Vessel Route Planning Optimization Combined with Time Windows versus Worker Scheduling for Offshore Windmill Maintenance

E. De Kuyffer, L. Martens, W. Joseph, T. De Pessemier
Department of Information Technology
Ghent University/IMEC
Ghent, Belgium
e-mail: Erik.DeKuyffer@UGent.be, Luc1.Martens@UGent.be, Wout.Joseph@UGent.be, Toon.DePessemier@UGent.be

***Abstract*—The high fuel prices and the important costs associated with windmill downtime during maintenance urge the need to minimize travel time and scheduling of jobs in a short time period. Since landing in windmills at sea is difficult and depends on meteorological parameters, the constraint of maintenance windows is added when searching for the optimal route. To minimize the distance traveled, the Vehicle Routing Problem with Time Windows (VRPTW) is solved, using three different methods. The VRPTW is applied to two separate databases, namely various sets of windmills to be maintained and several numbers of customers to be serviced. Applications with 8 to 175 windmills, divided over 3 farms have shown that the VRPTW solved by using three different methods resulted in a similar relative gain in travel distance, compared to a randomly chosen route. The main difference between the methods studied is the amount of calculation time needed, which varies from 1 second to 6 minutes for the different methods. To demonstrate the general applicability, the same three methods were executed on a set of service tasks performed at 8 to 40 customers of a window decoration company, distributed throughout Belgium, resulting in similar outcomes. In a second part of the paper, the Job Shop Scheduling Problem (JSSP) is solved to minimize the total maintenance span of offshore windmills as an additional objective function. This led to a relative gain of up to 62% in maintenance time, compared to the total maximum maintenance span for an application of 40 windmills. Finally, both objectives, minimal distance and minimal maintenance time span, are combined, resulting in a set of non-dominated maintenance sequences that can be used by the planner.**

Keywords-VRPTW; VRPy; OR Tools; ACO; Job Shop Scheduling; Pareto.

I. INTRODUCTION

Due to high fuel prices and significant labor costs, it is extremely important to limit the distance covered and the time consumed for offshore windmill maintenance. This paper focuses on the reduction of regular maintenance and repair costs. Vessel routing optimization for offshore windmill maintenance thereby is a very complex problem. It has been the topic of recent studies [1]–[4]. To demonstrate the general applicability of the VRPTW to obtain minimal distance routes and to show that all three methods studied - VRPy, Operational Research (OR) Tools and Ant Colony Optimization (ACO) - lead to similar results for other data sets, the procedures are applied to the discrete product installation planning. The interventions of companies that distribute and maintain unique products per customer - the so-called Value Added Resellers or VARs - can be split in the installation of the products and ad hoc maintenance of previously installed products. The planning of the delivery and installation can be considered as proactive planning, allowing optimization of the distance to be traveled, and thus the amount of fuel used. Maintenance interventions are more reactive of nature, making optimal planning more difficult. In a second part of this paper, we focus on reducing downtime by arranging maintenance jobs in such a way that the total service time span is minimized. To obtain this objective, the Job Shop Scheduling Problem (JSSP) is solved, in which the machines are replaced by workers. For each worker, a sequence is calculated so that all maintenance jobs are executed within a limited time frame, reducing the total downtime of all the windmills that need service. Finally, both objectives are applied to the same set of windmills, resulting in a Pareto front of non-dominated solutions offered to the planner to choose from (Section V). It will become clear from the list of these Pareto points that reaching both objectives at the same time is nearly infeasible, and thus the optimal sequence must be chosen out of this list of non-dominated solutions.

The novelties of this paper are: (i) the comparison of three solution methods for the vehicle routing problem with time windows applied to windmill maintenance vessels and to discrete product installation and service, (ii) the combined windmill sequence travel distance and maintenance time span optimization by solving both the VRPTW and the JSSP on the same data set, and (iii) the importance of reaching both objectives in maintenance cost reduction.

The remainder of the paper is organized as follows. In Section II, references are made to related work and Section III describes the problem formulation. The three methods used to solve the Vehicle Routing Problem with Time Windows, as well as the solution method for the JSSP are listed in Section IV. Section V lists the results of all the optimization methods discussed for VRPTW and JSSP and compares both by calculating the corresponding Pareto points. Finally, Sections VI and VII contain an evaluation of the results and provide a conclusion, respectively.

II. RELATED WORK

The Vehicle Routing Problem (VRP) was first investigated more than six decades ago. This routing problem initiated major developments in the fields of exact algorithms and heuristics [5]. The vehicle routing problem comprises the design of least cost delivery routes through a set of geographically dispersed locations, subject to one or more side constraints.

