Project Number 288094

eCOMPASS

e**CO**-friendly urban **M**ulti-modal route **P**l**A**nning **S**ervices for mobile u**S**ers

STREP
Funded by EC, INFSO-G4(ICT for Transport) under FP7

eCOMPASS – TR – 053

Eco-friendly Vehicle Routing via Balanced and Compact Clustering

Dimitrios Gkortsilas, Christos Zaroliagis

September 2014

Eco-friendly Vehicle Routing via Balanced and Compact Clustering*

Dimitrios Gkortsilas[1] and Christos Zaroliagis[1,2]

[1] Computer Technology Institute & Press "Diophantus", N. Kazantzaki str., University Campus of Patras, 26504 Rion, Greece
[2] Department of Computer Engineering & Informatics, University of Patras, 26504 Patras, Greece
{gkortsilas,zaro}@ceid.upatras.gr

4 September 2014

Abstract. We investigate the Vehicle Routing Problem with Time Windows (VRPTW) under an eco-friendly framework that demands the delivery of balanced and compact customer clusters. We present a new approach consisting of three major phases: (i) a first clustering of customers with compatible time windows; (ii) a second clustering of customers with close geographic proximity based on various methods (natural cuts, KaHIP, quad trees); (iii) a refinement phase that either splits a cluster into smaller ones, or merges clusters to form a bigger compact cluster. Our approach turns out to be beneficial when used in an on-line environment, where changes to the initial tour are requested (add a new customer to the tour or drop some customers). The new method serves as a warm starting point for re-evaluating and further optimizing the solution of VRPTW. Experiments with real data sets demonstrate that our approach compares favorably with standard approaches that start from a basic (cold) solution.

Keywords: Vehicle Routing Problem with Time Windows (VRPTW), geographic partition, compact and balanced clustering

Mathematics Subject Classifications: 05C85; 68T20; 68W05; 90B06; 90C59

1 Introduction

The Vehicle Routing Problem (VRP) [3] is one of the most known problems in computer science and operation research and has been studied extensively since when it was first introduced in 1959 [15]. Since then, many variations of the original problem have been introduced in the literature and there is a variety of methods to tackle the problem. The VRP is a generalization of the Traveling Salesman Problem (TSP), known to be NP-Hard, implying that it is

* This work was supported by the EU FP7/2007-2013 (DG CONNECT.H5-Smart Cities and Sustainability), under grant agreement no. 288094 (project eCOMPASS).

unlikely that exact solutions to real life instances of the VRP can be computed quickly. The most common ways of overcoming this hurdle is by using heuristics, metaheuristics, and approximation algorithms [15].

Many heuristics and metaheuristics have been used to solve variants of the VRP. The heuristics can be roughly classified into *construction* heuristics and *improvement* heuristics. As the name suggests, a construction heuristic is used to construct initial or candidate tours. These tours are then improved by an improvement heuristic. The classical construction heuristics are the savings-based method of [2] and the insertion heuristic [10]. Other methods, like the two phase method in [8], are also widely used. Among the improvement heuristics, the methods in [11] and [12] are well known and used.

The Vehicle Routing Problem with Time Windows (VRPTW) is a variation that captures real world applications, since each customer expects to be served during a specific time interval. Although the VRPTW has been extensively studied in the past (see e.g. [1, 2, 5, 8, 15]), little has been done, according to our knowledge, regarding the computation of eco-friendly routes or taking into account eco-friendly criteria. For example, a vehicle that performs a tour and is almost empty, wastes fuel and delivers less goods that it could if it was appropriately loaded. Furthermore, assigning a big truck to serve customers that are located in the city center may turn out to be a bad choice, since city centers have usually narrow streets and limited access. The choice of a route that uses the highway can be more eco-friendly than a route that uses local roads, although the latter may be shorter in distance. Towards providing more eco-friendly solutions, the industrial logistics sector has recently shown a growing interest for solving VRPTW by creating balanced and compact clusters. Customers are grouped together forming clusters that have the same total demand thus achieving balance. A vehicle must be able to serve all customers that belong to the same cluster thus achieving compactness. This approach provides fairness, since all drivers have the same workload to deal with and it is more eco-friendly since the clusters that are created have the property that all their customers are close together and have compatible time windows. Hence, a vehicle can serve them without wasting time going back and forth to the depot or travelling with low load.

