Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.00036

IMPROVED TABU SEARCH ALGORITHM FOR VEHICLE ROUTING PROBLEM WITH TIME WINDOWS

Do Duy Thanh[1], Nguyen Dinh Thuan[2], Nguyen Tan Phat[3], Nguyen Trong Nghia[4]

[1] University of Information Technology, VNU-HCM, Vietnam

duythanhdo.vn@gmail.com

ABSTRACT: *This paper presents the Metaheuristics algorithms, we called Improved Tabu Search algorithm (ITS). This algorithm addresses on Vehicle Routing Problem with Time Windows (VRPTW), one of NP-Hard problems. ITS algorithm is used to reduce number of vehicles and total travel distance in transportation. Authors introduce the improvement of well-known Tabu Search algorithm by 2-phases construction method and new move operator by combination relocation with reconstruction. The experiment based on the benchmark Solomon [1] with 56 cases yields some promising results.*

Keywords: *Improved Tabu Search, 2-phases construction, combination of relocation and reconstruction, diversification*.

I. INTRODUCTION

The Vehicle Routing Problem with Time Windows (VRPTW) is a problem when we have a given set of customers and fleet of vehicles so we have to find the optimized solution to serve each customer in individual time windows. In the existing literatures on the VRPTW, the number of vehicles to serve given customers is unlimited. So the first objective function is minimizing the number of used vehicles and second objective is minimizing distance.

There are many related works on VRPTW. Some early works by H. Pullen and M. Webb [2] introduced a solution which helps to schedule service shipping mail in Cental London in 1965, Madsen [3] adds time windows constraint in his solution solving scheduling of truck in 1976. But these solutions only applies simple heuristic. In 1999 Kohl is the first proposed an exact method Branch-and-Cut-and-Price for the VRPTW [4]. The branch-and-cut algorithm for the VRPTW was proposed first by Bard, Kontoravdis, and Yu [5]. In the following decade, there are many authors work on VRPTW and obtain many successful achievement as Lim and Zhang proposed Iterated Local Search to solve VRPTW with a limited number of vehicle in 2007 [6], Nagata and Bräysy have devised one of the most effective route minimization procedure in 2009 [7]. Another idea to solve the VRPTW is allowing infeasible solution by Cordeau, Laporte [8]. A. Haghani [16] used the genetic algorithm to resolve the pick-up or delivery vehicle routing problem with soft time windows with multi-vehicle with different capacities. V. Donati [17] introduced the other approach to resolve TDVRP by Ant colony algorithm with aim reducing the number of vehicles and total travel time. In 2012, F. Cordeau and M. Maischberger [18] used the parallel iterated local search framework with a combination of Tabu Search and perturbation mechanism to ensure the diversity search space. Recently,Phuong Khanh Nguyen [19] introduced Tabu search for time-depent multi-zone multi-trip vehicle routing problem with time windows (TMZT-VRPTW) which is extension of VRPTW with some new constraints such as: time-depent, multi-zone and multi-trip. In 2013, Jakub Nalepa [20] introduced a parallel memetic algorithm to solve the VRPTW with aims reducing the number of vehicles and total distance by combining two phases: using the parallel heuristic algorithm (PHA) for minimizing the number of vehicles, minimizing total distance traveled by using the parallel memetic algorithm (PMA). Ioannis Rogdakis [21] introduced a new efficient algorithm to resolve a real word VRP of a provisioning company in Crete in Greece in 2016. That is Island Memetic Algorithm (IMA), it's hybrid algorithm obtains by a combination of Greedy Randomized Adaptive Search Procedure (GRASP) and Iterated Local Search (ILS).

In this paper, we introduce some improvements with using Tabu Search for solving the VRPTW. The first step, we use 2-phases construction to initial the best solution. We also add a new move operator to Tabu Search, it's a combination of relocation and reconstruction. The secondary, we use an own new move on Tabu Search. It will minimize the number of vehicles and total distance at the same time.