Constraints to vehicle routing problems linked to capacity result in the Capacitated Vehicle Routing Problem (CVRP). If a time window is added to each location, asset, or customer, we talk about VRPTW [6]. In addition to the capacity constraint, a vehicle in the VRPTW has to visit a location, asset, or customer within a certain time frame. The vehicle - car, vessel, or other - is allowed to arrive before the time window opens, but the customer or asset cannot be serviced until the respective time window opens. In addition, it is not allowed to arrive after the time window has closed [7]. The different solution methodologies for the VRP can be divided into three categories: Exact methods, heuristics, and meta-heuristics. The exact methods generate optimal solutions and guarantee their optimality. This method class includes a variety of approaches, mainly branch and X (X being cut, bound, price, etc.), dynamic programming, and column generation methods. The heuristics aim to methodically find an acceptable solution within a limited number of iterations. Metaheuristics can finally be defined as a class of heuristics that search beyond the local optima if they exist [8]. Research papers on VRP, with or without time windows, are quite common, since their application in daily life is widely spread, for example, in the delivery of packages and the route planning of nurses [9]–[12]. The principle of using VRPTW to optimize offshore wind farm maintenance routes for multiple vessels has never been applied.

In addition, extensive research has been done on maintenance and production scheduling according to the flow shop method, as well as the job shop method in different industries [13]–[15]. Al-Shayea et al. designed a model to integrate production scheduling and maintenance planning for flow-shop production systems. This model is based on the optimal job sequence that will be processed on several connected machines in series. The objective of this study is to find the optimal sequence of jobs, while reducing total production and maintenance costs [16]. None of the papers apply, however, to maintenance scheduling in windmill farms, while costs for this is very high, and every hour of downtime (due to maintenance) results in an important loss of revenue.

III. Problem Formulation

A. Experimental Design - VRPTW

The goal of solving the VRPTW will be to find the optimal maintenance or installation sequence to minimize the total distance traveled, with a minimal number of vessels, taking into account that some maintenance tasks can only be serviced for a certain period of time. For both data sets, the load is never an issue, neither for the vessels that only need to transport maintenance people nor for the vans that are big enough to carry all products that need to be installed in one day. Finally, both data sets are directly obtained from the windmill maintenance company and the window decoration value added reseller, and no prefiltering or preprocessing was done, except a random selection of a predefined batch out of the total set, ranging from 8 to 175 windmills and from 8 to 40 VAR customers.

Figure 1 shows an example of a windmill configuration after applying the VRP solution method with time windows.

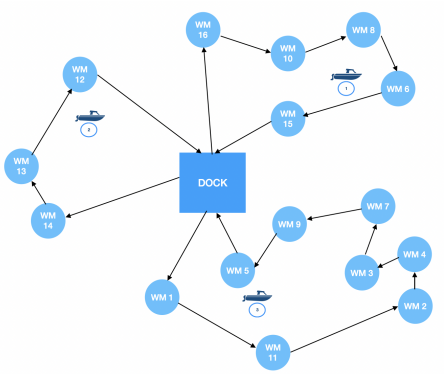

Figure 1. Optimal windmill service routes after solving the VRP.

The configuration used in this example has one dock and 16 windmills spread over three farms to be serviced. The different windmills are indicated as WM_i (i = 1-16), where each windmill must be visited exactly once. Solving the VRP leads to three routes that three different vessels must take, as shown in Figure 1. The fact that more than one vessel is necessary to service all windmills is caused by the time windows in which maintenance of a windmill needs to be carried out. A similar configuration can be set up for the customers of a Value Added Reseller, where the windmills are then replaced by customers to be visited for installation or maintenance of discrete products.

B. Experimental Design - JSSP

In a typical scheduling problem in the job shop, different jobs are scheduled on multiple machines to minimize the total production time [14][15]. However, for this research paper, we replaced the machines on which the work is performed by workers that manually execute the maintenance jobs. For each worker, the sequence of jobs is optimized in such a way that the total work span is minimized. Each job is represented by a different shade of color, and the size of the blocks corresponds to the amount of time it costs to complete the maintenance task. The workflow of a job shop is complex because it is different for every job. In this paper, the job shop scheduling will be discussed as it is a good match with the maintenance jobs scheduling for offshore windmill maintenance, making it the first time, according to the author's knowledge, the JSSP is used for maintenance planning of offshore windmills.

Figure 2 shows an example of a job sequence per worker obtained by solving the JSSP for a group of 16 windmills spread over 3 farms. The number of workers is set to three on the vertical axis, in analogy with the number of machines in the original JSSP used in a production environment. In each windmill, one worker needs to perform a service task and each worker needs to perform several maintenance tasks in separate windmills. Applying the JSSP solver to this configuration leads to an optimal sequence in which each worker needs to perform service on the windmills (s)he is responsible for, with a different service time on each windmill, shown on the horizontal axis. Thus, for each worker, a sequence of jobs is

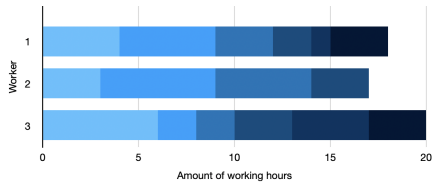

Figure 2. Optimal windmill service job sequence for 3 workers and different colored jobs of various length after solving the JSSP.

shown, each corresponding with a different color and with a size in accordance with the length of the job.

