

# Nature of Real-World Multi-objective Vehicle Routing with Evolutionary Algorithms

Juan Castro-Gutierrez, Dario Landa-Silva  
ASAP Research Group, School of Computer Science  
University of Nottingham, UK  
jpc@cs.nott.ac.uk, dario.landasilva@nottingham.ac.uk

José Moreno Pérez  
Dpto. de Estadística, I.O. y Computación  
Universidad de La Laguna, Spain  
jamoreno@ull.es

**Abstract**—The Vehicle Routing Problem with Time Windows (VRPTW) is an important logistics problem which in the real-world appears to be multi-objective. Most research in this area has been carried out using classic datasets designed for the single-objective case, like the well-known Solomon’s problem instances. Some unrealistic assumptions are usually made when using these datasets in the multi-objective case (e.g. assuming that one unit of travel time corresponds to one unit of travel distance). Additionally, there is no common VRPTW multi-objective oriented framework to compare the performance of algorithms because different implementations in the literature tackle different sets of objectives. In this work, we investigate the conflicting (or not) nature of various objectives in the VRPTW and show that some of the classic test instances are not suitable for conducting a proper multi-objective study. The insights of this study have led us to generate some problem instances using data from a real-world distribution company. Experiments in these new dataset using a standard evolutionary algorithm (NSGA-II) show stronger evidence of multi-objective features. Our contribution focuses on achieving a better understanding about the multi-objective nature of the VRPTW, in particular the conflicting relationships between 5 objectives: number of vehicles, total travel distance, makespan, total waiting time, and total delay time.

**Index Terms**—Multi-Objective Optimisation, Vehicle Routing Problem with Time Windows, Combinatorial Optimisation, Benchmark Datasets

## I. INTRODUCTION

The VRPTW consists of creating the set of routes to serve a number of customers with a fleet of vehicles that depart from a central depot (with unlimited capacity). Each vehicle within the fleet has a maximum capacity that must not be exceeded. Customers have a time window in which they must be served. Also, each customer has an associated service time, which is the actual time taken for serving the customer once the vehicle arrives at the customer’s location. Classic objectives when solving the (single-objective) VRPTW include: minimising the number of vehicles (fleet size), minimising the total travel time, minimising the total waiting time, etc. In recent years there has been much interest in tackling this problem in a multi-objective fashion. To the best of our knowledge, there are no public problem instances for the multi-objective version of the problem (MOVRPTW). Most related work in the literature uses either non-public scenario-specific small instances or problem instances artificially extended from classic single-objective datasets. Usually, such classic problem

instances do not provide the travel times between each pair of customers. Therefore, it is commonly assumed that one unit of travel distance corresponds exactly to one unit of travel time, which might not be realistic, particularly when Euclidean distance is given instead of on-the-road distance. As a result of this type of assumptions, the assessment of multi-objective algorithms applied to tackle the MOVRPTW might not be entirely accurate.

The Solomon’s dataset [11] is perhaps the most widely used VRPTW dataset in the literature. It consists of 56 instances with 100 customers. According to their geographical distribution, the dataset is divided into three subsets: *CXXX* (clustered) 17 instances, *RXXX* (uniformly distributed) 23 instances and *RCXXX* (mixed) 16 instances. Each of these three subsets consists of two subsets. The clustered instances are in subsets: *C1XXX* and *C2XXX*. The subset *C1XXX* has a different layout for customer locations and narrower time windows than subset *C2XXX*. For both *RXXX* and *RCXXX*, their two subsets share the same layout, but customers in the first subset have narrower time windows like in *CXXX*. The service time for customers is 90 time units for subsets *CXXX* and *RCXXX*, and 10 time units for subset *RXXX*. The demand varies depending on the customer and instance, but they take discrete values in  $\{10, 20, 30, 40\}$ . More information about other datasets and their characteristics can be found in [4].

Most published research on the multi-objective VRPTW uses the Solomon’s dataset. An issue with this dataset is that it relies on Euclidean distance for both travel distance and travel time. This is hardly a realistic scenario because it is often the case that travel time is not directly proportional to travel distance. In this work, we conduct experiments to show that the Solomon’s problem instances are perhaps not entirely adequate to investigate the multi-objective VRPTW. This is because the correlation between different objectives is weak which means that there is little interaction (conflict and harmony) between objectives when searching for solutions.

We have been working with a set of multi-objective VRPTW problem instances based on data from a distribution company in Tenerife, Spain. The company delivers food products and serves more than 1000 customers overall, with around 150 customers being served each day. Realistic data for the travel distance and travel time between each pair

of customers was obtained using Google maps. Moreover, time windows specifications were generated according to some information provided by the company. Demand specifications were established using a number of parameters in order to present different scenarios. Contrary to most of the benchmark datasets listed above, the distance and time matrices are distinct and non-symmetric, hence representing a realistic trade-off between travel distance and travel time. For example, for pairs of customers located within an urban area, travel time is high compared to the corresponding distance, reflecting the fact that travelling in urban areas is more time consuming than travelling in rural areas. Contrary to what happens with Solomon’s datasets, our experiments show that there is a much clearer interaction between different objectives in our problem instances. This is potentially a very good thing because the multi-objective nature of this important logistic problem can be better investigated using these realistic datasets. Also, multi-objective algorithms could be tested in a more accurate manner if truly multi-objective test instances are available. As an additional contribution of this paper, we make our problem instances available to the research community as a contribution towards a common framework that allows more research on the multi-objective VRPTW. An open source generator to create different problem instances and more information is available at [2]. We incorporate 5 minimisation objectives in the proposed benchmark datasets in order to assess the quality of a route-plan: *number of vehicles* (denoted as Z1) needed to serve all customers, *total travel distance* (Z2), *makespan* (Z3) or travel time of the longest route (from/to depot), *total waiting time* (Z4), and *total delay time* (Z5).

