This is the peer reviewd version of the followng article:

Constraint Programming models for the parallel drone scheduling vehicle routing problem / Montemanni, R., Dell'Amico, M.. - In: EURO JOURNAL ON COMPUTATIONAL OPTIMIZATION. - ISSN 2192-4406. - 11:(2023), pp. 100078-100078. [10.1016/j.ejco.2023.100078]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

14/11/2026 20:27

(Article begins on next page)

Journal Pre-proof



Constraint Programming models for the parallel drone scheduling vehicle routing problem

Roberto Montemanni, Mauro Dell'Amico

PII: S2192-4406(23)00022-9
DOI: https://doi.org/10.1016/j.ejco.2023.100078
Reference: EJCO 100078

To appear in: *EURO Journal on Computational Optimization*

Please cite this article as: R. Montemanni, M. Dell'Amico, Constraint Programming models for the parallel drone scheduling vehicle routing problem, *EURO J. Comput. Optim.* (2023), 100078, doi: https://doi.org/10.1016/j.ejco.2023.100078.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2023 Published by Elsevier.

Journal Pre-proof

Highlights

- Models for an actual logistic problem about distribution of goods with trucks and drones working together.
- Effective use of constraint programming.
- Improvements of several state-of-the-art upper and lower bounds for optimal solutions.

Journal Pre-proof

Journal Pre-proof

Constraint Programming models for the parallel drone scheduling vehicle routing problem

Roberto Montemanni*, Mauro Dell'Amico

Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy

Abstract

Drones are currently seen as a viable way for improving the distribution of parcels in urban and rural environments, while working in coordination with traditional vehicles like trucks. In this paper we consider the parallel drone scheduling vehicle routing problem, where the service of a set of customers requiring a delivery is split between a fleet of trucks and a fleet of drones. We consider two variations of the problem. In the first one, the problem is more theoretical, and the target is the minimization of the time required to complete the service and have all the vehicles back to the depot. In the second variant, more realistic constraints involving operating costs, capacity limitation and workload balance, are considered, and the target is to minimize the total operational costs. We propose different constraint programming models to deal with the two problems. An experimental champaign on the instances previously adopted in the literature is presented to validate the new solving methods. The results show that, on top of being a viable way to solve problems to optimality, the models can also be used to derive effective heuristic solutions and high-quality lower bounds for the optimal cost, if the execution is interrupted before its natural end.

Keywords: Parallel Drone Scheduling Vehicle Routing Problems, Constraint Programming, Drones, Optimization

1. Introduction

In the last few years, drone technology has seen investments for billion of dollars, due to its potentials. Forbes [4] defined such a phenomenon the "Drone Explosion". Drones can be applied to many sectors, among which logistics, surveillance and disaster relief [13]. The most prominent application is probably logistics related to e-commerce, which has experienced an exponential growing

*Corresponding author

Email addresses: roberto.montemanni@unimore.it (Roberto Montemanni), mauro.dellamico@unimore.it (Mauro Dell'Amico)

Journal Pre-proof

Journal Pre-proof

in the last decades (Statista, [18]). In [19], the authors forecast that autonomous vehicles will deliver about 80% of all parcels in the following ten years. Several advantages associated with the use of aerial drones can be identified: they do not have to stick to the road network but can fly approximately straight line. Moreover, they are not affected by road traffic congestions. The technology might lead to innovative solutions of interest for the companies (operational costs reduction), for the customers (faster deliveries) and for the whole society (sustainability). In this work we will focus on operational delivery strategies with a mixed fleet using both trucks and drones.

The seminal work of Murray and Chu [10] pioneered a new routing problem in which a truck and a drone collaborate to make deliveries. From an operations research perspective, the authors present two new prototypical variants expanding from the traditional Traveling Salesman Problem (TSP) called the Flying Sidekick TSP (FSTSP) and the Parallel Drone Scheduling TSP (PDSTSP). In both cases a truck and drones collaborate to deliver parcels, the difference being however that in the former model drones can be launched and collected from the truck during its tour, while in the latter one drones are operated directly from the central depot, while the truck executes a traditional delivery tour. In the remainder of the paper we will focus on the latter problem, addressing the interested reader, for example, to [2] for full details and some solution strategies for the FSTSP.

