reading or wireless sensor network operations. In brief as shown in Fig. 2 each customer i is characterized by:

- A disk shaped neighborhood referred to as D_i.
- A service time β_i that indicate the needed time to service the customer i and the service start time denoted by s_i that indicates the beginning time of the service.
- A ready time a_i or (a release time) indicate when is possible to start serving customer i.
- A due time b_i or (a deadline) when the customer should been already served.
- An arrival time t_i that indicates the arrival time of the salesman to service customer i.

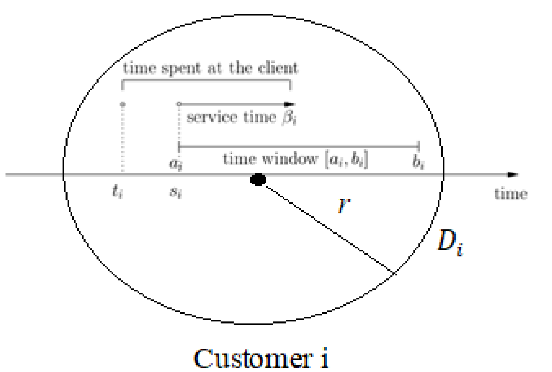


Fig. 2 An illustration of a customer i characteristics.

We define a binary decision variables k_i (which equals one if node i is represented by a hitting point p_i on the tour i.e. node i have been visited on the tour, and zero otherwise) and C_{ij} (which equals one if node j is visited (or represented) right after node i, this means that the path from node i to node j is selected in the tour and zero otherwise). The objective of optimization is to find the shortest travel tour, along which the salesman can meet all customers in their predifined time windows and come back to the starting position (depot).

Our Mixed Integer Nonlinear formulation of the CETSP-TW is given as follows:

$$\min \sum_{i=0}^{n} \sum_{j=0}^{n} C_{ij} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (3.1)$$

Subject to:

$$(x_i - m_i)^2 + (y_i - n_i)^2 = r^2, \quad for\ i = 1, \dots, n \qquad (3.2)$$

$$\sum_{i \in V:\,(i,j) \in E} C_{ij} = k_j, \qquad \forall\, j \in V \qquad (3.3)$$

$$\sum_{j \in V:\,(i,j) \in E} C_{ij} = k_i, \qquad \forall\, i \in V \qquad (3.4)$$

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq £, \quad for\ i, j = 1, \dots, n \qquad (3.5)$$

$$\sum_{i,j \in V:\,(i,j) \in E} C_{ij} \leq |V| - 1, \ \forall\, V \subseteq \mathbb{N} - \{1\} \qquad (3.6)$$

$$\beta_j + t_i - t_j - \beta_i \leq (1 - C_{ij})T, \ \forall\, i, j \in V \subseteq \mathbb{N} - \{0\} \qquad (3.7)$$

$$0 \leq t_i \leq a_i\,, \qquad \forall\, i, \in V \subseteq \mathbb{N} - \{0\} \qquad (3.8)$$

$$a_i k_i \leq s_i \leq b_i k_i\,, \qquad \forall\, i, \in V \subseteq \mathbb{N} - \{0\} \qquad (3.9)$$

$$k_0 = 1 \qquad (3.10)$$

$$C_{ij},\ k_i \in \{0,1\},\ (x_i, y_i) \in \mathbb{R} \times \mathbb{R},\ (m_i, n_i) \in \mathbb{R} \times \mathbb{R}, \quad \forall (i,j) \in E,\ i \in \mathbb{N} \qquad (3.11)$$

The objective function (3.1) minimizes the total distance of the global travel tour. However, the distance coefficients are not constants. The coefficients are functions of the hitting points. Constraint (3.2) requires each hitting point p_i to be located on the boundaries of the disk that surround node i determined in the plan by its Cartesian coordinates i (m_i, n_i). Constraints (3.3) and (3.4) are the edges degree constraints known as the assignment constraints that require each node to be visited exactly once and guarantee the connectivity of the obtained tour. Constraint (3.5) requires the distance between any two hitting points to be strictly positive where £ is defined as the smallest length of any edge. Without (3.5), edges in the tour can have length 0. If k disks have the same hitting point, all $\binom{k}{2}$ edges between the k disks have length 0 and could be selected without increasing the objective function value as exactly explained in [6]. Constraints (3.6) are the sub-tour elimination constraints ensure that no sub-tour is present in a solution. Constraints (3.7) restricts the service time to the time window, while T is a given time limit which allows constraints (3.7) to be satisfied. Constraints (3.8) define the bounds of the arrival time to node i. Constraints (3.9) indicate that for each node i, the service start time s_i must begin within the time window $[a_i, b_i]$. Constraint (3.10) ensures that the depot is on the tour. Finally, constraint (3.11) defines the variables definition domain.