The analysis conducted in this paper to compare the multi-objective nature of Solomon’s and our dataset is based on the work by Purshouse and Fleming [10]. They indicate that three main relationships may occur between pairs of objectives: *conflict*, *harmony* or *independence*. If there is a dependence between the objectives, they can be *conflicting* (if it is not possible to improve one without worsening the other) or *harmonious* (the improvement in one witnesses an enhancement in the other). Conversely, if the optimisation of one objective does not affect the other, the relationship is of *independence*. When conducting multi-objective optimisation benchmarking, the most important relationships are that of dependence. Multi-objective problems with independent relationships among their objectives can be addressed by decomposing the problem into sub-problems [10]. However, dependent relationships present a real challenge to multi-objective algorithms. In general, the more conflicting objectives exist in a given problem, the more suitable this problem is for benchmarking. Purshouse and Fleming [10] also state the importance of keeping harmonious objectives in the optimisation process because, for example, this might provide additional knowledge to the decision maker. We conducted our study using a well known multi-objective genetic algorithm (NSGA-II) in order to make pair-wise comparison between all combinations of objectives.

The rest of this paper is organised as follows. Section II gives essential concepts related to this work. In Section III, the

main characteristics of the dataset are presented. In Section IV, we describe our experiments and discuss the results in section V. Conclusions and further work is stated in Section VI.

## II. BACKGROUND

### A. Multi-objective Optimisation (MOO)

MOO is the process of simultaneously optimising two or more conflicting objectives subject to a number of constraints. In mathematical terms, a multi-objective optimisation problem (MOP) can be written, without loss of generality, as  $\min f(x) = (f_1(x), f_2(x), \dots, f_p(x))$  subject to  $x \in X \subseteq \mathbb{R}^n$ , where  $X$  is a constraint set in the multi-dimensional space of the problem specified by  $X = \{x \in \mathbb{R}^n : g_i^{(x)} \leq 0, i = 1, \dots, m; h_j^{(x)} = 0, j = 1, \dots, l\}$ .

Given two feasible solutions  $x$  and  $y$ , we say that  $x$  dominates  $y$ , if  $\forall i : f_i(x) \leq f_i(y)$  and  $\exists j : f_j(x) < f_j(y)$ . Moreover,  $x$  is said to be *Pareto Optimal* if it is not dominated by any other feasible solution. Then, the aim is to find the set of *Pareto Optimal* solutions usually called *Pareto Set*. This set contains a number of *non-dominated points in the objective space* creating the so-called *Pareto Frontier*.

### B. Vehicle Routing Problem with Time Windows (VRPTW)

The VRPTW is defined on a graph  $G(N, A)$  where  $N$  is the set of nodes representing a set of customers, denoted by  $C$ , enumerated 1 to  $n$ , and 0 representing the depot. All routes must start and end at node 0. Each customer has a demand  $p_i, i \in C$ . The customers must be served just once using a set of  $m$  identical vehicles, denoted by  $V$ , with each vehicle having maximum capacity  $Q$ . The arc set  $A$  denotes all possible connections among the nodes. Each arc has a cost which usually represents the distance or travel time from customer  $i$  to customer  $j$  and it is denoted by  $d_{ij}$  or  $t_{ij}$ , respectively. Each customer has a time window  $[a_i, b_i]$  during which service at this customer may start. In case of the vehicle arriving earlier, it must wait until the time window opens. On the other hand, arriving after the time window is usually not allowed. However, in this implementation, we consider the violation of the upper limit of time windows as a soft constraint. Once a vehicle has arrived at the customer’s site, the delivery incurs in a delay known as service time

A large number of objectives have been used in the VRPTW literature over the years. Some of the most common objectives used are the minimisation of: the number of vehicles, the total travel distance, the total travel time, and the maximisation of: the quality of service (which might be interpreted as reducing delay times), the drivers satisfaction (minimise waiting times and makespan), etc. We have chosen to minimise five objectives commonly used across different datasets in the literature: *number of vehicles* (Z1), *total travel distance* (Z2), *makespan* (Z3), *total waiting time (in case of early arrival)* (Z4), and *total delay time* (Z5). For an overview on the most commonly used objectives for multi-objective vehicle routing problems in the literature, see [7].

### C. Non-dominated Sorting Genetic Algorithm (NSGA-II)

NSGA-II [3] has been used before to deal with the VRP. For example, Murata and Itai [9] applied this algorithm on a VRP with two different types of demands. They focused the study on the optimisation of *number of vehicles* and *makespan* considering the similarity of sets of non-dominated solutions. Jozefowicz et al. [6] presented an improved NSGA-II with two enhancements: (1) parallelisation using an island model and (2) an elitist diversification technique. In this approach, the objectives tackled were to minimise *travel distance* and *the difference of length between the longest and shortest route*. Wei et al. [13] carried out research on a real-world test case in Michigan (USA) using NSGA-II. In this study, the authors considered the minimisation of three objectives: *travel distance*, *distribution time* and *number of vehicles*. Garcia-Najera and Bullinaria [5] used NSGA-II to compare the performance of their algorithm, a MOEA that incorporates two similarity measures: Jaccard similarity coefficient and Edit distance. The performance was assessed using the *travel distance* and *number of vehicles*.

## III. MOVPTW DATASET

In this Section, we describe the characteristics of our benchmark dataset. We consider the characterisation of the time windows and the customers' demands. Finally, we specify which parameters were used to generate the proposed dataset.