More formally, in the PDSTSP there is a truck that can leave the depot, serve a set of customers, and return to the depot, and a set of drones, each of which in the meantime can leave the depot, serve a customer, and return to the depot before serving other customers. Not all the customer can be served by the drones, either due to their location or the characteristic of their parcel. The objective of the problem is to minimize the completion time of the last vehicle returning to the depot (or a cost function related to this), while serving all the customers.

A first Mixed Integer Linear Programming (MILP) model for the PDSTSP and some simple heuristic methods are proposed in [10]. A more refined mixed integer programming model and the first metaheuristic method, based on a two steps strategy, embedding a dynamic programming-based component, are discussed in [6]. A similar two steps approach, but based on matheuristics concepts is presented in [1]. A hybrid ant colony optimization metaheuristic is discussed in [3]. In [8] a constraint programming approach is discussed, which is able to solve to optimality all the benchmark instances previously adopted in the literature for both exact and heuristic methods. An improved variable neighbour search metaheuristic is discussed in [5]. More recently, in [12] another exact approach based on branch-and-cut was proposed, together with some new benchmark problems.

Several PDSTSP variants are also introduced and studied in the literature. We refer the interested reader to [13] and [14] for a complete survey. In the following we review only the extensions of the original problem relevant to the present study, where multiple trucks are employed out of a same depot, and more realistic constraints involving load balancing, capacity and costs are eventually

Journal Pre-proof

considered.

The recent work [7] discusses the *Parallel Drone Scheduling Multiple Traveling Salesman Problem* what we will here refer to as the MT-PDSVRP (*Min-Time Parallel Drone Scheduling Vehicle Routing Problem*), which is a straightforward extension of the PDSTSP where multiple trucks are employed and the target is to minimize the time required to complete the delivery to the last customer serviced and go back to the depot. The authors propose a hybrid metaheuristic algorithm along the line of the method previously introduced in [6], a mixed integer linear model and a branch-and-cut approach working on such a model. What is basically the same problem is also introduced at the same time in [16], where the authors propose three mixed integer linear programming models, one of which is arc-based and the other two are set covering-based, together with a branch-and-price approach based on one of the set covering-based models. A heuristic version of the branch-and-cut method is also discussed, targeting larger instances. A more realistic variation of the PDSTSP, which we will refer to as the MC-PDSVRP (*Min-Cost Parallel Drone Scheduling Vehicle Routing Problem*) is introduced in [11]. In this version of the problem concepts such as capacity, load balancing and decoupling of costs and times are taken into account. A formal definition of the problem will be provided in Section 3. The authors propose a mixed integer linear programming model and a ruin&recreate metaheuristic for the problem.

In this paper we aim to explore the potential of constraint programming on PDSVRPs, in the hope of exploiting the recent advances of solvers in dealing with TSP and VRP problems. We present tailored models for the different PDSVRPs considered. Experimental results show that several best-known upper and lower bounds can be obtained by the new models. Based on such results, some considerations about the performance of the different models, related to the characteristics of the benchmark instances considered, are also provided.

The rest of the paper is organized as follows. Section 2 is devoted to the the MT-PDSTSP. The problem is formally defined and two Constraint Programming models are presented. Symmetrically, in Section 3 the MC-PDSVRP is discussed, again presenting Constraint Programming models. Experimental results for both the models are discussed in Section 4, while Section 5 contains some conclusions and ideas for future work.

2. The Min-Time Parallel Drone Scheduling Vehicle Routing Problem

The MT-PDSVRP can be represented on a complete directed graph $G = (V, A)$, where the node set $V = \{0, 1, \ldots, n\}$ represents the depot (node 0) and a set of customers $C = \{1, \ldots, n\}$ to be serviced. A set T of homogeneous trucks and a set D of homogeneous drones are available to deliver parcels to the customers. Each truck starts from the depot 0, visits a subset of the customers, and returns back to the depot, operating a single route. The drones operate back and forth trips from the depot to customers, delivering one parcel per trip and operating multiple routes if necessary. Not all the customers can be served