C. Output Parameters

The output parameters for the VRPTW problem are the optimal number of vessels or trucks, the optimal sequence, and the travel time for each vessel or truck in minutes. Furthermore, the relative gain in travel time (ΔG_t) is calculated by dividing a randomly chosen travel time ($TTt|Random$) minus the Total Travel time (TTt) by a randomly chosen travel time (see (1)). The output parameters for the JSSP problem are an optimized sequence of maintenance tasks per worker to minimize Total Downtime (TDt). This is then compared to the total time needed for one worker ($TDt|OneWorker$) and a relative gain is calculated (ΔG_w) by using (2).

$$\Delta G_t = 100 \cdot \frac{TTt|Random - TTt}{TTt|Random}(\%) \tag{1}$$

$$\Delta G_w = 100 \cdot \frac{TDt|OneWorker - TDt}{TDt|OneWorker}(\%) \tag{2}$$

IV. Method

A. Solution method - VRPTW

In this paper, three methods are discussed, namely VRPy, a tool that uses a Column Generation Approach (CGA) [17], the OR Tools solver, developed by Google Operational Research [11], and an ACO algorithm, a metaheuristic solving method [18]. Table I summarizes the three methods and describes their characteristics. While VRPy takes (much longer) to arrive at a result, especially for large datasets, OR Tools generates a result almost instantaneously but of slightly less quality. These conclusions are numerically confirmed in Tables II and III. All calculations are done on a MacBook Pro from 2021 with the new Apple M1 chip and 8Mb RAM.

TABLE I
Comparison of all methods used to solve the VRPTW

Method	Advantage	Disadvantage
VRPy	Easy Interface	Less Powerful
OR Tools	Fast and Accurate	No optimal result
ACO	Optimal results	No Easy Interface

To calculate the distances between the windmills or customers and between the starting point and the windmills or customers, spherical trigonometry formulas are used. In this paper, all vehicles are considered the same: they have the same velocity, the same capacity, and unit freight. Furthermore, the capacity and cargo of the vessel are not considered constraints.

When defining t_i as the time it takes for the vessel to arrive at location i, e as the cost of waiting and f as the cost of arriving too late, the objective of solving the VRPTW for a collection of vehicles A, can be written as:

$$min(\sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{a=1}^{A} x_{ija} * d_{ij} + \sum_{i=1}^{N} max\{e*(m_{bi} - t_i); 0; f*(t_i - m_{ei})\} \tag{3}$$

Where:

$$x_{ija} = \begin{cases} 1 & \text{if the vehicle a travels from i to j,} \\ 0 & \text{in all other cases} \end{cases} \tag{4}$$

$$t_{ija} = \sum x_{ija}(t_i + \frac{d_{ij}}{v} + s_i) \quad (t_0 = 0, s_0 = 0) \tag{5}$$

The constraints are:

$$\sum_{j=1}^{N}\sum_{a=1}^{A} x_{jia} = \sum_{j=1}^{N}\sum_{a=1}^{A} x_{ija} = A \quad (i = 0) \tag{6}$$

$$\sum_{j=0}^{N}\sum_{a=1}^{A} x_{ija} = 1 \quad (i \in N) \tag{7}$$

$$\sum_{i=0}^{N}\sum_{a=1}^{A} x_{ija} = 1 \quad (j \in N) \tag{8}$$

$$\sum_{j=1}^{N} x_{ija} = \sum_{j=1}^{N} x_{jia} = 1 \quad (i = 0 \quad a \in A) \tag{9}$$

In (3), the second part of the equation - sum of maximums - defines the time window constraint. In (5), t_{ija} is the time it takes the vehicle to travel from location i to j, v is the speed and s_i the service at location i. At the depot (node 0 in the equations), both t and s are equal to zero. The constraint in (6) implies that the number of vehicles that start from the loading point and go back there is A. Constraints (7) and (8) mean that each location can be visited only by one vehicle. Finally, constraint (9) represents that all the vehicles that start from the loading point also go back there. VRPy solves the vehicle routing problem with a column generation approach [17]. The term refers to the fact that, continuously, routes are generated with a pricing problem and fed to a master problem. The latter selects the best routes among a pool so that each node (windmill or customer in this case) is serviced exactly once. The pricing problem is actually a shortest elementary path problem. Additional constraints, such as the time windows discussed in this paper, contribute to a shortest-path problem with resource constraints. VRPy does not lead to an optimal solution, even without time limits. Hence, when solving pricing problems does not result in a route with negative marginal cost, the master problem is solved as mixed integer programming. This price-and-branch strategy does not guarantee an ideal solution.

Next, the above solution will be compared with the results found for the same operational VRPTW using the solver developed for Google OR Tools (Table I) [11]. The algorithm based on the Python routing library wrapper results in a new set

of optimal routes, taking into consideration that all windmills or customers need to be serviced in a specific time frame. As for the first method, no other restrictions are taken into account. The algorithm used to solve VRPTW starts with the creation of input data, followed by a callback function. After adding the time constraints, the default search parameters and a heuristic method are set for the first solution. Finally, the same function is used to solve the Traveling Salesman Problem (TSP), resulting in the route for each vehicle, the total travel time of the vehicle route, and the solution windows of each location. The solution window at a location is defined as the time interval during which a vehicle must arrive, so it stays on schedule.