### A. Characterisation of Time Windows

The time window specifications for each customer have been designed to imitate what the delivery company faces everyday. The opening time of the depot is 8 hours. In the first time windows profile (Profile 1 - Fig. 1), all customers are available all day (8 hours = 480 minutes). In the following three profiles, we distinguish three types of customers, early customers (those who want to be served in the morning), midday customers (those who want to be served at midday) and late customers (those who want to be served the latest). In order to cover the whole day with these three types of customers, we create time windows with a length of 480 minutes / 3 types of customers = 160 minutes/type of customer, as seen in Profile 2 - Fig. 1. In this profile, early customers will be served in the time window [0, 160] minutes. Midday customers will be served in the time window [160, 320] minutes. Late customers will be served in the time window [320, 480] minutes. For the third and fourth profiles, we follow a similar approach. However, we decrease the length of each time window by 30 and 60 minutes respectively. Thus, in the third profile (Profile 3 - Fig. 1), a time window of 160 minutes length is decreased to 130 minutes. So, the opening hours will be: early customers [0, 130], midday customers [175, 305] and late customers [350, 480]. Time windows of midday customers are symmetric with respect to the midday. For example, if the length of a time window is 100, it will grow 50 to the left and 50 to the right respect to the central point 240 (4th hour). Similarly, in the fourth profile (Profile 4 - Fig. 1), we consider time windows of 100 minutes. Therefore, the time windows

will be: early customers [0, 100], midday customers [190, 290], and late customers [380, 480]. In a fifth time windows profile (Profile 5), customers are associated with one of the 10 time windows types contained in the previous profiles (see Fig. 1). That is, Profile 5 will contain the only time window type of the Profile 1, three time window types of Profiles 2, 3 and 4; which in total sums up 10.

That is, we suggest 5 time windows profiles for the dataset. Each profile, except the first one, has three time windows. Given a time window profile, a customer has the same probability of having any of the time windows within that profile.

### B. Characterisation of Demands

Let  $p_i$  be the demand of customer  $i$  and  $Q$  the maximum capacity for each vehicle. This capacity cannot be less than the maximum demand among the customers  $\underline{D}$ , otherwise some customers might not be served. Moreover, the natural upper bound on the capacity of vehicles is the sum of all customers' demands. But in this case a single vehicle could serve all customers (if we do not take into account the time windows) and this is also not quite adequate. Therefore, the capacity of each vehicle  $Q$  is bounded by  $\underline{D} = \max_i \{p_i\} \leq Q \leq \overline{D} = \sum_{i=1}^n p_i$ , where  $n$  is the number of customers. The closer  $Q$  is to its lower bound, the more constraint each vehicle will be with respect to the total demand. Conversely, the larger is  $Q$ , the more spare capacity the fleet will have.

Parameter  $\delta$  (delta) is used to modulate the slack margin of an instance,  $Q = \underline{D} + \delta/100(\overline{D} - \underline{D})$ , where  $\delta \in [0, 100]$ . If  $\delta$  takes values close to 0, the capacity of the vehicle  $Q$  will be very limited. On the other hand, for large values of  $\delta$ , the vehicles will have a capacity  $Q$  close to the total demand.

Regarding the fleet size, since we work with soft time windows, the maximum number of vehicles needed is equal to the total customers' demand divided by the vehicle capacity. However, since  $Q \geq \underline{D}$ , an upper bound for the size of the fleet would be:  $m \leq \overline{D}/\underline{D}$ .

### C. Dataset Settings

Based on the guidelines mentioned in this section, we have created our benchmark dataset using the following combinations: (1) Number of Customers: {50, 150, 250}, (2) Time Windows: Profiles {1, 2, 3, 4, 5}, within each profile, the same probability was assigned to each time window type. For example, a probability of 1/3 was assigned to each time window type in Profiles {2, 3, 4}. While a probability of 0.10 was used in Profile 5 for each type, since it has 10 time window types. (3) Customer Demand: three types {10, 20, 30}, each with probability 1/3 and three types of  $\delta = \{60, 20, 5\}$ . (4) Service Times were set to 10, 20 and 30 minutes, each with probability 1/3. (5) Seeds were all fixed to 0.

Summing up, a total of 45 MOVPTW instances were generated (3 sizes \* 5 time windows profiles \* 3 deltas). This dataset and the configuration files can be download from [2].

## IV. EXPERIMENTAL DESIGN

In order to study the pair-wise relationships among the objectives stated in Sect. II-B for both datasets, we conducted

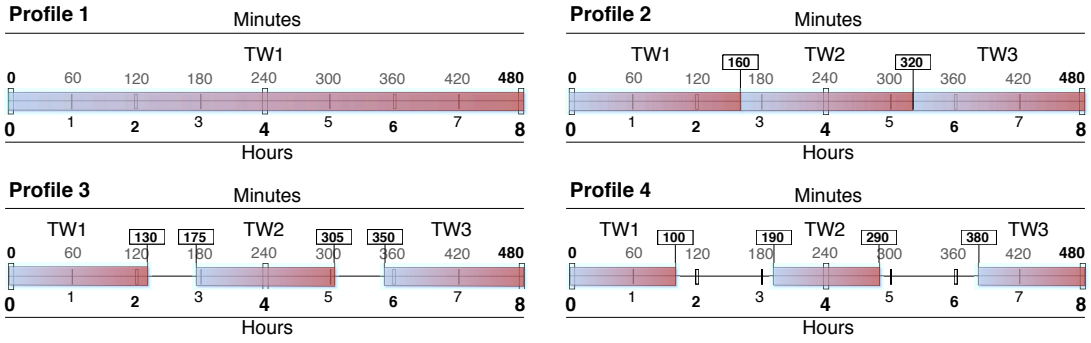


Fig. 1. Four of the time windows profiles. This figure shows the opening times and closing times for each profile.

a number of experiments using NSGA-II (Sect. II-C). This development was not carried out from scratch, we used the implementation of an Evolutionary Algorithm (EA) for the VRPTW [8] as a starting point. This implementation is based on the optimisation framework ParadisEO-MOEO [1]. We extended this implementation to support multiple objectives (Sect. II), process our dataset and use the NSGA-II.