Journal Pre-proof

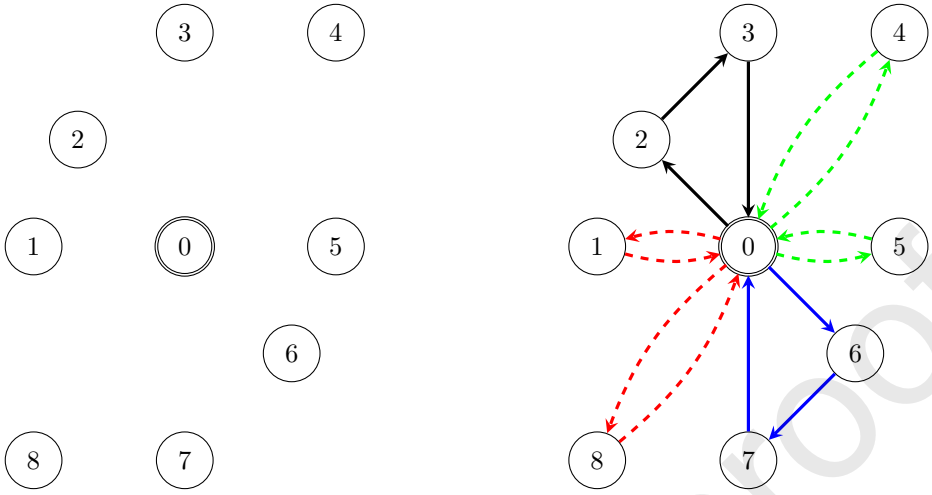


Figure 1: An example of an instance is provided on the left; we assume that two trucks and two drones are available (travel times are omitted for the sake of simplicity). A solution of the MT-PDSVRP is provided on the right. The black and blue arcs represent the tour of the two truck. The first visits nodes 2 and 3 before going back to the depot, the second visits nodes 6 and 7 and goes back to the depot. The dashed red arcs depict the missions of the first drone (that serves nodes 1 and 8), while the dashed green arcs depict the missions of the second drone (that serves nodes 4 and 5).

by a drone, due to the weight of the parcel, an excessive distance of the customer location from the depot, or eventual terrain obstacles such as hills or areas with high-rise buildings. Let $C^D \subseteq C$ denotes the set of customers that can be served by drones. These customers are referred to as *drone-eligible* in the remainder of the paper. The travel time incurred by a truck to go from node i to node j is denoted as t^T_{ij}, while the time required by a drone to serve a customer i (back and forth) is denoted as t^D_i. The trucks and the drones start from the depot at time 0, and the objective of the MT-PDSVRP is to minimize the time required to complete all the deliveries and to have all the trucks and all the drones back at the depot. Note that since truck and drones work in parallel, the objective function translates into minimizing the time required by the vehicle of the fleet with the longest total operational time.

An example of a solution of the MT-PDSVRP for a small instance is provided in Figure 1.

2.1. Constraint programming models

We propose two alternative models that exploit different functions made available by modern Constraint Programming solvers. In the first model the truck tours are represented as separated non-overlapping entities, while in the second they are modelled similarly to the classic giant tour representation [9].

Journal Pre-proof

2.1.1. MT-3IDX: A model based on separated truck tours

The variables of the model that will be referred to as MT-3IDX are as follows. The binary variable z_{kij}, with $k \in T$ and $i, j \in V$ takes value 1 (or equivalently *True*) if node i is visited right before node j in the tour of truck k, and value 0 (or equivalently *False*) otherwise. The binary variable x_{di}, with $d \in D$ and $i \in C^D$, takes value 1 is the customer i is visited by the drone d, and value 0 otherwise. Finally, the variable α is a continuous variable introduced to implement the min max objective function. The model is as follows.