The third method to solve the VRPTW with the same entry data - like position of the windmills or customers, service windows - is based on an ant colony optimization algorithm [4], [18]. Ant Colony Optimization (ACO) is one of the most recent metaheuristic approaches to combinatorial optimization problems. The pseudocode is shown below. All three solution methods, VRPy, OR Tools, and ACO are heuristic methods. When it is impossible or impractical to find an optimal solution, heuristic methods can be used to accelerate the process of discovering a satisfactory solution. These heuristics can be described as strategies derived from previous experiences with similar problems.

```
procedure ACO Meta-Heuristic is
    while not terminated do
        ConstructAntsSolutions()
        UpdatePheromones()
        daemonActions()
    repeat
end procedure
```

ACO is based on the foraging behavior of real ants. They arbitrarily explore the environment, using pheromone deposits to find the shortest routes. Therefore, ACO algorithms are probabilistic techniques suitable for solving optimization problems that aim at minimizing the distance traveled (e.g., TSP and VRP). In the first step - *ConstructAntsSolutions* - of the algorithm, each artificial ant generates a solution: thereby it randomly chooses the next city to visit, based on a heuristic combination of the distance to that city and the amount of virtual pheromone left behind on the arc to that city. The ants explore and dump pheromone on each arc they traverse until they have all completed a tour (see (10)). At this point, the ant that has completed the shortest tour deposits virtual pheromone along its complete route (*UpdatePheromones*). Equation (14) shows that the amount of pheromone deposited is inversely proportional to the length of the tour. Thus, the shorter the route, the more pheromone the ant deposits on the arcs of the corresponding tour. The *deamonActions* procedure is used to carry out centralized actions that cannot be carried out by individual ants, as they do not possess global knowledge. A typical example of these *deamonActions* is the collection of global information that can be used to decide whether it might be useful to deposit additional pheromone to bias the search process from a nonlocal perspective. As long as the termination condition is not met, these three steps are repeated [19]. The pheromone τ_{ij}, associated with the edge joining locations i and j, is updated as follows:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \tag{10}$$

where:

$$\rho = \text{evaporation rate,} \tag{11}$$

$$m = \text{number of ants,} \tag{12}$$

$$\Delta\tau_{ij}^{k} = \text{the quantity of pheromone laid on edge (i, j) by ant k,} \tag{13}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & \text{if ant k used edge (i,j) in its tour,} \\ 0 & \text{in all other cases} \end{cases} \tag{14}$$

$$Q = \text{a constant,} \tag{15}$$

$$L_k = \text{is the length of the tour built by ant k,} \tag{16}$$

To determine and confirm that the solutions obtained by the three methods to solve VRPTW are applicable in other areas, the same solution procedures were applied to two data sets. In addition to windmill farms, a data set is used consisting of the customer coordinates at which discrete Taylormade products (curtains) are to be installed and maintained. Although this data set differs quite extensively from that of the windmills, the optimization goals are the same, namely travel time, and thus fuel consumption reduction. Transport for Taylormade products goes over land and cannot follow a straight line, the distances between customers are smaller than for the windmill farms (typically a few tens of kilometers versus a few hundreds for the windmills) and not clustered around different farms, making the data sets for windmills and customers quite different.

B. Solution Method - Job shop

To apply the solver to maintenance planning, we have made the following assumptions: the machines in the JSSP are replaced by the workers performing maintenance jobs (the job is a sequence of windmills to be serviced), and the tasks are linked to the windmills. The processing time is chosen randomly, as are the workers for each maintenance job. The final result of the algorithm created to solve the JSSP will be a schedule optimized for each worker to minimize the total maintenance span. Figure 3 shows the building blocks of the algorithm used to solve JSSP with the OR solver.

Each of the steps in the flow chart are further defined as:

- Data Creation: For each maintenance job, several tasks are defined, that is, the windmills or customers to be serviced. For every windmill or customer, the worker that needs to perform the task and the service time needed are given.
- Declaration of the model, a Constraint Programming (CP) model that includes variables and constraints that will be solved via the CP solver.

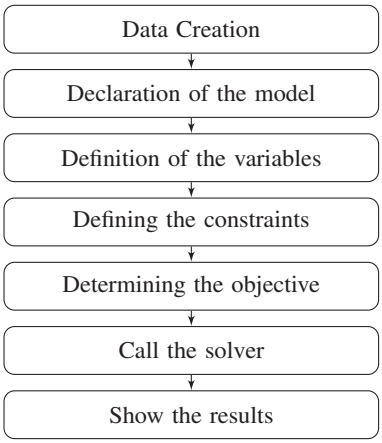

Figure 3. Flow Diagram of the JSSP solution method.