In this implementation, the encoding of an individual is a list of routes (a list of lists). Each element in these lists represent a customer and their positions within each list specify the turn in which they will be served.

The population is initialised using a constructive method that aims at satisfying first the customers farthest from the depot. After the initialisation process, all individuals are evaluated. The fitness assignment procedure of NSGA-II is called non-dominated sorting criterion [3]. It consists of dividing the population into non-dominated fronts. This way, the fitness of an individual depends on the depth of its front.

Once the individuals are evaluated, a sub-group is selected for crossover. This process recombines two parents (solutions) with certain probability  $\gamma$ , creating one or two offsprings (new solutions). In this implementation, NSGA-II has 3 standard crossover operators: (1) One-point crossover, (2) Edge crossover and (3) Generic crossover.

In the One-point crossover, a random number of consecutive customers are copied from one parent to another, removing duplicates. The Edge crossover consists of (1) constructing new intermediate solutions by joining edges from both route-plans (parents) and (2) merging sub-tours creating feasible solutions. In the Generic crossover, an entire route is copied from one route-plan to another, removing duplicates.

In order to promote diversity within the population, the offspring solutions go through a mutation operation with certain probability  $\nu$ . NSGA-II was run using four basic mutation operators (Swap, Insertion, Inversion and Displacement).

The Swap mutation interchanges the position of two customers within a route-plan. The Insertion mutation consists of moving a random customer to a new position within the route-plan. In the Inversion mutation, customers in a portion of the route-plan are reversed. The Displacement mutation is a generalisation of the insertion mutation in which a number of consecutive customers are moved.

In order to re-use the genetic operators of the previous implementation, route-plans were forced to be feasible in terms of capacity constraints. This process was carried out by splitting routes in which the vehicle capacity was exceeded. In terms of time windows, a maximum delay of 30 minutes was allowed per customer (if this limit is violated, routes are also split). This is the maximum delay the company is willing to assume per customer to provide a good quality of service.

In the experiments, NSGA-II evolved a population of 50 individuals for 10000 generations. We compared Solomon's 100 customer dataset against our dataset with 150 customers. The algorithm is applied to each dataset 10 times (repetitions) with the same parameters and seeds. This new implementation is open source and is available at [2].

## V. DISCUSSION OF RESULTS

### A. Correlation Between Objectives

Following the platform for the treatment of large number of criteria by Purshouse and Fleming [10], we treated the results using scatterplot matrices. An example of this matrix is shown in Fig. 2. The objectives  $Z1 \sim 5$  are shown in the main diagonal of the matrix ( where  $Z1$  is the *number of vehicles*,  $Z2$  is the *travel distance*,  $Z3$  is the *makespan*,  $Z4$  is the *waiting time* and  $Z5$  is the *delay time*). Each element in the matrix below the main diagonal shows a pair-wise comparison of objectives. Above the main diagonal, a correlation value is associated to each comparison. The closer this value is to 1 or  $-1$ , the larger is the font type, and the stronger is the dependence relationship between the corresponding two objectives. Since all objectives are to be minimised, a positive correlation value indicates a harmonious relationship. The opposite occurs when the correlation values are negative. Figure 2 shows the scatterplot matrices of the Solomon's instance  $R101$  (on the left) and our instance  $d2.tw2$  for 150 customers (on the right). At first sight, the correlation values seem much larger for  $d2.tw2$ . An example of harmonious behaviour is seen in the instance  $d2.tw2$  when comparing  $Z1$  vs.  $Z4$ . Its correlation value of 0.98 indicates that, as we decrease the *number of vehicles* ( $Z1$ ), the *waiting time* ( $Z4$ ) for each of them gets shorter, which is logical. In the same instance, an example of conflict arises comparing  $Z1$  vs.  $Z5$ . Here, the correlation value of  $-0.96$  means that by

decreasing the number of vehicles  $Z1$ , the delay time  $Z5$  gets longer and vice-versa. In the Solomon's  $R101$ , the pair-wise dependence relationships between objectives appears to be weak. For example,  $Z1$  seems not to be related to any of the other objective under study. The bi-criterion shape sections of the trade-off surface are plotted below the main diagonal for both instances. It is clearly seen that  $d2.tw2$  presents more uniform shapes than Solomon's  $R101$ . It is also worth noting that  $d2.tw2$  has a much larger number of non-dominate solutions than  $R101$ . In this particular case,  $d2.tw2$  has 3784 non-dominated solutions against the 394 of  $R101$ .

In the rest of this section, we will discuss the rest of our experiment results based on average correlation values. We first present a subsection with results for the Solomon's instances and then another with results for the MOVRPTW instances with 150 customers.

### B. Solomon's Dataset

We present three tables corresponding to each type of Solomon's 100 customers dataset  $\{CXXX, RXXX, RCXXX\}$  (Tables I, II and III). Each table shows in its first column the name of the instance. Each column represents a pair-wise comparison. For example,  $Z1-Z2$  compares the *number of vehicles* against the *travel distance*. Correlation values vary from  $-1$  (conflict) to  $1$  (harmony). Those correlations values equal to NA indicate that one or both objectives have the same value across all the non-dominated solutions found. All values are averaged over the 10 runs.