$$MT{-}3IDX: \quad \min \alpha \tag{1}$$

$$s.t. \quad \text{Circuit}(z_{kij}; i, j \in V; i \neq 0 \vee j \neq 0) \qquad k \in T \tag{2}$$

$$\sum_{d \in D} x_{di} + \sum_{k \in T} \neg z_{kii} = 1 \qquad i \in C^D \tag{3}$$

$$\sum_{k \in T} \sum_{i \ inV} \neg z_{kij} = 1 \qquad j \in C \setminus C^D \tag{4}$$

$$\alpha \geq \sum_{i \in V} \sum_{j \in V, i \neq j} t^T_{ij} z_{kij} \qquad k \in T \tag{5}$$

$$\alpha \geq \sum_{i \in C^D} t^D_i x_{di} \qquad d \in D \tag{6}$$

$$z_{kij} \in \{0; 1\} \qquad k \in T; i, j \in V \tag{7}$$

$$x_{di} \in \{0; 1\} \qquad d \in D, i \in C^D \tag{8}$$

The objective function (1) minimizes α, that will be assigned the time taken by the latest vehicle to complete its tasks. Constrains (2) assures that the z variables associated to each truck k take values in such a way to form a valid tour, eventually with self-loops for the variables associated to customers visited by the drones. This is imposed through the use of the "*Circuit*" statement. In the logic of such a command, if a customer i is not visited by the a truck k then z_{kii} is set to 1. Constrains (3) imposes that if a customer $i \in C^D$ is not visited by any truck, then one of the drones in D must visit it. Note that the negation operator "¬" is used, and that if $z_{kii} = 1 \ \forall k \in T$, then customer i is not visited by any truck. Constraints (4) imposes that if a customer $i \in C \setminus C^D$ then one truck in T must visit it. Inequality (5) constraints α to be at least as large as the length of the tour of each truck. Constrains (6) set α to be at least as large as the time spent by each drone d to execute the tasks assigned to it. Finally, constrains (7) and (8) define the variables domain.

This model presents the drawback of having a considerable number of variables, in the order of $O(|T| \cdot |V|^2 + |D| \cdot |V|)$. Moreover, these variables are also constrained to take mutually-consistent values. This might affect the quality of the lower bounds in some circumstances.

2.1.2. MT-2IDX: A model based on a giant truck tour

In this section we present a model with less variables than that discussed in Section 2.1.1. This is achieved by exploiting the "*MultipleCircuit*" statement

Journal Pre-proof

present in modern Constraint Programming tools [15].

The meaning of variables x remain the same as in Section 2.1.1, while z variables are substituted by a new set of variables y, with y_{ij}, $i, j \in V$, that for $i \neq j$ takes value 1 if node i is visited right before node j in one of the truck tours, and value 0 otherwise. When $i = j$, y_{ii} takes value 1 if node i is visited by a drone, 0 otherwise. A new set of continuous variables γ_i, $i \in V$ is introduced to account for the operating time of each truck $k \in T$. The resulting model is as follows.

$$MT{-}2IDX: \quad \min \alpha \tag{9}$$

$$s.t. \quad \text{MultipleCircuit}(y_{ij}, i, j \in V; i \neq 0 \vee j \neq 0) \tag{10}$$

$$\sum_{j \in V \setminus \{0\}} y_{0j} \leq |T| \tag{11}$$

$$\sum_{d \in D} x_{di} + \neg y_{ii} = 1 \qquad i \in C \tag{12}$$

$$\gamma_0 = 0 \tag{13}$$

$$y_{ij} \implies \gamma_j = \gamma_i + t^T_{ij} \qquad i \in V; j \in V \setminus \{0\} \tag{14}$$

$$y_{i0} \implies \alpha \geq \gamma_i + t^T_{i0} \qquad i \in V \setminus \{0\} \tag{15}$$

$$\alpha \geq \sum_{j \in C^D} t^D_j x_{dj} \qquad d \in D \tag{16}$$

$$y_{ij} \in \{0; 1\} \qquad i, j \in V \tag{17}$$

$$x_{di} \in \{0; 1\} \qquad d \in D, i \in C^D \tag{18}$$

$$\gamma_i \geq 0 \qquad i \in V \tag{19}$$

The objective function (9) minimizes α, that will be assigned the time taken by the latest vehicle to complete its tasks. Constrains (10) assures that the y variables take values in such a way to form a set of valid truck tours, and eventually self-loops for the variables associated to customers visited by the drones. This is imposed through the use of the "*MultipleCircuit*" statement. Constrain (12) states that the number of truck tours (outgoing arcs from the depot 0) is limited to $|T|$. Constrains (11) imposes that if each customer $i \in C$ must be visited either by a truck ($y_{ii} = 0$) or by one of the drones in D. Constraint (13) initializes to 0 the time counter at the depot. Constraints (14) increase the operational time of each truck (by setting the time each customer i is visited). Inequalities (15) force α to be at least as large as the last return time of a truck to the depot. Constrains (16) counts the total operating time of each drone d and imposes that α must be greater than it. Finally, constrains (17), (18) and (19) define the domain of the variables.