IV. The proposed Heuristic and the Computational Results

A. The proposed Heuristic

In the Close Enough TSP context (CETSP), the CETSP-TW new variant is drastically relevant as we discussed earlier. However, methods for the CETSP cannot be used directly for the CETSP with Time Window. Since, in these methods time windows constraints are not considered and therefore applying these algorithms can violate the time window constraints. Although, a combination of solution approaches designed for CETSP and TSP with time window (TSPTW) would be very interessant to be used in our case. See the survey [15] for details about TSPTW algorithms and state-of-art instances.

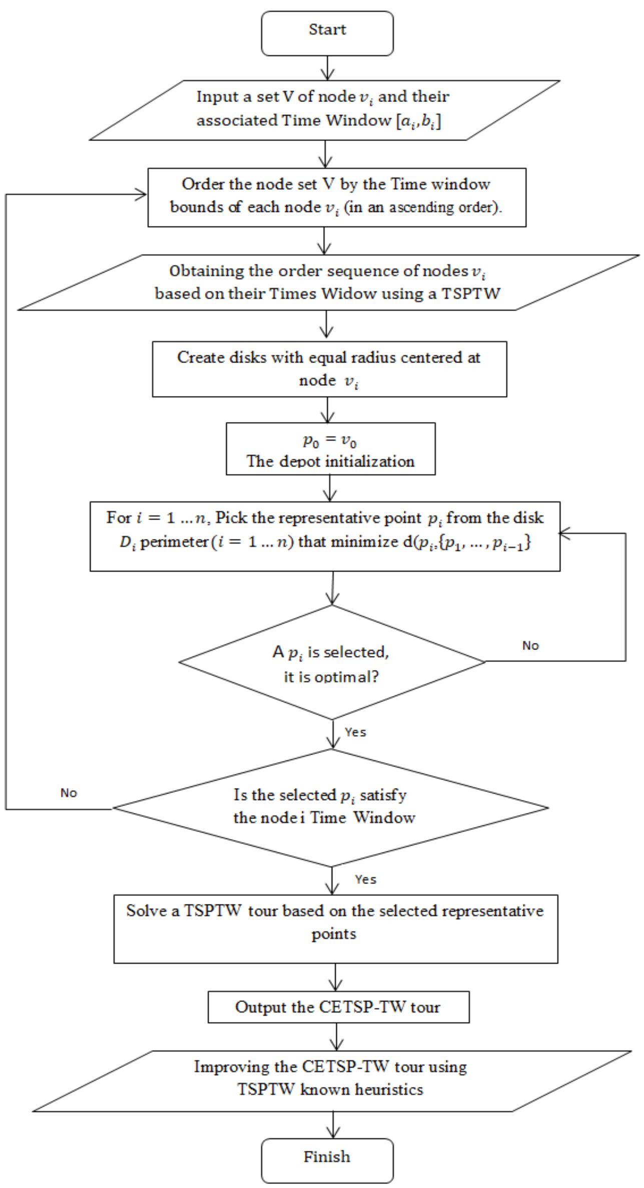


Fig. 3 Flow chart of the proposed heuristic to solve the CETSP-TW.

To solve the CETSP-TW, we develop a heuristic called PNSTW (Perimeter Neighborhood Search Time Window) to solve the CETSP-TW in a three phased procedure. The most distinct character of the proposed heuristic is to keep the selection of node's i hitting point only in the perimeter of disk D_i in order to deal with the time window constraints in a straight way without confusion by using the TSPTW features and techniques. We describe the details of the proposed PNS-TW heuristic as follow:

Phase I: We order the set of nodes V based on their time window service by comparing their ready time a_i and their due time b_i in an ascending order.

Phase II: Using the node's time window order, we start locating and selecting the node's i hitting point from the perimeter of the disk i that surround node i only.

Phase III: Construct a TSP time window (TSPTW) tour based on the best selected hitting points found in phase II and improve it using TSPTW state of art heuristics.

Using a simple TSPTW solver, an initial visiting sequence is determined in phase I based on the time window ordering. In phase II, we locate the node i hitting point based on the visiting sequence previously found. A hitting point is seen as results of the intersection between the circular boundary of each disk neighborhood and the straight line connecting the previous selected hitting point p_{i-1} to the current hitting point p_i to be added by the tour as shown in Fig. 4 in order to not affect the time constraint. Then, we create a feasible tour over these selected hitting points. Finally the feasible tour is iteratively improved by using TSPTW optimization effective algorithms. The flow chart of the proposed heuristic is illustrated in Fig. 3.