Table I shows the average correlation values for all instances with customers in clusters. This subset of instances can be divided into two new subsets:  $C1XX$  and  $C2XX$ . The former of these subsets presents the customers in well defined clusters and their time windows are narrow. On the other hand, the latter has a more relaxed geographical distribution of the customers and the time windows are wider. This explains why in the case of the  $C1XX$  subset, the correlation values are so low. The first block of the table shows, as Tan et al. [12] and Garcia-Najera and Bullinaria [5] previously stated, that no conflict exists between *number of vehicles* ( $Z1$ ) and *travel distance* ( $Z2$ ). However, our results indicate that no dependency holds for any objective against  $Z1$  in almost all the instances in  $C1XX$ . The largest values for this subset seem to appear in the comparison of *travel time* ( $Z2$ ) against *makespan* ( $Z3$ ). Conversely, in the second block of the table, the results for the subset  $C2XX$  seem to indicate clear pair-wise dependency relationships. In this subset, the most conflicting objectives are *number of vehicles* ( $Z1$ ) against *delay time* ( $Z5$ ). And it is worth noting that in this set, the improvement of *makespan* ( $Z3$ ) is in harmony with the reduction of *delay time* ( $Z5$ ).

In Table II, we show the average correlation results for the randomly spread customers. This subset is also divided into two subsets:  $R1XX$  and  $R2XX$ . In this case, both share the same geographical distribution of their customers. However,  $R1XX$  has narrower time windows than  $R2XX$ . In the first block of the table corresponding to  $R1XX$  subset, we appreciate a similar behaviour as in  $C1XX$ . That is, the

*number of vehicles* seems not to have a pair-wise dependence relationship to any other objective under this study. The only considerable conflict relationship arises in  $R103$ ,  $R107$  and  $R110$  comparing *travel distance* ( $Z2$ ) against *waiting time* ( $Z4$ ). In the second block of the table, we find the results for the  $R2XX$  subset. In a similar fashion to the  $CXXX$  instances, the subset  $R2XX$  presents better values than  $R1XX$ . However, the average correlation values for  $R2XX$  does not seem to be as good as for  $C2XX$ . According to the results, the most conflicting relationship is *makespan* ( $Z3$ ) versus *waiting time* ( $Z4$ ). Additionally, it is important to highlight that in the pair-wise comparison of *makespan* ( $Z3$ ) versus *waiting time* ( $Z4$ ), we have a conflicting relationship in the subset  $R1XX$  and a harmonious one in  $R2XX$ . It is also interesting to see that the instances  $R204$  and  $R208$  do not hold the same pair-wise relationship than the rest in comparisons involving *delay time*.

The average correlation values for the random-cluster Solomon's instances appear in Table III. As the other subsets,  $RCXXX$  possesses two subsets:  $RC1XX$  and  $RC2XX$ . The difference lies in that  $RC1XX$  has narrower time windows than  $RC2XX$ . In the first subset, there seems to be very little interaction between the *number of vehicles* ( $Z1$ ) and the other objectives. However, unlike the subsets  $C1XX$  and  $R1XX$ , in this subset the *number of vehicles* ( $Z1$ ) presents a conflict relationship with the *delay time* ( $Z5$ ). We find a similar situation as in  $R2XX$  with comparing *makespan* ( $Z3$ ) versus *waiting time* ( $Z4$ ). The subset  $RC1XX$  shows a conflict relationship, while for the  $RC2XX$  is harmonious.

### C. Proposed MOVRPTW Dataset

Table IV presents the average correlation values for our dataset with 150 customers. This table has three main blocks according to  $\delta$  denoted by  $d$ , where  $d$  takes values  $d0 = 60$ ,  $d1 = 20$  and  $d2 = 5$  (Sect. III). The time windows profiles denoted by  $tw$ , where  $tw = 0$  contains only one time windows  $[0, 480]$ ;  $tw = 1$ ,  $tw = 2$  and  $tw = 3$  contain three time windows of length 160, 130 and 100 minutes respectively. Finally,  $tw = 4$  contains 10 time windows as a combination of all the above (Sect. III). In general, the correlation values across all instances are close to  $1$  or  $-1$ . This indicates strong dependency relationships. At first sight, it can be seen some null values in all instance whose time windows profile is  $0$  ( $d0.tw0$ ,  $d1.tw0$  and  $d2.tw0$ ). This is because this profile considers wide open time windows, thus the *waiting time* ( $Z4$ ) is always  $0$ . The most important conflicting relationships in these instances are *number of vehicles* ( $Z1$ ) versus *delay time* ( $Z5$ ) and *makespan* ( $Z3$ ) versus *waiting time* ( $Z4$ ).

Summarising, the proposed MOVRPTW dataset, which is designed based on real-world data and with a multi-objective mindset, presents better dependency relationships using pair-wise comparisons of the objectives under consideration. We have seen that the subset of Solomon's instances containing narrow time windows ( $C1XX$ ,  $R1XX$  and  $RC1XX$ ) are not suitable to be addressed as real multi-objective instances. Moreover, Solomon's instances with wide time windows and our dataset follow the same pattern with regards to dependency

TABLE I  
AVERAGE CORRELATION VALUE FOR ALL INSTANCES IN SUBSET  $CXXX$ . [SOLOMON'S DATASET] (10 RUNS).