The solution of VRP/VRPTW by providing balanced and compact clusters is a challenging problem and is partially addressed in [1, 5]. In [1] the authors suggest a modification of the well known k-means algorithm to create clusters of customers. In addition, they suggest a two-phase algorithm that creates connected and balanced clusters. The first phase is a construction phase and the second one is an improvement phase. The connectivity requirement is met by creating a spanning tree in each cluster and the balance requirement is met by maintaining roughly the same number of customers for each cluster. However, this approach does not take into account time windows. In [5], the authors take into account time windows. They suggest an algorithm consisting of three phases and introduce a Mixed Integer Linear Program (MILP) for solving VRPTW. Since the model can solve small instances (up to 25 customers) they suggest a clustering method. In Phase I feasible clusters are identified, in Phase II clusters are assigned to vehicles using the MILP. Phase III solves the VRPTW for every cluster created. For the creation of clusters they introduce a new heuristic clustering algorithm based on time windows. However, the approach in [5] does provide balanced clusters.

In this work, we provide a new heuristic approach for solving VRPTW, by delivering both compact and balanced clusters of customers, consisting of three phases. Firstly, it creates clusters of customers that have compatible time windows. Secondly, it creates clusters of customers that are geographically close together. After the clusters are created a third refinement phase takes place. During the refinement phase clusters are merged together, if this is feasible, or split further if they need to. For example, if there are two clusters with compatible time windows and are also geographically close, they are merged into one cluster. On the contrary, if there is a cluster that has customers that are geographically close but their time windows are not compatible it is split into smaller clusters. Our approach delivers a baseline solution which turns out to be rather robust when used in an online environment, e.g., when small changes happen to the (initially computed) tour. In particular, clusters allow us to fit an extra unplanned delivery easily within a cluster. Likewise, if a customer cancels his order, we find the cluster that this customer belongs to and drop him out of the tour. Hence, our approach is beneficial in a dynamic setting since it does not require to run it from scratch and it is the first one that solves VRPTW by providing both balanced and compact clusters.

The rest of the paper is organized as follows: in Section 2, we describe the basic definitions and notations of our model, in Section 3, we give an in depth description of our method, in Section 4, we provide the experimental study of the approach and in Section 5 we present our conclusions and give directions for future work. A preliminary version of this work appeared in [9].

2 Preliminaries

We begin by giving some definitions and basic notations that are necessary to present our model. A fleet of vehicles is available with a maximum capacity of Q units per vehicle. Every customer is associated with a *Time Window (TW)*, a period of time when the customer expects to be served. A time window of a customer i, is described by the *earliest arrival time* e_i and the *latest departure time* l_i that a vehicle can arrive and leave customer i respectively. Furthermore, every customer i is associated with a unique id, a demand d_i (the amount of units that must be delivered), the longitude $long_i$ and the latitude lat_i of his location. A $N \times N$ cost matrix $Cost$ is provided, where N denotes the total number of customers. Each entry $Cost_{ij}$ is a number representing the cost from traveling from customer i to customer j. The cost is usually expressed in distance (meters) or time (seconds).

The goal is to compute routes (paths that start from the depot and return to the depot) that minimize the cost and serve all customers without violating their time windows. An additional goal of our approach is to compute eco-friendly routes. This is achieved by constructing balanced and compact clusters. A *cluster* is a set of nodes that have similar properties. A cluster is called *compact* if all customers that belong to this cluster have compatible time windows. Two clusters are called *balanced* if each cluster has the same total demand. The ecological aspect is taken into account implicitly. The final clusters that are created have the property that all their customers are close together and have compatible time windows. Thus, a vehicle can serve them without wasting time going back and forth to the depot or travelling with low load.