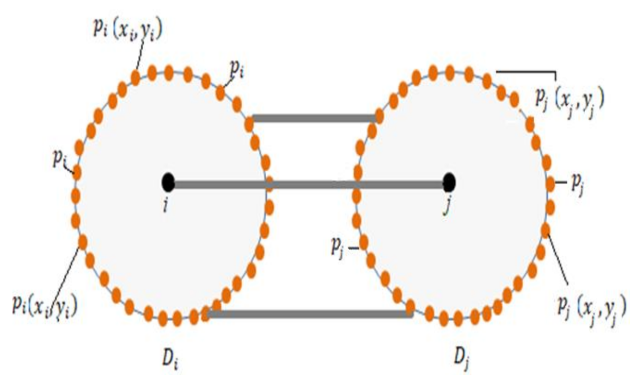


Fig. 4 A possible hitting points connection lines.

B. The Computational results

The proposed PNSTW heuristic was coded with Python in an Intel Core i5 with 4GB of RAM. We used the same instances proposed by Dumas et al. [1] available on line in [15] designed for the TSPTW problem and adapted to the

CETSP-TW context by adding a disk D_i centered at node i with the same radius r to evaluate the performance of the proposed PNSTW heuristic. A view of the adapted instances is shown in Fig. 5. The value of r was set to 200. The computational time is set to 3600 s. We used OR-tools [16] algorithms when dealing with TSPTW initial visiting sequence and to improve the CETSP-TW final tour as well. The performance of the proposed heuristic is compared to the best values stated for Dumas et al. [15] instances to highlight the gain from using the Close Enough Time Window concept.

CUST NO	mCOORD	nCOORD	RADIUS	ARRIVAL TIME	READY TIME	DUE TIME	SERVICE TIME
1	16.00	23.00	200	0.00	0.00	408.00	0.00
2	22.00	4.00	200	0.00	62.00	68.00	0.00
3	12.00	6.00	200	0.00	181.00	205.00	0.00
4	47.00	38.00	200	0.00	306.00	324.00	0.00
5	11.00	29.00	200	0.00	214.00	217.00	0.00
6	25.00	5.00	200	0.00	51.00	61.00	0.00
7	22.00	31.00	200	0.00	102.00	129.00	0.00
8	0.00	16.00	200	0.00	175.00	186.00	0.00
9	37.00	3.00	200	0.00	250.00	263.00	0.00
10	31.00	19.00	200	0.00	3.00	23.00	0.00
11	38.00	12.00	200	0.00	21.00	49.00	0.00

Fig. 5 The content of the file n20w20.001.txt from Dumas et al. [15] instances set adapted to the CETSP-TW context.

Table I. summarize the results for our test on Dumas et al. [15] instances adapted to the CETSPTW context as shown in Fig. 5. The columns of this table are as follow: Instance_ indicates the instance name;T_{OPT}_reports the best known optimal solutions obtained for the same instance; PNSTW_ T_{CE}_ reports the length of the shortest tour obtained by the proposed heuristic; Time (s)_denotes the computational run time in second; Gap (%)_indicates the gap between T_{OPT} and the T_{CE} tour length calculated as follow:

$$\text{Gap} = ((T_{OPT} - T_{CE}) / T_{OPT}) \times 100 \tag{4.1}$$

The best value of the tour length and Gap is shown in bold.

Table I. The outputs solution for the proposed PNSTW heuristic on Dumas et al. [15] instances set.

Instance	T_{OPT}	PNSTW		Gap (%)
		T_{CE}	*Time (s)*	
n20w20	370.4	**210.68**	118.03	**68.13%**
n20w40	342.8	**207.04**	114.55	**66.58%**
n20w60	362.0	**220.77**	121.03	**66.57%**
n40w20	521.2	**322.85**	89.56	**82,82%**
n40w40	512.2	**314.30**	100.39	**80.40%**
n40w60	481.4	**301.95**	100.67	**79.09%**
n40w80	486.6	**305.07**	98.73	**79.71%**
n40w100	463.0	**298.62**	99.35	**78.54%**
n60w20	626.8	**420.83**	133.89	**78.64%**
n60w60	672.8	**431.79**	145.90	**78.31%**
n60w80	628.2	**422.45**	143.67	**77.13%**
n60w100	620.2	**415.32**	105.49	**82,99%**
n80w20	748.2	**498.43**	155.87	**79.17%**
n80w60	712.6	**720.10**	547.87	**23.12%**
n100w20	823.0	**511.03**	200.05	**75.69%**
n100w40	821.0	**507.39**	198.21	**75.86%**
n100w60	817.2	**515.65**	189.98	**76.75%**
n150w20	978.4	**588.90**	300.74	**69.26%**
n150w40	990.4	**603.99**	476.84	**51.85%**
n150w60	981.4	**624.38**	501.68	**48.88%**
n200w20	1137.8	1140.71	3600	-216.4%
n200w40	1156.0	818.38	800.56	**30.75%**