- Definition of the variables, which are the start and end time, the duration (end minus start) and the interval of the task.
- Defining the constraints: A worker cannot work at two windmills at the same time, plus the condition that, for any two consecutive tasks in the same maintenance job, the first must be completed before the second can be started.
- Determining the objective as the minimization of the make span.
- Call the solver and show the results.

The objective function (17) and the constraints of the job shop scheduling problem are written as:

$$min\ C_S \tag{17}$$

where:

$$C_{1j} - w_{1j} = r_j + p_{1j} \qquad \forall j \tag{18}$$

$$C_{i-1j} - C_{ij} + w_{ij} = -p_{ij} \qquad i = 2, ..., m_j, \quad \forall j \tag{19}$$

$$C_{ik} - C_{ij} \geq p_{ik} \quad \text{or} \quad C_{ij} - C_{ik} \geq p_{ij} \quad \forall i, \quad \forall j, k \in J_i \tag{20}$$

$$C_{ij}, w_{ij} \geq 0 \quad i = 1, ..., m_j \quad \forall j \tag{21}$$

Constraint (18) implies that a maintenance job can only start after its respective ready time. Constraint (19) specifies that a job j follows its processing sequence. The machine capacity constraint (20) finally ensures that a worker can process only one operation at a time, and an operation will be finished once it starts.

C. Pareto front

Solving the VRPTW and JSSP leads to sequences in which windmills or customers need to be visited, each with a different objective function. Hence, for the VRPTW, the objective is to minimize the total distance traveled, while the JSSP attempts to reduce the total maintenance time. Since both objectives are possibly contradictory, the solution methods are compared by calculating Pareto points and a corresponding Pareto front. Therefore, the maintenance sequences resulting from the VRPTW, with a minimal route distance, are offered to the second objective function to calculate the corresponding total maintenance time. Additionally, maintenance sequences with minimal maintenance are calculated by solving the JSSP and the corresponding total route distance is determined. Both lead to a set of two-dimensional coordinates of which the Pareto points are calculated. Pareto optimal points are non-dominated, meaning that there does not exist another solution that rigorously dominates the Pareto optimal solution in terms of any objective. The Pareto front is the multi-objective and multi-dimensional alternative for the individual optimal solution resulting from single objective optimisation problems (VRP and JSSP).

V. Results

A. Sequence comparison VRPy - OR Tools - ACO for windmill maintenance

Table II lists the best results obtained by applying all three solution methods, and this for different configurations, ranging from 8 to 175 windmills. The relative gain shows how much better the optimized solution is than the randomly chosen one. The optimal number of vessels proposed by the VRPTW solvers is shown in the Vessel column, and finally the Runtime column lists the time needed to solve the VRPTW problem. For VRPy and OR Tools, the same sequence was obtained when running 10 tests for each. With ACO, the best result represents the shortest routes obtained after 20 tests, with 50 ants and 1000 iterations. The randomly selected route is considered identical and is expressed in minutes of travel time for the maintenance vessels. The values correspond to the total travel time of all vessels used in the maintenance schedule. Applying VRPy on a selection of 16 windmills to be maintained, to solve the Vehicle Routing Problem with time constraints, leads to a relative gain compared to the randomly chosen route of a little more than 44%. To obtain this minimal total travel time, three vessels need to be deployed simultaneously, each following a separate route.

Table II shows that the three solution methods, VRPy, OR Tools, and ACO, lead to an almost equal relative gain compared to a random route time of all vessels involved. This accounts for all configurations, varying from 8 to 40 windmills, and increases gradually as the number of windmills to be maintained grows. Except for the configuration of 8 Windmills, the number of vessels proposed by each method are the same, making comparison easier. The only significant difference between the VRPTW solvers is the calculation time required to obtain an optimized solution. Although the average calculation time for the smallest configuration is almost zero and comparable for all options, it rises very fast - almost exponentially - for the VRPy solution, up to more than 350 seconds for the 40 windmills. The calculation time of the ACO algorithm also increases, but is linear and thus not as distinct as for the VRPy solution method. OR Tools finally results in a set of optimized routes instantaneously, even for the set of 40 windmills.

Table II further contains the results obtained using the three VRPTW solution procedures for a large set of 175 windmills. For this sample, there is a (very) high relative gain for all three solvers, but also a significant difference between the yields obtained by VRPy and OR Tools and that by ACO. Although

TABLE II
OPTIMIZATION RESULTS FOR ALL METHODS FOR DIFFERENT WM CONFIGURATIONS

Method	Rel gain (%)	Vessels	Run-time (sec)
8 Windmills			
VRPy	11.8%	3	0.33
OR Tools	16.6%	2	0.03
ACO	16.6%	2	1.23
16 Windmills			
VRPy	44.1%	3	1.59
OR Tools	44.1%	3	0.04
ACO	44.0%	3	2.87
24 Windmills			
VRPy	68.3%	3	10.02
OR Tools	68.4%	3	0.05
ACO	68.3%	3	4.52
32 Windmills			
VRPy	70.2%	3	70.61
OR Tools	70.3%	3	0.12
ACO	70.2%	3	7.67
40 Windmills			
VRPy	77.3%	3	351.39
OR Tools	77.3%	3	0.09
ACO	76.3%	3	19.71
175 Windmills			
VRPy	91.4%	3	>24h
OR Tools	**91.7%**	**3**	**3.51**
ACO	81.2%	9	507.50

not negligible, the calculation time for OR Tools is only 3.5 seconds, while ACO now requires more than 8 minutes to obtain a much worse result for a larger number of vessels. VRPy takes an extremely long time to get to a set of optimized routes.