<i>Instance</i>	Z1-Z2	Z1-Z3	Z1-Z4	Z1-Z5	Z2-Z3	Z2-Z4	Z2-Z5	Z3-Z4	Z3-Z5	Z4-Z5
C101	0.22	-0.13	0.29	-0.16	-0.60	0.31	-0.30	-0.77	-0.21	0.27
C102	NA	NA	NA	NA	-0.68	0.29	-0.32	-0.51	0.18	0.31
C103	NA	NA	NA	NA	-0.61	0.12	-0.17	-0.43	-0.06	0.48
C104	NA	NA	NA	NA	-0.48	-0.27	0.29	-0.22	-0.04	0.06
C105	0.16	-0.06	0.19	-0.20	-0.59	0.37	-0.33	-0.68	-0.14	0.11
C106	NA	NA	NA	NA	-0.41	-0.14	-0.42	-0.63	0.08	0.11
C107	0.09	-0.04	0.07	-0.09	-0.69	0.52	-0.25	-0.65	-0.15	0.00
C108	NA	NA	NA	NA	-0.74	0.45	-0.09	-0.73	-0.32	0.56
C109	NA	NA	NA	NA	-0.79	0.27	-0.20	-0.59	-0.19	0.68
C201	0.70	-0.82	0.99	-0.87	-0.81	0.69	-0.80	-0.83	0.96	-0.87
C202	0.82	-0.85	0.99	-0.88	-0.86	0.81	-0.88	-0.86	0.95	-0.88
C203	0.60	-0.80	0.98	-0.88	-0.70	0.57	-0.75	-0.82	0.86	-0.86
C204	0.60	-0.80	0.97	-0.85	-0.60	0.49	-0.63	-0.80	0.76	-0.76
C205	0.75	-0.84	0.99	-0.89	-0.85	0.73	-0.85	-0.84	0.96	-0.88
C206	0.76	-0.84	0.99	-0.90	-0.83	0.74	-0.84	-0.85	0.96	-0.90
C207	0.63	-0.82	0.99	-0.90	-0.79	0.61	-0.76	-0.83	0.95	-0.89
C208	0.74	-0.85	0.99	-0.91	-0.84	0.71	-0.83	-0.85	0.95	-0.90

TABLE II  
AVERAGE CORRELATION VALUE FOR ALL INSTANCES IN SUBSET  $RXXX$ . [SOLOMON'S DATASET] (10 RUNS).

<i>Instance</i>	Z1-Z2	Z1-Z3	Z1-Z4	Z1-Z5	Z2-Z3	Z2-Z4	Z2-Z5	Z3-Z4	Z3-Z5	Z4-Z5
R101	0.10	-0.06	0.03	-0.05	-0.34	-0.40	-0.50	-0.39	0.09	0.35
R102	NA	NA	NA	NA	-0.25	-0.57	-0.15	-0.33	-0.06	0.15
R103	NA	NA	NA	NA	-0.18	-0.70	0.03	-0.22	0.06	-0.12
R104	NA	NA	NA	NA	-0.67	-0.52	-0.09	0.21	-0.04	0.25
R105	0.17	0.02	0.15	-0.16	0.33	-0.50	0.17	-0.58	0.44	-0.22
R106	NA	NA	NA	NA	-0.36	-0.48	-0.19	-0.37	-0.02	0.12
R107	NA	NA	NA	NA	-0.16	-0.64	-0.07	-0.22	-0.07	-0.01
R108	NA	NA	NA	NA	-0.35	-0.38	0.18	-0.33	-0.30	0.25
R109	NA	NA	NA	NA	-0.54	-0.16	0.05	-0.36	-0.16	0.36
R110	NA	NA	NA	NA	-0.31	-0.70	-0.23	-0.20	0.34	0.08
R111	0.06	-0.09	0.19	-0.02	-0.39	-0.35	0.31	-0.47	-0.39	0.20
R112	NA	NA	NA	NA	-0.15	-0.44	0.03	-0.53	-0.29	0.21
R201	0.61	-0.66	0.79	-0.67	-0.66	0.33	-0.70	-0.73	0.84	-0.67
R202	0.53	-0.63	0.65	-0.52	-0.62	0.16	-0.70	-0.74	0.71	-0.48
R203	0.70	-0.60	0.64	-0.24	-0.42	0.12	-0.32	-0.75	0.37	-0.33
R204	0.87	-0.78	0.90	0.43	-0.71	0.63	0.20	-0.81	-0.27	0.42
R205	0.59	-0.60	0.61	-0.63	-0.64	0.14	-0.69	-0.69	0.78	-0.57
R206	0.83	-0.58	0.54	-0.40	-0.58	0.21	-0.57	-0.70	0.61	-0.32
R207	0.89	-0.61	0.62	-0.30	-0.65	0.33	-0.46	-0.59	0.37	-0.22
R208	0.88	-0.62	0.90	0.43	-0.62	0.67	0.24	-0.65	-0.49	0.52
R209	0.84	-0.62	0.75	-0.32	-0.74	0.53	-0.52	-0.74	0.69	-0.45
R210	0.62	-0.67	0.68	-0.63	-0.62	0.19	-0.60	-0.73	0.76	-0.62
R211	0.94	-0.60	0.89	-0.12	-0.65	0.79	-0.23	-0.61	0.39	-0.09