II. OBJECTIVES AND CONSTRAINTS

In Vehicle Routing Problem with Time Windows, the constraints are:

(C1) Each customer is assigned in only one route.

(C2) Each vehicle cannot be loaded more than its capacity Q.

(C3) Each vehicle leaves depot location 0, after visiting the customer, it finished its route by returning the depot location n+1.

(C4) The point of time for delivery at each customer must be within the time window of this customer.

The Objectives of VRPTW will be prioritized depends on purpose of use. These objectives are:

(O1) Minimum number of used vehicles.

(O2) Minimum the total travel distance.

(O3) Minimum the waiting time for supplying all customers.

In our paper, we concentrate on objective (O1) and (O2).

III. PROBLEM FORMULATION

The VRPTW is defined on the directed graph G = (V, A), where V = {k1, k2…} is the set of homogenous vehicle. A is the set of location which has 0 as a depot location and n+1 as a returning location. C = A\{0,n+1} ={1, 2… n} is the set of customer's location.

Each arc (i, j) with i ≠ j is the travel path from customer i to customer j and it is associated with cij as travel cost and tij as travel time. Q is the vehicle capacity and di is represented as demand at customer i.

Time window [ei, li] at customer i defines the earliest time ei and latest time li which customer i must be served. This time window restricts the arriving time of vehicle at customer i must be before li. If the vehicle arrives earlier than ei, it must wait until ei for serving customer i with service time denoted by si. Tik is the starting point of service at customer i.

[e0, l0] and [en+1, ln+1] are the time window at depot and returning location. The vehicle must leave the depot location and return within e0 and ln+1.

VRPTW is presented as follow formulation:

(F1) $min \sum_{k\in V} \sum_{i\in A} \sum_{j\in A} c_{ij}\, x_{ijk} \quad i \neq j$

Subject to,

(F2) $\sum_{k\in V} \sum_{j\in A} x_{ijk} = 1 \;\; \forall i \in C$

(F3) $\sum_{i\in C} d_i \sum_{j\in A} x_{ijk} \;\leq Q \;\; \forall k \;\in V$

(F4) $\sum_{j\in A} x0_{jk} = 1, \;\; \forall k \;\in V$

(F5) $\sum_{i\in A} x_{ihk} - \sum_{j\in A} x_{hjk} = 0, \;\; \forall h \in C, \forall k \;\in V$

(F6) $\sum_{i\in A} x_{i,n+1,k} = 1, \;\; \forall k \in V$

(F7) $x_{ijk}\left(T_{ik} + s_i + t_{ij} - T_{jk}\right) \leq 0, \;\; \forall k \in V, \forall i,j \in A$

(F8) $e_i \leq T_{ik} \;\leq l_i, \;\; \forall k \in V, \forall i \in A$

(F9) $x_{ijk} \in \{0,1\}, \;\; \forall k \in V, \forall i,j \in A$

(F2), (F3) are mathematical expression of constraint (C1), (C2) respectively. (F4), (F5), (F6) are expression of constraint (C3). (F7) and (F8) are expression of constraint (C4). (F9) defines xijk as binary variable, it will be 1 if vehicle k travels from i to j, and 0 otherwise.

IV. IMPROVED TABU SEARCH SOLUTION

Base on the traditional Tabu Search of Fred Glover [9], we improved this Metaheuristics algorithm in each components. In particular, we enhance the initialize solution, suggest new operator for neighborhood generation, especially we applied the mining technique DCI closed proposed by Lucchese [10] as the intensification strategy.

4.1. ITS features

4.1.1. Initialized Solution

To initialize the solution (X), we suggest 2-phases construction method. This method is the modification of sequential insertion heuristics (I1) of Solomon [11].

We suggest small modification (I* heuristics), in which we firstly define the way how to choose the initialized route so-called seeded route. This seeded route has two options (β) once with first un-routed customer and the farthest distance, and second with first un-routed customer and max demand. After choosing the seeded route, the insertion happens like I1 heuristics.