TABLE III
OPTIMIZATION RESULTS FOR ALL METHODS FOR DIFFERENT CUSTOMER CONFIGURATIONS

Method	Rel gain (%)	Vessels	Runtime (sec)
8 Customers			
VRPy	29.1%	3	0.45
OR Tools	28.7%	3	0.03
ACO	28.8%	3	2.04
16 Customers			
VRPy	52.5%	4	1.70
OR Tools	52.9%	3	0.03
ACO	52.1%	4	2.98
24 Customers			
VRPy	65.0%	3	11.53
OR Tools	59.9%	3	0.04
ACO	63.4%	4	6.64
32 Customers			
VRPy	64.8%	3	47.93
OR Tools	64.8%	3	0.09
ACO	62.8%	4	7.89
40 Customers			
VRPy	70.5%	4	70.09
OR Tools	**70.3%**	**4**	**0.12**
ACO	64.1%	4	11.08

B. Sequence comparison VRPy - OR Tools - ACO for customer interventions

The same solution methods were applied to another data set. This set contains the coordinates of customers of a company that performs interventions on site. These clients are distributed throughout Belgium and are chosen at random from the company's database. The main difference with the windmill configuration is the way the locations are spread: while the windmills are grouped in three so-called parks, the customers are scattered throughout the Belgian territory.

Table III shows that the relative gain obtained by the OR solver, VRPy and ACO is again increasing as the number of customers to be served grows. Also, the conclusions about the calculation times are similar to those made for the windmill case: very limited for OR Tools, being almost instantaneously; slightly increasing for the ACO algorithm, ranging from 2 seconds for 8 customers up to 11 seconds for 40 customers and evolving in a more or less linear way; and finally more largely increasing for VRPy, from less than 1 second for 8 to over 70 seconds for 40 customers, following a more exponential curve. However, there are some important differences. First, there are slightly larger gaps between the relative gain obtained for every solution method, while for the windmill case, the results are nearly equal. This is probably due to the fact that there is a clustering around the different farms, making it easier for each method to get stuck in local minima much faster in the previously discussed windmill case. In the taylormade data set, there is no clustering and thus this phenomenon does not arise. Second, the optimal number of vehicles is not always equal for each solution, making the comparison more difficult.

C. Job Shop - Workers and windmill maintenance combined

Table IV shows the results of the tests with a different number of windmills, divided over 3 separate farms, ranging from 8 to 40 assets. If all maintenance jobs would be carried out consecutively by one worker without waiting time - being the worst-case scenario for the total maintenance time span - the total time span for all jobs would be 18h for 8 windmills and 120h for 40 windmills. However, if we optimize the schedule for more workers, the total time span would be much lower, being 11 hours for 2 workers in the 8 windmills configuration and 45h for 6 workers in the 40 windmills configuration. This corresponds to a relative gain in maintenance time of respectively around 39% and 62% in the total maintenance time span with respect to the single worker case. By employing more workers simultaneously, the total maintenance time lost is (more than) halved, and therefore downtime is reduced by (more than) 50%. Although the total number of working hours is higher when using three workers instead of one, the amount of money gained by halving the downtime is significantly higher, hence the huge advantage of the JSSP solver. According to the average price per kWh in December 2022, the loss per windmill for 1h downtime is at least 203€ per hour if we presume that a windmill operates 24h per day, 365 days per year. A reduction of the downtime by 67h (with 3 workers) thus leads to a cost reduction of more than 13.6K Euro per

windmill. If we further estimate the average labor cost per worker at 60 euros per hour and compare the total amount of hours worked by three workers (154 hours) with the 120 hours needed for one worker, then the extra costs would be 34 times 60 euros, or 2K euros. The net gain would then be 13.6K minus 2K, thus 11.6K.

TABLE IV
JSSP OPTIMIZATION BY USING OR TOOLS

Use Case	Relative gain
08 Windmills - 2 workers	38,9%
16 Windmills - 3 workers	48,7%
24 Windmills - 4 workers	53,8%
32 Windmills - 5 workers	59,6%
40 Windmills - 6 workers	62,5%

Table V shows that in the 40 windmill configuration, the largest downtime gain is obtained when switching from one to two workers (44%) and a much lower but significant gain when switching to three workers. From 4 workers onwards, the total maintenance time span does not lower very much when adding extra workers. The trade-off can thus be put at 4 workers or, when labor is expensive, at 3 workers. Remark that when using as many workers as there are tasks to perform, the relative gain is obtained by dividing the longest task by the total time for all tasks, and thus results in a very high optimization (95% in our case).