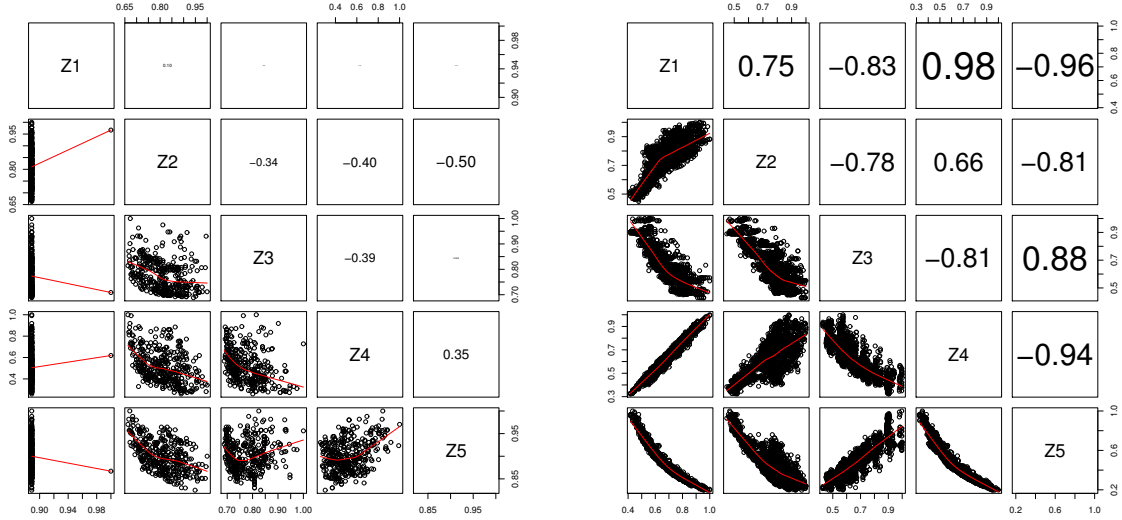


Fig. 2. Scatterplot matrix for Solomon’s  $R101$  (on the left) and our  $d2.tw2$  for 150 customers (on the right). In the main diagonal the objectives  $Z1 \sim 5$  are shown (where  $Z1$  is the number of vehicles,  $Z2$  is the travel distance,  $Z3$  is the makespan,  $Z4$  is the waiting time and  $Z5$  is the delay time).

TABLE III  
AVERAGE CORRELATION VALUE FOR ALL INSTANCES IN SUBSET  $RCXXX$ . [SOLOMON’S DATASET] (10 RUNS).

Instance	Z1-Z2	Z1-Z3	Z1-Z4	Z1-Z5	Z2-Z3	Z2-Z4	Z2-Z5	Z3-Z4	Z3-Z5	Z4-Z5
$RC101$	0.11	0.03	0.17	-0.64	-0.52	-0.33	-0.54	-0.25	0.19	0.13
$RC102$	0.11	-0.03	0.12	-0.53	-0.17	-0.48	-0.16	-0.33	-0.29	0.08
$RC103$	-0.06	-0.14	0.17	-0.41	-0.02	-0.74	0.48	-0.29	0.04	-0.38
$RC104$	0.20	0.16	0.12	-0.50	-0.42	-0.36	-0.15	-0.12	-0.27	-0.18
$RC105$	0.13	-0.22	0.23	-0.66	-0.47	-0.46	-0.36	-0.27	0.23	0.08
$RC106$	0.00	-0.08	0.14	-0.60	-0.55	-0.37	-0.30	-0.20	0.05	0.26
$RC107$	-0.04	-0.35	0.23	-0.67	-0.02	-0.42	0.31	-0.59	0.17	-0.26
$RC108$	0.11	-0.19	0.17	-0.66	-0.19	-0.57	0.08	-0.20	0.23	-0.28
$RC201$	0.69	-0.77	0.82	-0.82	-0.79	0.42	-0.83	-0.76	0.89	-0.74
$RC202$	0.69	-0.77	0.84	-0.76	-0.71	0.36	-0.80	-0.76	0.79	-0.61
$RC203$	0.66	-0.70	0.81	-0.55	-0.57	0.27	-0.63	-0.72	0.60	-0.43
$RC204$	0.85	-0.76	0.68	0.21	-0.73	0.30	-0.04	-0.62	-0.09	0.36
$RC205$	0.69	-0.73	0.83	-0.78	-0.73	0.39	-0.84	-0.73	0.85	-0.67
$RC206$	0.78	-0.69	0.79	-0.73	-0.76	0.43	-0.85	-0.69	0.83	-0.59
$RC207$	0.69	-0.54	0.56	-0.56	-0.67	0.09	-0.76	-0.55	0.77	-0.35
$RC208$	0.86	-0.49	0.46	-0.49	-0.68	0.24	-0.70	-0.52	0.74	-0.33

relationships. Table V shows the conflicting and harmonious relationships for each pair-wise comparison.

## VI. CONCLUSIONS

In this work, we present a study on the suitability of Solomon’s dataset for investigating the multi-objective VRPTW. Our experiments are based on the platform proposed by Purshouse and Fleming [10]. Therefore, we study the dependence (conflict and harmony) and independence of five objectives commonly used in the literature. Using a well-known high performing EMOA (NSGA-II), we conducted a

series of experiments to make pair-wise comparisons among all five objectives using correlation. Results indicate that Solomon’s instances with narrow time windows are not suitable to test multi-objective algorithms. Moreover, this study also revealed that those instances in the Solomon’s dataset that have wider time windows present a sound but still not ideal benchmark scenario for multi-objective VRPTW. This study led us to design a new benchmark dataset for the Multi-objective Vehicle Routing Problem with Time Windows (MOVRPTW). This dataset consists of 45 instances with three different number of customers  $\{50, 150, 250\}$ . These instances

TABLE IV  
AVERAGE CORRELATION VALUE FOR ALL INSTANCES IN SUBSET 150 CUSTOMERS. (10 RUNS).