We will have eight combinations of parameters (μ, λ, α1, α2, β) which they can yield the best construction result: (1, 1, 1, 0, 1), (1, 2, 1, 0, 1), (1, 1, 0, 1, 1), (1, 2, 0, 1, 1), (1, 1, 1, 0, 2), (1, 2, 1, 0, 2), (1, 1, 0, 1, 2), (1, 2, 0, 1, 2).

First phase, we run I* with eight combinations of parameters. The best one with smallest number of used vehicles and minimum total of distance will be selected.

Second phase, we consider each route of solution in first phase as a small problem and run I* again with eight cases of parameter for this route. The reason of two phases is that the better solution could be reached by sum of minimized routes.

4.1.2. Objective Function

Objective function in formula (F1) aims for reaching objective (O1) and (O2).

4.1.3. Neighborhood Generator

We implement 3 operations to generate the neighbor of current solution: relocation, 2 opt and Combination of relocation and reconstruction

a) Relocation Operation (Figure 1 and Figure 2)

Prosser and Shaw [12] suggest the inter relocation heuristic to relocate one customer from one to another route. Our implementation for this idea is: for each route (A) in solution, we try to reduce the number of customers in route by insertion sequent each delivery into randomly selected route (B).

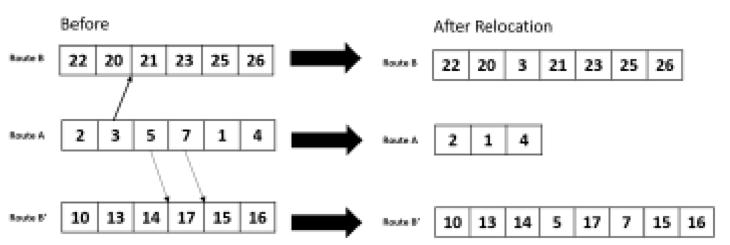

Figure 1. Inter relocation

Intra relocation operation idea is the same with Inter relocation operation, but this operation is applied for each route only.

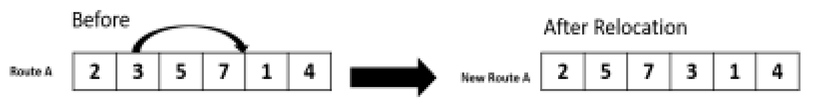

Figure 2. Intra relocation

b) 2-opt Operation

ITS implements 2-opt or 2-exchange idea as Bräysy and Gendreau [13]. In Figure 3, this operation is applied for two routes, in which we remove two arcs (i, i+1), (j, j+1) and replace with two other arces (i, j+1), (j, i+1).

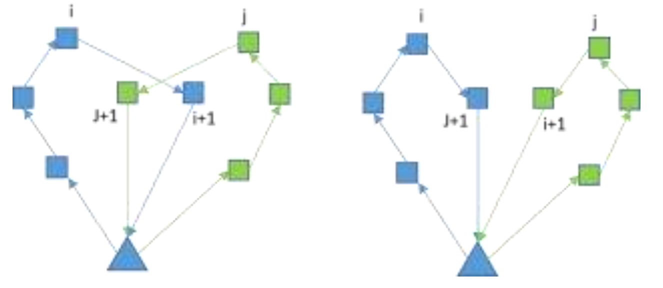

Figure 3. Inter 2-opt operation

c) Combination of Relocation and Reconstruction

Relocation and 2-opt operation sometime will be stuck in some circumstances because of the constraint of time window. Therefore, we cannot generate a new feasible move. To overcome these cases, we suggest the new heuristics which is combination of relocation and reconstruction heuristics (Figure 4).