TABLE V
JSSP OPTIMIZATION IN FUNCTION OF THE NUMBER OF WORKERS FOR 40 WM AND 80 WM

Number of workers	Rel gain 40WM	Rel gain 80WM
2	44.2%	45.7%
3	55.8%	56.0%
4	60.0%	59.0%
5	60.8%	61.2%
6	62.5%	62.5%

Table V also shows similar results for a configuration of 80 windmills. A large reduction in total maintenance time when a second worker is added, with a relative trade-off at 4 workers. The same results can be extrapolated to the use case of Taylormade products, since results are based on randomly chosen maintenance times, and the location of windmills and customers does not influence the final results of the JSSP.

D. Combined results and Comparison

In order to determine the link between the optimal route resulting from solving VRPTW and the routes determined by solving the JSSP to minimize the time span of all maintenance jobs, a Pareto front is calculated. To compute this Pareto front with non-dominated solutions, tests were run on the two separate problems, and each result was then offered to the other problem. To clarify this, the following example is described: the ACO algorithm, VRPy and OR Tools solution methods ran to solve the VRPTW resulted in maintenance sequences with minimal total traveling distance. For this windmill sequence, the corresponding total time span for all maintenance tasks are calculated by adding the maintenance time for all jobs in this sequence. On the other hand, the total distances are computed for the sequences resulted from solving the JSSP (with minimal time span). This is done for sequences of 40 windmills and 3 vessels. Figure 4 shows the Pareto front.

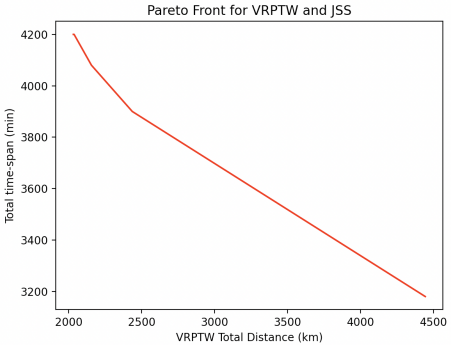

Figure 4. Pareto front for VRPTW-JSS comparison.

Our research has led to a group of Pareto optimal maintenance sequences as a result of the multi-objective optimization model (see coordinates in Table VI). However, it has proven to be very difficult to find a maintenance sequence that is optimal for both objectives. For example, the first and second jobs to be carried out initially according to minimize the total time span can be far away from each other, resulting in a total distance higher than the one obtained by solving the VRPTW. The tests resulted in maintenance paths that either have a low total time span and a high distance, or have a low distance but a high maintenance time span. In determining the optimal sequence, the planner has to decide which parameter is most important when making the choice. From all studied maintenance sequences, a list of 4 non-dominated solutions is obtained. Three of the solutions offer a path with a lower distance and a higher maintenance time span, one is showing a large distance and a lower time span. None of the tests resulted in a path with low values for both objectives.

TABLE VI
PARETO POINTS

Coordinates	Distance VRPTW (min)	Distance JSSP (min)
1	2156	4080
2	2436	3900
3	2037	4200
4	4444	3180

To compare the financial gain obtained by applying VRPTW and JSSP, we consider the case of 40 windmills and 3 workers.

- When reducing the distance from 8380km to 2064km with VRPTW, the financial gain is around 6180€. Fuel consumption is calculated as the distance traveled, multiplied by the weight of the vessel (30 tons), divided by 1000. The speed of the vessel is set at 5.5 km per hour, the fuel price is 2€ per liter, and no wind or current is taken into account.

- When setting the cost per hour downtime at 203€ and, from Table V, the reduction of the downtime at 67h, the total financial gain is 13600€.

The JSSP method thus leads to a much greater benefit than the distance reduction of the VRPTW solution.

VI. Discussion | Evaluation

Of all methods tested to solve VRPTW, the OR Tools solver offers the quickest solutions, while VRPy and ACO generate similar results but much slower. Both use cases - windmill maintenance and product installation - show similar results with respect to the outcome of the solution method used and the calculation time needed. Also, for the JSSP, the OR Tools solver has proven to be fast and accurate. Comparing the solutions for both objective functions, being distance minimization and maintenance time span optimization, led to sequences that are only optimal for one of the two objectives. Therefore, Pareto points are calculated to obtain solutions that are as optimal as possible for both objectives. The planner can then use these resulting sequences to schedule maintenance tasks for a windmill park to minimize the distance traveled, downtime, or both. In all cases, this leads to a significant reduction in maintenance costs by reducing the fuel used or the loss of energy production. However, several constraints were not taken into account when solving VRPTW, such as sea currents, wind, and the capacity of the vessel. These can be integrated in future work to determine the impact they could have on the final results.