3 The 3-Phase Approach

Our approach consists of three phases. Phase I is the Clustering with Time Windows Phase, where the customers are divided into clusters. The goal of Phase I is to create clusters with the following property: a vehicle serving a customer within a cluster can also serve all the other customers in the same cluster. In other words, each cluster forms a strongly connected component not in the real life instance but in a modified graph. The construction of the modified graph is explained in Section 3.1. Phase II is the Partition Phase, where the original graph is partitioned into cells. A *cell* is a group of customers that are geographically close. The main idea is that customers that belong to the same cell are geographically close to each other and they may belong to the same final cluster if their time windows are compatible. For the Partition Phase we use three techniques: quad trees [7], natural cuts [4] and KaHIP [13]. These techniques are presented in detail in Section 3.2. Phase III is the Merge & Split Phase, where the previously created clusters and cells are merged together or split in order to form the final clusters.

3.1 Phase I - Clustering Using Time Windows

For the representation of our model we use a *graph* $G = (V, E)$ where the set V represents vertices and the set E represents edges of the graph. A depot is represented as a special node denoted as v_0. Each customer i is represented as v_i, a vertex (node) of the graph and is associated with a time window $[e_i, l_i]$ where e_i is the earliest arrival time at customer i and l_i is the latest departure time from customer i. For two customers (nodes) v_i, v_j, variable t_{ij} denotes the traveling time needed to travel from customer i to customer j, and variable d_{ij} denotes the distance between nodes i and j in meters. An edge e_{ij} connects nodes i, j if $l_i + t_{ij} < l_j$ as shown in Figure 1. We note that t_{ij} takes also into account the required service time for customer j.

The inequality shows that when a vehicle serves a customer i and leaves at the customer's latest departure time, it can reach customer j taking into account the time needed to travel from i to j respecting customer's j latest departure time. After all edges have been created for all customers the process of creating the clusters begins. The main idea is to find the *Strongly Connected Components*

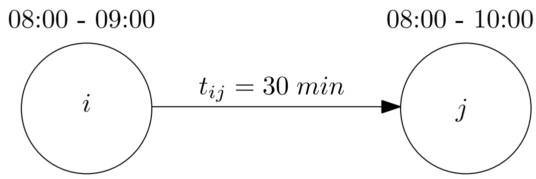

Fig. 1. Two customers with compatible time windows. If a vehicle leaves customer i it can then serve customer j.

(SCC) of the graph G. A Strongly Connected Component is a maximal subgraph H of G with the following property: for any two nodes $v_i, v_j \in H$ there is a path from v_i to v_j and also there is a path from v_j to v_i. Each strongly connected component k is then considered as a cluster C_k. For every strongly connected component the following property holds: node $v_i \in C_k$ is reachable from any other node v_j that belongs to the same cluster C_k.

After Phase I the compact clusters are created. Due to their construction all customers that belong to the same cluster have compatible time windows. This means that a vehicle can be used to serve all customers that belong to the same cluster.

3.2 Phase II - Clustering Using Geographic Proximity

The second phase makes a geographical partition of the area. We use three different techniques to achieve geographic partition: Quad Trees [7], KaHIP [13] and Natural Cuts [4]. Each technique tries to partition a given geographical area into smaller parts. We investigate all three of them in order to see which technique suites better for our problem.

Quad Trees. The first technique used is the Quad Trees [7]. Since we are dealing with instances where each customer is associated with coordinates (longitude, latitude) an instance can be represented on a map by its coordinates. Hence, given an area (usually urban) the main idea is to create a partition of M cells, where customers that belong to the same cell are geographically close to each other.