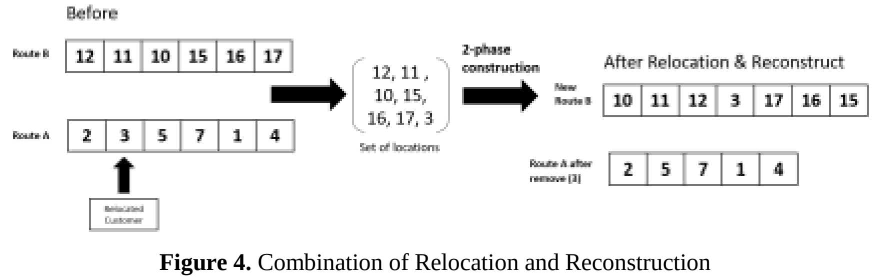

Figure 4. Combination of Relocation and Reconstruction

The basic idea is: we consider a set of locations from one location of route A and all locations of route B as a small instance of VRPTW, then we apply the 2-phase heuristics (section 4.1.1). New move is accepted if only from this small instance, just one feasible route is constructed.

4.1.4. Tabu List

Purpose of Tabu list is for prevention of turn back into the old searching space or old moves generated by neighborhood generator. Once generated moves are stored in tabu list, it will be remained during a number of iterations so-called Tabu tenure. We designed two kinds of Tabu list: one is the normal tabu list for storing the selected moves. One is mined Tabu list constructed after mining step. The mined Tabu list contains the nodes which have the max support index after mining. This mined Tabu list is designed as hard tabu list. It means that the elements in this list will be remained until the next mining point. Mining point is described in intensification strategy.

4.1.5. Aspiration Criteria

Aspiration criteria is for removing some of tabu moves which could gain the better solution. In our case, we design this aspiration criteria by checking if the current move could raise a smaller number of routes or total distance, we will remove it from the tabu list.

4.1.6. Intensification strategy

We apply mining technique DCI closed to explore the promised searching space. The idea is that: after a number of iterations so-called mining point, we get the list of best solutions. This list is considered as a light database for DCI closed. After mining, we select some best supported patterns, from them we construct the mined solution and the nodes from these patterns will be stored in the mined tabu list. The mined solution will be the initialized solution for the next iterations. The reason we choose DCI closed is that it constructs the frequent closed itemset which has a smaller size than frequent itemset. DCI closed algorithm has a better performance than others.

4.1.7. Diversification strategy

For diversification we implement the restart strategy Michel Gendreau [14] with some modifications. In particular, in each iteration we select the current solution with an associated operation (section 4.1.3) which populates the most number of feasible moves, then we store this current solution and operation into list so-called Promised List. After the number of iterations in which program doesn't find out the new best solution, one item in Promised List will be selected randomly and set as the current solution for next move generation.