VII. Conclusion and Future Work

When adding constraints to the VRP in the form of time windows for every windmill in which maintenance was needed, the relative gain obtained is 77% for a set of 40 windmills and 17% for a group of 8 windmills, spread over 3 farms, and this for all VRPTW solution methods used. Similar results were found and the same conclusions can be drawn for the second use case, the installation and maintenance of discrete products, showing the general applicability of all methods used to solve the VRPTW. A relative gain of the total maintenance span of almost 62.5% compared to the situation where all maintenance was done by one worker for a configuration with 40 windmills and 39% for 8 windmills was obtained when solving the JSSP. The total time needed for every added worker resulted in a higher total number of working hours to be paid. However, the total maintenance time span was more than halved, resulting in a significant gain in up-time.

In future research, other methods for solving VRPTW and JSSP can be studied and benchmarked. Possible other modi operandi to solve the VRPTW are (nonexhaustive): Harmony Search Algorithms (HAS), Memetic algorithms (MA), Genetic Algorithms (GA), the Hexaly solver, etc. For calculating the JSSP, heuristics or metaheuristics - such as Simulated Annealing, Tabu Search, ACO and Genetic Algorithms - can be compared. In addition, extended and different data sets can be investigated to further determine the applicability of the methods discussed.

Acknowledgement

This work was executed within the project OPAL, Offshore Predictable Accessibility by Learning from data and experts with reference HBC.2020.3227, and funded by the Flemish government.

References

[1] D. Fan et al., “A hybrid heuristic optimization of maintenance routing and scheduling for offshore wind farms”, *Journal of Loss Prevention in the Process Industries*, vol. 62, 2019.

[2] R. Dawid, D. McMillan, and M. Revie, “Decision Support Tool for Offshore Wind Farm Vessel Routing under Uncertainty”, *Energies*, vol. 11, 2018.

[3] D. Juliandri, H. Mawengkang, and F. Bu’ulolo, “Discrete Optimization Model for Vehicle Routing Problem with Scheduling Side Constraints”, *IOP Conference Series: Materials Science and Engineering*, vol. 300, 2018.

[4] Z. Y. Zhang, “Multi-ACO Application in Routing and Scheduling Optimization of Maintenance Fleet (RSOMF) Based on Conditions for Offshore Wind Farms”, *Journal of Power and Energy Engineering*, vol. 6, pp. 20–40, 2018.

[5] G. Laporte, “Fifty years of vehicle routing”, *Transportation Science*, vol. 43, pp. 408–416, 2009.

[6] P. Toth and D. Vigo, “The vehicle routing problem”, *Society for Industrial and Applied Mathematics*, vol. 107, pp. 32–42, 2002.

[7] N. A. El-Sherbeny, “Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods”, *Journal of King Saud University - Science*, vol. 22, pp. 123–131, 2010.

[8] R. Goel and R. Maini, “Vehicle routing problem and its solution methodologies: A survey”, *Int. J. Logistics Systems and Management*, vol. 28, pp. 419–435, 2017.

[9] K. Braekers, K. Ramaekers, and I. Van Nieuwenhuyse, “The Vehicle Routing Problem: State of the Art Classification and Review”, *Computers & Industrial Engineering*, vol. 99, 2015.

[10] D. G. N. D. Jayarathna, G. H. J. Lanel, and Z. A. M. S. Juman, “Industrial vehicle routing problem: a case study”, *Journal of Shipping and Trade*, vol. 7, 2022.

[11] Google OR-Tools, “Vehicle Routing Problem with Time Windows”, *https://developers.google.com/optimization/routing/vrptw*, 2022.

[12] F. Arnold, M. Gendreau, and K. Sörensen, “Efficiently solving very large-scale routing problems”, *Computers and Operations Research*, vol. 107, pp. 32–42, 2019.

[13] J. Teunissen, “Improving production planning by flow shop scheduling algorithms : A case study at forfarmers”, M.S. thesis, University of Twente, 2018.

[14] K. Bülbül and P. Kaminsky, “A Linear Programming-Based Method for Job Shop Scheduling”, *Journal of Scheduling*, vol. 16, pp. 161–183, 2013.

[15] Y. Yu, “A Research Review on Job Shop Scheduling Problem”, *E3S Web of Conferences*, vol. 253, 2021.

[16] A. Al-Shayea, E. Fararah, E. A. Nasr, and H. A. Mahmoud, “Model for Integrating Production Scheduling and Maintenance Planning of Flow Shop Production System”, *Procedia Engineering*, vol. 8, pp. 208 826–208 835, 2020.

[17] R. Montagné and D. Torres Sanchez, “VRPy Documentation”, *https://vrpy.readthedocs.io/en/latest/*, 2020.

[18] B. Catay, “Ant Colony Optimization and Its Application to the Vehicle Routing Problem with Pickups and Deliveries”, *Natural Intelligence for Scheduling, Planning and Packing Problems. Studies in Computational Intelligence*, vol. 250, 2009.

[19] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization”, *IEEE Computational Intelligence Magazine*, vol. 1, pp. 28–39, 2006.