The algorithm that performs the partition is the following: given the four outermost points and some parameters describing the height h and the width w of each cell, as well as the depth d of the partition, the area is partitioned into $l = h \cdot w$ cells. This creates the first level of partition Level 0. Then, the process is repeated d times where d denotes the number of levels that needs to be created. The challenge is to find the best trade-off for the values of h, w, d. Since, for instance, we would like to avoid creating a few cells, because all nodes will be gathered there, and also avoid creating too many small cells as this will lead to many empty cells or cells that have 1 or 2 customers in them. This is a preprocessing step thus it can be executed off-line and does not create extra burden for the actual computation of the routes. An example of the Partition Phase can be seen in Figures 2 and 3. For simplicity the initial area is represented by a square although this may not be the case. To conclude, a cell corresponds to a geographical area and its size depends on its depth. For example, cells that belong to Level 0 correspond to a wider geographical area than cells that belong to Level 1. This can be seen on Figures 2 and 3 where cell 0 is divided into cells 00 through 08.

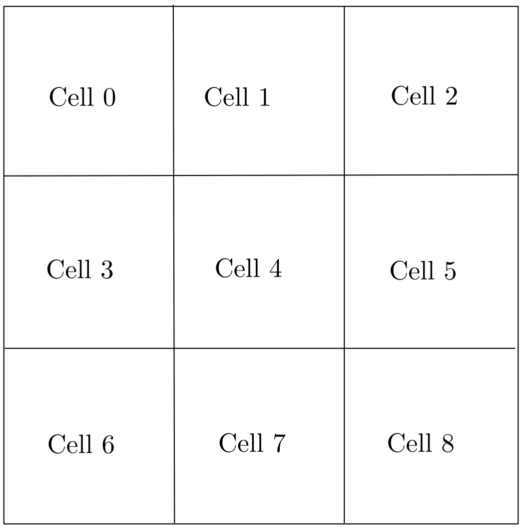

Fig. 2. First level of Geographic Partition. An area is divided into nine cells.

KaHIP (Karlsruhe HIgh Quality Partitioning). The second technique used is the KaHIP [13] partitioning software. KaHIP is a family of graph partitioning programs. It includes a multilevel partitioning algorithm called KaFFPa along with its variants Strong, Eco and Fast depending on what type of partition one is interested in. KaHIP uses max-flow/min-cut algorithms to create the desired partition. KaHIP needs as input a graph G to be partitioned (in a special form called DIMACS 10) and the number of blocks into which the graph will be partitioned. The user can also provide more arguments such as partition type (strong, eco, fast), time limit, and balance. KaHIP provides more algorithms that perform partition such as KaFFPaE (KaFFPaEvolutionary) which is a parallel evolutionary algorithm that uses KaFFPa to provide combine and mutation operations, as well as KaBaPE which extends the evolutionary algorithm. In our experiments we used the KaFFPa algorithm.

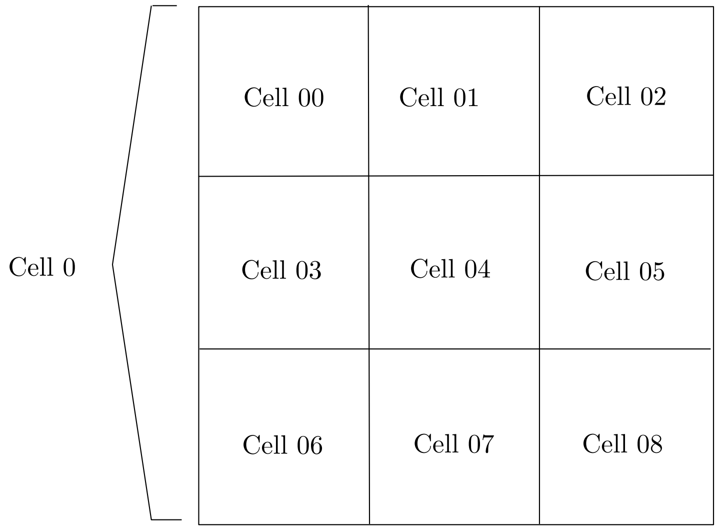

Fig. 3. Second level of Geographic Partition. Each cell from the first level is further divided into nine cells.