Instance	Z1-Z2	Z1-Z3	Z1-Z4	Z1-Z5	Z2-Z3	Z2-Z4	Z2-Z5	Z3-Z4	Z3-Z5	Z4-Z5
d0.tw0	0.95	-0.58	NA	-0.64	-0.62	NA	-0.62	NA	0.71	NA
d0.tw1	0.80	-0.86	0.98	-0.92	-0.82	0.72	-0.85	-0.84	0.95	-0.90
d0.tw2	0.81	-0.87	0.99	-0.94	-0.84	0.73	-0.85	-0.85	0.93	-0.93
d0.tw3	0.71	-0.82	0.97	-0.90	-0.74	0.67	-0.76	-0.85	0.90	-0.95
d0.tw4	0.71	-0.71	0.97	-0.90	-0.63	0.64	-0.71	-0.75	0.88	-0.93
d1.tw0	0.97	-0.50	NA	-0.60	-0.38	NA	-0.47	NA	0.77	NA
d1.tw1	0.80	-0.86	0.98	-0.92	-0.82	0.72	-0.85	-0.84	0.95	-0.90
d1.tw2	0.81	-0.87	0.99	-0.94	-0.84	0.73	-0.85	-0.85	0.93	-0.93
d1.tw3	0.71	-0.82	0.97	-0.90	-0.74	0.67	-0.76	-0.85	0.90	-0.95
d1.tw4	0.59	-0.74	0.97	-0.90	-0.60	0.53	-0.61	-0.76	0.88	-0.93
d2.tw0	NA	NA	NA	NA	-0.87	NA	NA	NA	NA	NA
d2.tw1	0.57	-0.81	0.98	-0.93	-0.52	0.47	-0.60	-0.83	0.89	-0.94
d2.tw2	0.75	-0.83	0.98	-0.96	-0.78	0.66	-0.81	-0.81	0.88	-0.94
d2.tw3	0.57	-0.70	0.97	-0.91	-0.57	0.49	-0.62	-0.74	0.83	-0.95
d2.tw4	0.57	-0.73	0.97	-0.93	-0.44	0.49	-0.54	-0.76	0.84	-0.96

TABLE V  
GENERAL DEPENDENCY RELATIONSHIPS ACROSS THE SOLOMON'S INSTANCES SUBSETS: C2XX, R2XX, RC2XX AND MOVRPTW DATASET. CONFLICT RELATIONSHIP IS DENOTED WITH  $\ominus$ , WHILE HARMONY USES  $\oplus$ .

	Z1-Z2	Z1-Z3	Z1-Z4	Z1-Z5	Z2-Z3	Z2-Z4	Z2-Z5	Z3-Z4	Z3-Z5	Z4-Z5
Relationship	$\oplus$	$\ominus$	$\oplus$	$\ominus$	$\ominus$	$\oplus$	$\ominus$	$\ominus$	$\oplus$	$\ominus$

have different profiles to characterise customer demand and also the width of time windows. Our results show that our test instances represent more realistic and challenging MOVRPTW cases. This is because in our instances, the average correlation values between objectives are close to  $-1$  or  $1$ , which indicates a more truly multi-objective nature, better for assessing the performance of multi-objective optimisation algorithms.

The main contribution of this paper is a better understanding of the multi-objective nature of the VRPTW. As an additional contribution, an open source generator of MOVRPTW problem instances is made available to the research community. The instances generator includes raw data corresponding to the position of customers (latitude, longitude) and distinct and non-symmetric travel time and travel distance matrices, all this based on data obtained from a real-world distribution company.

As future work, we will extend this study to understand which elements intervene in the creation of conflicting or harmonious relationships, as well as, to determine which components make instances more difficult to solve.

## REFERENCES

- [1] Cahon, S., Melab, N., Talbi, E.: Paradiseo: a framework for the reusable design of parallel and distributed metaheuristics. *Journal of heuristics* **10**, 357–380 (2004)
- [2] Castro-Gutierrez, J., Landa-Silva, D., Moreno Perez, J.A.: Movrptw dataset (2010). URL <https://github.com/psxjpc/>
- [3] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002)
- [4] Dorronsoro, B.: VRP web. URL <http://neo.lcc.uma.es/radi-aeb/WebVRP/>
- [5] Garcia-Najera, A., Bullinaria, J.: An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research* **38**(1), 287 – 300 (2011)
- [6] Jozefowicz, N., Semet, F., Talbi, E.G.: Enhancements of NSGA-II and its application to the vehicle routing problem with route balancing. In: *7th International Conference on Artificial Evolution - EA 2005, Lecture Notes in Computer Science*, vol. 3871, pp. 131–142. Springer (2006)
- [7] Jozefowicz, N., Semet, F., Talbi, E.G.: Multi-objective vehicle routing problems. *EJOR* **189**(2), 293 – 309 (2008)
- [8] LaTorre, A., Clautiaux, F., Talbi, E., Peña, J.: VRP-extended: When confidence and fleet size are also important. In: E.G. Talbi (ed.) *Proceedings of the 2nd International Conference on Metaheuristics and Nature Inspired Computing, META 2008* (2008)
- [9] Murata, T., Itai, R.: Multi-objective vehicle routing problems using two-fold emo algorithms to enhance solution similarity on non-dominated solutions. In: *Evolutionary Multi-Criterion Optimization, LNCS*, vol. 3410, pp. 885–896. Springer Berlin / Heidelberg (2005)
- [10] Purshouse, R., Fleming, P.: Conflict, harmony, and independence: relationships in evolutionary multi-criterion optimisation. In: *Proceedings of the 2nd international conference on evolutionary multi-criterion optimization (EMO 2003)*, LNCS 2632, pp. 16–30. Springer (2003)
- [11] Solomon, M.: Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations research* **35**, 254–265 (1987)
- [12] Tan, K., Chew, Y., Lee, L.: A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Computational Optimization and Applications* **34**, 115–151 (2006)
- [13] Wei, T., Fan, W., Xu, H.: Greedy non-dominated sorting in genetic algorithm-ii for vehicle routing problem in distribution. *Chinese Journal of Mechanical Engineering* **21**(6), 18–24 (2008)