In general from Table I, we can state that nearly all the instances were solved to optimality, about 20 instances out of 21 in a very suitable running time away from the test time limits except for the instance n200w20 that was solved in more than an hour. The reported Gap was also very significant with a value of 82.99 % for the n60w100 and the only negative value was recorded for the instance n200w20. We may doubt here that the run time increases with increasing the number of nodes, since only one instance was solved out of the test time limits with a quite high number of node. However, based on these results, we can state that the proposed heuristic proved a good performance in terms of both solution quality and computational time on various sizes of Dumas et al. [15] instances which leads us to confirm that the gain from using the Close Enough Time Window features is very significant.

V. Conclusion

This paper introduced a new variant of the Close Enough Traveling Salesman Problem that considers Time Window constraint abbreviated as CETSP-TW variant. We provided the description of the CETSP-TW and formulated it as a Mixed Integer Nonlinear Program (MINLP) considering the Time Window constraints. Then, we proposed a three phased heuristic combining with Traveling Salesman Problem with Time Window (TSPTW) state of art algorithms and effective solver. The heuristic was able to solve the CETSP-TW variant efficiently in very suitable running times when tested on Dumas et al. [15] instances. Although the initial results appear favorable further researches might consist in adapting fast algorithms basically proposed for the TSP or the TSPTW minimization problems to the CETSP-TW context and generating a baseline test instances for comparison. Also, in order to get much optimized solutions a tradeoff must be made between runtime and solutions quality especially when the number of nodes is high.

References

[1] Y. Dumas, J. Desrosiers, E. Gelinas, and M. M. Solomon, “An optimal algorithm for the traveling salesman problem with time windows,” in Operations Research, vol. 43, no. 2, 1995, pp. 367-371.

[2] M. R. Garey, D. S. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness,” in the Mathematical Sciences, 1979, pp. x+338.

[3] DJ. Gulczynski and JW. Heath, “Close-Enough Traveling Salesman Problem: A Discussion of Several Heuristics,” Perspectives in Operations Research /Computer Science Interfaces Series, vol. 36. Springer, Boston, 2006.

[4] G. Gutin and A. P. Punnen, Eds., “The Traveling Salesman Problem and its Variations” Norwell, MA, USA: Kluwer, 2002.

[5] Z. Zhang, “Introduction to machine learning: k-nearest neighbors,” in Transl Med, vol. 4, no. 11, 2016, pp. 218.

[6] W. K. Mennell, “Heuristics for Solving Three Routing Problems: Close-enough Traveling Salesman Problem, Close-enough Vehicle Routing Problem, Sequence-dependent Team Orienteering Problem,” in university of Maryland, 2009.

[7] W. Mennell, B. Golden, E. Wasil, “A Steiner-Zone Heuristic for Solving the Close-Enough Traveling,” in C. 12th INFORMS Computing Society Conference: Operations Research. Monterey, California, 2011.

[8] Bo. Yuan and M. Orlowska. “On the optimal robot routing problem in wireless sensor networks,” in IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 9, September 2007, pp. 1252–1261.

[9] J. Silberholz and B. Golden. “The Generalized Traveling Salesman Problem: A New Genetic Algorithm,” Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies, vol. 37, pp. 165–181. Boston, MA, USA.

[10] B. Behdani and J. Smith, “An Integer-Programming-Based Approach to the Close-Enough Traveling,” in INFORMS Journal on Computing, vol. 26, no. 3, 2014, pp. 415-643.

[11] W.P. Countiho and R. Q, “A branch-and-bound algorithm for the close-enough traveling salesman problem,” in INFORMS Journal on Computing, vol. 28, no. 4, 2016, pp. 603-799.

[12] F .Carrabs, C. Cerrone, and R. Gaudioso,“A novel discretization scheme for the close enough traveling salesman problem,” in Computers & Operations Research, vol. 78, February 2017, pp. 163-171.

[13] Z. Yang, and X.-Q.-W.-L.-F.-L, “A double-loop hybrid algorithm for the traveling salesman problem with arbitrary neighborhoods,” in European Journal of Operational Research, vol. 265, no. 1, February 2018, pp. 65-80, 16.

[14] X. Wang, B. Golden, E. Wasil, “A Steiner Zone Variable Neighborhood Search Heuristic for the Close-Enough Traveling Salesman Problem,” in Computers & Operations Research, 2019.

[15] The TSPTW survey, https://homepages.dcc.ufmg.br/~rfsilva/tsptw/.

[16] Or-tools algorithms, https://developers.google.com/optimization/.