4.2. ITS algorithm flow diagram

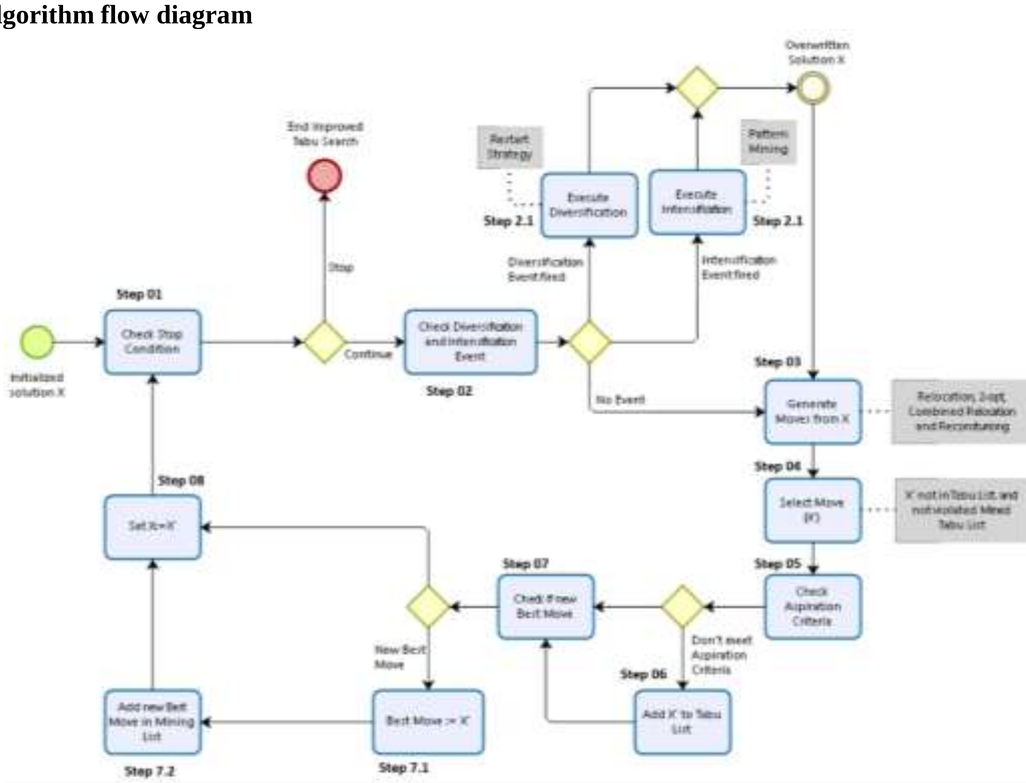

Figure 5. Improved Tabu Search flow diagram

Figure 5 describes in more detail of our ITS. It begins with initialized solution (X). ITS executes an exploring the searching space in number of iterations. In each iteration, ITS generates the neighbors of solution (X) (step 03) with operations mentioned in section 4.1.3. The move (neighbor of X) which not in Tabu list or mined Tabu list and don't meet the Aspiration Criteria will be added to Tabu list (step 06). If this move is better than the global best solution, it will be saved as global best solution (step 7.1) and stored in Light Database for mining function. Step 08 indicates that the new move will be considered as solution X in the next iteration. Step 02 is designed for plugins of diversification and intensification strategies.

V. COMPUTATION RESULT

5.1. Benmark instances

Our algorithm was tested on the well-known benchmark instances of Solomon[1]. The benchmark set includes 3 sets of 25, 50, 100 customers, with 56 instances in each set. There are 6 groups in each set C1, C2, R1, R2, RC1, and RC2. The C classes have the customers in clustered location. R classes, each customer have a random location. Especially the RC classes contain customer who lives in both clustered and random location.

5.2 Experimental settings

ITS algorithm was tested with nine different settings. Our parameters were set as follows : Tabu Search Iteration = 20000, resetIterations= {5000, 10000, 20000}, mining one time = 5000, mined node = 30, min sup = 90, Hard = 5000. Moreover we have some changeable parameters in each experimental test such as combine resetPoints = [{1000, 2000, 3000}, {1000, 2500, 3000}, {1000, 3500, 4000}] with Soft = [1000, 2000, 3000]. After combining all to work together we have 9 tests.

In the tables below, we show the experimental result of ITS (6th and 7th columns) compared with simple Tabu Search (4th and 5th columns) and the best results of Solomon benchmarking published in website[15] (2nd and 3rd columns). The gray cells in these tables are the best result of ITS.

When using simple Tabu Search, optimizing the number of vehicles and total distance is difficult in some instances. So we thought about modification some components of the simple Tabu Search algorithm to optimize more instance that did not reach global optimization as the best published. ITS has solved that. After using ITS we found the most noticeable change is that the number of vehicles was optimized and reached the best result. The experimental results have reflected that effectiveness of the ITS algorithm when compared to the simple Tabu Search.

In Table 1, ITS reaches the equivalent results compared with Solomon [15] in all instances. It indicates that ITS is very good for solving Clustering instance in both wide and narrow time windows. In Table 2 and Table 3, ITS yields some equivalent result in terms of number of vehilces such as RC103, RC104, R101, R201... In these tables also show that there is a trade-off between number of vehicles and total distance. However, the first optimized priority is number of vehicles, therefore we can not compare the total distance while our number of vehicles is still bigger than Solomon result [15]. Overal ITS gains 17/56 instances with equivalent results in both used vehicles and distance, and 30/56 instances with the equivalent number of used vehicles compared with Solomon [15].