Natural Cuts. The third technique used was natural cuts [4]. This method consists of two phases. Firstly, it identifies and contracts dense regions of the graph by using a series of minimum-cut computations. The first phase reduces the size of the graph, while maintaining its general structure. This procedure is called the filtering phase. Secondly, it uses a combination of greedy and local search heuristics to create the final partition of the graph. The technique performs well on road networks, which have plenty of natural cuts such as bridges, mountain passes, ferries, rivers etc. Nevertheless, since it provides very good partitions, we also consider it in our study.

3.3 Phase III - Split and Merge

The third phase of our approach deals with the clusters and cells created in Phases I and II, respectively. Recall that Phase I creates clusters that achieve a first level of compactness and Phase II creates cells in order to get balanced routes. The main idea of Phase III is the refinement of the previous two phases in order to eliminate possible problems. For example, there may exist a cluster C where there is a path connecting any two customers but their time windows are incompatible, e.g. some have to be served in the morning and others in the afternoon. This cluster must be split into two (or more, if necessary) sub-clusters that will satisfy compactness (connectivity), and balance (each sub cluster must have the same total demand). Another case is that two cells created from Phase II can contain customers that are geographically close and they may have compatible time windows. In this case the two cells are merged to create a bigger cell that satisfies the properties of compactness and balance. Also, if there are empty cells from Phase II, they can be merged with their neighbouring cells. Then for each final cluster any heuristic or metaheuristic algorithm can be executed in order to compute the actual routes of the vehicles. The situation is depicted in Figures 4 and 5. In Figure 4, clusters C_1, C_2 are merged because they are both connected and geographically close, whereas in Figure 5 cluster C_3 is split into two sub-clusters because it contains customers that lay in different geographical areas.

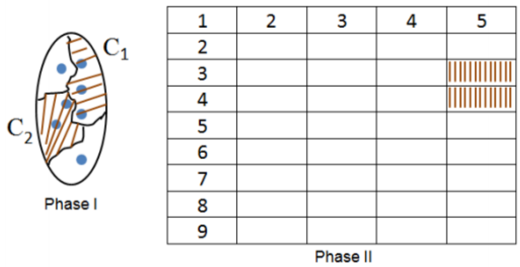

Fig. 4. Phase III: Cluster Refinement - Merge Operation. Examine phases I and II and perform a merge operation.

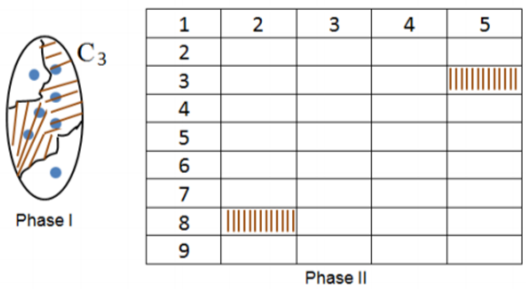

Fig. 5. Phase III: Cluster Refinement - Split Operation. Examine phases I and II and perform a split operation.

4 Experiments with real-world data

In order to evaluate the results of our approach, we conducted experiments with small and large size datasets kindly provided to us for scientific use in the frame of [6] by PTV.

The small dataset consists of two data sets of Munich one for parcel delivery and one for furniture delivery. The comparison of our method is done against PTV's Smartour [14] software using the baseline approach without optimization. The baseline approach is the customary starting point for all scenarios, which are subsequently further optimized. This is particularly true for dynamic scenarios where we need to compute a basic solution fast which is later optimized after the dynamic changes (addition of new customer orders or deletion of existing ones) are performed.

The quality measures that are reported are: total driving distance (in km), total driving time, number of vehicles used for each scenario, number of tours, CO_2 emissions and number of tour stops. We also report on three dynamic scenarios: the Incremental Scenario in which we add 3 customer orders, the Decremental Scenario in which we remove 2 customer orders and the Fully Dynamic Scenario in which there is a sequence of 8 insertions and 3 deletions. Both data sets provide the following information: total number of customers, a unique customer id, a location of each customer (longitude, latitude), one (or more) time window(s) of each customer, the weight of each customer (a number representing the amount of goods that have to be delivered) and a distance matrix with the real distance among all customers. The geographic partition method used for the Munich data sets was the quad trees.

The three large datasets concern the areas of Malta, Hamburg and Germany. Our first experiment focused on the different partition methods. Thus we do not consider time windows and evaluate the behaviour of the partition techniques in a small geographical area (Malta), in a medium geographical area (Hamburg) and in a large geographical area (Germany). The next set of experiments focused on time windows and on dynamic scenarios. We used all three partition techniques (quad trees, KaHIP, natural cuts) together with time windows to examine their behaviour in this case. For the dynamic case, we created three dynamic scenarios. The Incremental Scenario adds 20 additional customer orders with random coordinates to the initial dataset. The Decremental Scenario removes 10 customer orders from the initial dataset. In the Fully Dynamic Scenario there is a sequence of 20 insertions and 5 deletions of customer orders. All three datasets consist of 1000 customers.

In the rest of this section, we report our experimental findings.

4.1 Experiments with Small Datasets

Parcel Delivery. The tour planning results for the parcel courier express service providers are listed in Table 1. Without traffic information a total tour length of 163.32 km for serving 32 customer orders was computed. For the process of delivery one vehicle is needed for the generated tour.

	Baseline [14]	3-Phase Approach	Incremental	Decremental	Fully Dynamic
Total km driven	163.32	114.01	125.35	109.54	137.59
Total driving time	4h 12 min	3h 32min	3h 50min	3h 10min	3h 57min
CO_2 emissions	62.45kg	41.33kg	45.13kg	39.43kg	49.53
Total vehicles used	1	1	1	1	1
Number of tours	1	1	1	1	1
Tour stops	34	34	37	32	39

Table 1. Small Dataset: Performance indicators for the parcel delivery scenario. The vehicle type chosen for CO_2 emissions calculation was truck (7,5t).

As we can see from Table 1, our 3-Phase approach improves the total kilometres driven and the total driving time reducing also the CO_2 emissions. Part of the tour uses the motorway.

Furniture Delivery. The second scenario considered is the delivery of furniture, in particular kitchen furniture from a furniture store to various customers in the city center of Munich. The furniture store with its warehouse is located outside of Munich in the district of Taufkirchen. For this scenario it is assumed that the furniture can be ordered directly in the furniture store by the customer and every piece of furniture is available from the stock. Thus, a suitably short period of time between the point of order and delivery will be accepted. For simplicity the furniture store's warehouse is operating 24/7.

After the customers chose the pieces of furniture they wish to receive, the warehouse processes their orders and the delivery will be planned. As furniture is often bulky, the delivery process of the furniture is modeled as mid-size truck operations. We modeled the distribution process with two trucks and assumed 5 tons payload. Furthermore we assumed a service time of 15 minutes for a drop per truck stop for unloading the pieces of furniture. As a consequence, a vehicle is not immediately ready for use again after the point of delivery. After finishing the tours the trucks return to the furniture store/warehouse.

For the case of furniture delivery the calculations are based on a data set with 150 customer orders for a time period of about two weeks. The handled information are real, but made anonymous. For simplicity, the delivery time windows, in which customers can receive their furniture were standardised from 08:00 to 18:00 o'clock from Monday to Friday. The database contains 31 orders for each Monday. For the initial tour planning solution the results are shown in Table 2.

Based on the given information, without any traffic, three tours operated by two vehicles suffice to serve all 31 customers on Monday. As you can see from Table 2, the 3-Phase approach improves both the total kilometres driven and the total driving time by reducing also the CO_2 emissions. The tours generated do not use the motorway.