Table 1. Result of problem sets C1 & C2

Instance	Vehicle best	Distance best	Vehicle TS	Distance TS	Vehicle ITS	Distance ITS
C101	10	828.94	10	828.94	10	828.94
C102	10	828.94	10	828.94	10	828.94
C103	10	828.06	10	828.06	10	828.06
C104	10	824.78	10	824.78	10	824.78
C105	10	828.94	10	828.94	10	828.94
C106	10	828.94	10	828.94	10	828.94
C107	10	828.94	10	828.94	10	828.94
C108	10	828.94	10	828.94	10	828.94
C109	10	828.94	10	828.94	10	828.94
C201	3	591.56	3	591.56	3	591.56
C202	3	591.56	3	591.56	3	591.56
C203	3	591.17	3	591.17	3	591.17
C204	3	590.6	4	634.8	3	590.6
C205	3	588.88	3	588.88	3	588.88
C206	3	588.49	3	588.49	3	588.49
C207	3	588.29	3	588.29	3	588.29
C208	3	588.32	3	588.32	3	588.32

Table 2. Result of problem sets RC1 & RC2

Instance	Vehicle best	Distance best	Vehicle TS	Distance TS	Vehicle ITS	Distance ITS
RC101	14	1696.95	15	1667.9	15	1669.15
RC102	12	1554.75	14	1503.82	13	1509.99
RC103	11	1261.67	12	1309.71	11	1320.49
RC104	10	1135.48	11	1161.72	10	1139.11
RC105	13	1629.44	15	1534.84	15	1540.94
RC106	11	1424.73	12	1393.71	13	1389.97
RC107	11	1230.48	11	1232.26	11	1239.59
RC108	10	1139.82	11	1141.29	11	1139.16
RC201	4	1406.94	4	1516.26	4	1515.94
RC202	3	1365.65	4	1317.15	4	1309.66
RC203	3	1049.62	4	1125.51	4	1047.63
RC204	3	798.46	3	877.37	3	850.83
RC205	4	1297.65	5	1501.98	5	1335.9
RC206	3	1146.32	4	1303.72	4	1171.9
RC207	3	1061.14	4	1066.66	4	1061.07
RC208	3	828.14	3	973.52	3	940.65

Table 3. Result of problem sets R1 & R2

Instance	Vehicle best	Distance best	Vehicle TS	Distance TS	Vehicle ITS	Distance ITS
R101	19	1650.8	19	1656.42	19	1655.88
R102	17	1486.12	18	1475.61	18	1479.03
R103	13	1292.68	14	1216.16	14	1224.01
R104	9	1007.31	10	991.27	10	986.53
R105	14	1377.11	15	1375.49	15	1367.11
R106	12	1252.03	13	1262.37	13	1253.36
R107	10	1104.66	11	1120.26	11	1081.38
R108	9	960.88	10	956.99	10	952.1
R109	11	1194.73	12	1164.81	12	1163.82
R110	10	1118.84	11	1107.18	11	1089.68
R111	10	1096.72	11	1073.82	12	1060.93
R112	9	982.14	10	963.7	10	961.9
R201	4	1252.37	4	1376.29	4	1317.38
R202	3	1191.7	4	1203.16	4	1202.38
R203	3	939.5	3	955.54	3	959.78
R204	2	825.52	3	798.47	3	799.89
R205	3	994.43	3	1086.16	3	1085.27
R206	3	906.14	3	985.61	3	992.36
R207	2	890.61	3	886.33	3	855.72
R208	2	726.82	3	787.04	3	774.17
R209	3	909.16	3	977.63	3	955.52
R210	3	939.37	3	1000.08	3	977.82
R211	2	885.71	3	949.55	3	875.21

Figure 6. Compare number of vehicles

Figure 7. Compare total distance

VI. CONCLUSION

ITS introduces new move operator and using mining technique as intensification. The experimental result show that ITS get a very good result in cluster instance (C1 and C2). Compare with best result [15], ITS need a lot of works in instances of group R and RC. However, mining technique with DCI Closed is really a new idea and could be enhanced to get a better result.