	Baseline [14]	3-Phase Approach	Incremental	Decremental	Fully Dynamic
Total km driven	204.36	103.15	119.32	94.78	129.36
Total driving time	4h 29min	2h 53min	3h 07min	2h 31min	3h 22min
CO_2 emissions	115.46kg	57.61kg	66.81kg	53.07kg	72.44kg
Total vehicles used	2	2	2	2	3
Number of tours	3	3	3	3	3
Tour stops	37	37	40	35	42

Table 2. Small Dataset: Performance indicators for the furniture delivery scenario. The vehicle type chosen for CO_2 emissions calculation was truck (7,5t).

4.2 Experiments with Large Datasets

The next step in our experimental evaluation was to consider larger datasets. There are three datasets: Malta, Hamburg and Germany. All datasets consist of 1000 customers. Malta as a small geographical area, Hamburg as a middle scale geographical area and Germany as a large geographical area.

Partitioning Methods. The aim of our first experiment is to focus on the behavior of the partition techniques on these geographical areas of varying size. We consider that every customer is available for delivery the whole day. Hence, their time windows are 24 hours long. The results are reported in Table 3.

	Malta		Hamburg		Germany		
	km	CO_2	km	CO_2	km	CO_2	Areas
Quad Trees	556	311.36	6917	3873.52	23852	13357.12	16
KaHIP	536	300.16	6703	3753.68	24072	13480.32	10
Natural cuts	479	268.24	6341	3550.86	23158	12968.00	10

Table 3. Performance indicators for large datasets. The numbers report total km driven and CO_2 emissions in kg for every partition method and areas of partition.

A first observation is that the natural cuts technique is more suitable for road networks. The KaHIP technique performs better than quad trees which were outperformed by the other two techniques. Another interesting observation is that for bigger geographical areas such as Hamburg and Germany the improvement in total distance is higher when the natural cuts technique is used. Since time windows are constant for each customer, the difference in the results lies in the method used for geographic partition.

Time Windows and Dynamic Scenarios. In our next experiment with large data sets, we take into account time windows. The time windows range is: 08:00 - 12:00, 12:00 - 16:00, 16:00 - 20:00 and 20:00 - 23:00. Each customer is assigned a time window at random. Then, we compute the new tours using our approach. The results are reported in Table 4. In order to make a comparison, it is customary (in the industrial logistics sector) to solve the problem without any constraint (time window, partition method, number of vehicles available). This will yield an ideal solution that will not be reached, as we add constraints such as time windows and number of available vehicles. This procedure is called free planning. Hence, we compare our approach against free planning to determine how far away we are from the free planning solution.

	Malta		Hamburg		Germany	
	km	Difference (%)	km	Difference (%)	km	Difference (%)
Quad Trees	587	24.6	7077	9.1	25264	9.0
KaHIP	575	22.0	7059	8.9	24933	8.5
Natural cuts	493	4.6	6866	5.9	25039	9.0
Free Planning	471	0.0	6481	0.0	22969	0.0

Table 4. Performance indicators for large datasets against free planning. The numbers report total distance driven in km. Difference shows how far away our approach is from the free planning case.

As expected, our solutions are inferior than those of free planning but they are away by a small margin. The presence of time windows makes the tours longer since they pose an additional constraint. Another observation is that the natural cuts method performs better for Hamburg and Malta datasets, while for Germany the best method is KaHIP.

One of the advantages of the suggested approach is that it can be used in an online environment. In real life, one or more customers may cancel their order unexpectedly. On the contrary, some new customers may appear that need to be served as soon as possible. The way many planners deal with such cases is to run a planning algorithm from scratch. In our case, since we have formed clusters we can easily check in which cluster to assign a new customer. The cancellation of an order is treated easily. We just remove the customer that canceled his order from the route. This will not affect the whole tour since all customers that belong to this route have compatible time windows.