REFERENCES

[1]. M.Solomon, Algorithms for the vehicle routing and scheduling problem with time window constraints, Operations Research, 35 (1987), pp. 254-265.

[2]. H. Pullen, M. Webb, A computer application to a transport scheduling problem, Computer Journal, 10 (1967), pp. 10-13.

[3]. O. B. G. Madsen, Optimal scheduling of trucks - A routing problem with tight due times for delivery, in Optimization Applied to Transportation Systems, H. Strobel, R. Genser and M. Etschmaier, eds., IIASA, International Institute for Applied System Analysis, Laxenburgh, 1976, pp. 126-136.

[4]. N. Kohl, J. Desrosiers, O. B. G. Madsen, M. Solomon, and F. Soumis, 2 -Path cuts for the vehicle routing problem with time windows, Transportation Science, 33 (1999), pp. 101-116.

[5]. J. Bard, G. Kontorvdis, and G. Yu, A branch-and-cut procedure for the vehicle routing problem with time windows, Transportation Science, 36 (2002), pp. 250-269.

[6]. Lim and X. Zhang, A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows, INFORMS Journal on Computing, 19 (2007), pp. 443-457.

[7]. Y. Nagata and O.Bräysy, A powerful route minimization heuristic for the vehicle routing problem with time windows, Operations Research Letters, 37 (2009), pp. 333-338.

[8]. F. Cordeau, G. Laporte, and A. Mercier, Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows, Journal of the Operational Research Society, 55 (2004), pp. 542-546.

[9]. F. Glover, Future paths for integer programming and links to artificial intelligence, Computers & Operations Research, 13 (1986), pp. 533-549.

[10]. C. Clusshese, DCI Closed: a Fast and Memory Efficient Algorithm to Mine Frequent Closed Itemsets.

[11]. O. Bräysy, Michel Gendreau, Vehicle Routing Problem with Time Window, in Transportation Science Vol.39, 2005, pp. 107

[12]. Patrick Prosser, Paul Shaw, Study of Greedy Search with Multiple Improvement Heuristics for Vehicle Routing Problems.

[13]. O. Bräysy, M. Genddreau, Vehicle routing problem with time windows, part I: Route construction and local search algorithms, Transportation Science, 39 (2005), pp. 104-118.

[14]. M. Gendreau, An introduction to tabu search, in Handbook of Metaheuristics, Operations Research & Management Science, 57(2003), pp.37-54.

[15]. Transportation Optimization Portal - TOP https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/.

[16]. Haghani and S. Jung, A dynamic vehicle routing problem with timedependent travel times, Computers & Operations Research, 32 (2005), pp. 2959-2986.

[17]. V. Donati, R. Montemanni, N. Casagrande, E. RizzoliI, and M. Gambarella, Time dependent vehicle routing problem with a multi ant colony system, European Journal of Operational Research, 185 (2008), pp. 1174-1191.

[18]. F. Cordeau and M. Maischberger, A parallel iterated tabu search heuristic for vehicle routing problems, Computers & Operations Research, 39 (2012), pp.2033-2050.

[19]. Phuong Khanh Nguyen, Teodor Gabriel Crainic, Michel Toulouse, Tabu based local search for Time dependent Multi zone Multi trip Vehicle Routing Problem with Time Windows, European Journal of Operational Research, 231(2013), pp 43-56.

[20]. Jakub Nalepa, Zbigniew J. Czech, A Parallel Memetic Algorithm to Solve the Vehicle Routing Problem with Time Windows, Computer Science, 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2013, pp.144-151.

[21]. Ioannis Rogdakis, Magdalene Marinaki, Yannis Marinakis, and Athanasios Migdalas, An island memetic algorithm for real world vehicle routing problems, Operational Research in Business and Economics, pp 205-223.