In our experiments we used the datasets of Malta, Hamburg and Germany and we created three dynamic scenarios. The Incremental Scenario adds 20 additional customers with random coordinate orders to the initial dataset. The Decremental Scenario removes 10 customer orders from the initial dataset. In the Fully Dynamic Scenario there is a sequence of 20 insertions and 5 deletions of customer orders. The number of customer orders that were inserted/deleted was deliberately kept small since in real life cases a large number of new unexpected orders or cancellations is unlikely to happen. The results are shown in Table 5. We report total distance in km. As expected, the Incremental scenario computes routes that cover more total distance than the initial scenario. On the other hand, the Decremental scenario computes routes that cover less total distance than the initial scenario. The Fully Dynamic scenario computes routes that cover more total distance (by a small amount though) than the initial scenario, since there are more insertions to the tour than deletions.

	Initial	Incremental	Decremental	Fully Dynamic
Malta	479	492	459	487
Hamburg	6341	6367	6312	6358
Germany	23158	23204	23129	23179

Table 5. Initial datasets and dynamic scenarios are presented. The numbers report total distance driven in km.

To conclude, our experimental study shows that our 3-phase approach is beneficial in dynamic scenarios.

5 Conclusion & Future Work

In this paper we presented a new 3-phase approach for solving the VRPTW problem by providing compact and balanced clusters. The experiments that were conducted show that for road networks the natural cuts method is the most suitable for geographic partition. The 3-phase approach is flexible in the sense that should a better partition method is used the approach can adapt to it and create better clusters. Furthermore, the 3-phase method finds quickly a warm starting point in order to find a new solution for the online/dynamic version of the VRPTW problem. For future work, we will focus on other partition methods and heuristics for the VRPTW problem.

References

1. Buyang Cao and Fred Glover. Creating balanced and connected clusters to improve service delivery routes in logistics planning. *ISSN: 1004-3756*, DOI: 10.1007/s11518-010-5150-x, 2010.
2. G. Clarke and J. V. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:568–581, 1964.
3. G. B. Dantzig and R. H. Ramser. The Truck Dispatching Problem. *Management Science*, 6, 1959.
4. Daniel Delling, Andrew V. Goldberg, Ilya Razenshteyn, and Renato F. Werneck. Graph partitioning with natural cuts. In *In IPDPS. IEEE Computer Society*, 2011.
5. Rodolfo Dondo and Jaime Cerda. A cluster-based optimization approach for the multi-depot heterogeneous feet vehicle routing problem with time windows. *European Journal of Operational Research*, pages 1478–1507, 2007.
6. eCOMPASS Project. http://www.ecompass-project.eu/.
7. Raphael A. Finkel and Jon Louis Bentley. Quad trees: A data structure for retrieval on composite keys. *Acta Inf.*, 4:1–9, 1974.
8. M. L. Fischer and R. Jaikumar. A generalized assignment heuristic for the vehicle routing problem. *Networks*, 11:109 – 124, 1981.

9. Dimitrios Gkortsilas and Christos Zaroliagis. Eco-friendly vehicle routing via balanced and compact clustering. *3rd Workshop on Vehicle Routing and Logistics Optimization - VeRoLog*, 2014.
10. S. R. Jameson and R. H. Mole. A sequential route building algorithm employing a generalized saving criteria. *Operational Research Quarterly*, 27:503–511, 1976.
11. B. W. Kernighan and S. Lin. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498 – 516, 1973.
12. G. A. P. Kindervater and M. W. P. Savelsbergh. Vehicle routing: Handling edge exchanges. *In E. H. L. Aarts and J. K. Lenstra, editors, Local Search in Combinatorial Optimization*, pages 337 – 360, 1997.
13. Peter Sanders and Christian Schulz. Think Locally, Act Globally: Highly Balanced Graph Partitioning. In *Proceedings of the 12th International Symposium on Experimental Algorithms (SEA'13)*, volume 7933 of *LNCS*, pages 164–175. Springer, 2013.
14. PTV Smartour. http://smartour.ptvgroup.com/en-uk/home/.
15. Paolo Toth and Daniele Vigo, editors. *The Vehicle Routing Problem.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.