With respect to the model discussed in 2.1.1, MT-2IDX has the advantage of having less variables, with those used to model the truck tours which are reduced to $O(|V|^2 + |D| \cdot |V|)$ (the factor $|T|$ is not present anylonger). However, the new model requires extra new constraints in the order of $O(|V|^2)$ and extra new variables in the order of $O(|V|)$ to account for the length of each single truck

Journal Pre-proof

tour.

3. The Min-Cost Parallel Drone Scheduling Vehicle Routing Problem

The MC-PDSVRP is a modification of the MT-PDSVRP described in Section 2, where more realistic components are considered, leading to a model with more complex objective function and more constraints. In particular, the following elements are added to concepts already analyzed for the MT-PDSVRP and have to be taken into account: A transportation cost c_{ij}^T is incurred by a truck to travel on the edge $(i,j) \in E$, while c_i^D is the transportation costs incurred by each drone to complete a mission to customer $i \in C^D$. These costs are used to define a new objective function to be minimized. The delivery request from each customer $i \in C$ is associated with a parcel weight w_i. Each truck has to respect some capacity constraints, being limited to transport at most Q^T units of weight in its route. Finally, there are upper bounds τ^T and τ^D on the maximum working time for each truck and drone, respectively. Note that these latter constraints are normally used to introduce load-balancing concepts into the optimization.

3.1. Constraint programming models

Analogously to what seen in Section 2.1 for the MT-PDSVRP, two models will be presented.

3.1.1. MC-3IDX: A model based on separated truck tours

The variables used in this model are the same used for the model MT-3IDX in Section 2.1.1.

$$MC{-}3IDX: \quad \min \sum_{k\in T}\sum_{i\in V}\sum_{j\in V, j\neq i} c_{ij}^T z_{kij} + \sum_{k\in D}\sum_{i\in V} c_i^D x_{ki} \tag{20}$$

$$s.t. \quad \text{Circuit}(z_{kij}; i,j \in V) \qquad k \in T \tag{21}$$

$$\sum_{d\in D} x_{dj} + \sum_{k\in T}\sum_{i\ inV} \neg z_{kij} = 1 \qquad j \in C^D \tag{22}$$

$$\sum_{k\in T}\sum_{i\ inV} \neg z_{kij} = 1 \qquad j \in C \setminus C^D \tag{23}$$

$$\sum_{i\in V}\sum_{j\in V, i\neq j} t_{ij}^T z_{kij} \leq \tau^T \qquad k \in T \tag{24}$$

$$\sum_{j\in C^D} t_j^D x_{dj} \leq \tau^D \qquad d \in D \tag{25}$$

$$\sum_{j\in C} w_j z_{kij} \leq Q^T \qquad k \in T \tag{26}$$

$$z_{kij} \in \{0;1\} \qquad k \in T, i,j \in V \tag{27}$$

$$x_{di} \in \{0;1\} \qquad d \in D, i \in C^D \tag{28}$$

Journal Pre-proof

The objective function (20) minimizes the total cost, given by sum of the costs of each truck and drone, incurred to service all the customers. Constrains (21) assures that the z variables associated to each truck k take values in such a way to form a valid tour, eventually self-loops for the variables associated to customers visited by the drones. Constraints (22) imposes that if a customer $i \in C^D$ is not visited by any truck, then one of the drones in D must visit it. Constraints (23) imposes that if a customer $i \in C \setminus C^D$ then one truck in T must visit it. Inequalities (24) make sure that each truck does not exceed its maximum working time. Inequalities (25) are the constraints on maximum working time for the drones. Constraints (26) model the capacity constraints for the trucks. Finally, constraints (27) and (28) define the domain of the variables.

The more realistic settings of this problem with respect to the Min-Time version discussed in Section 2.1.1, imply a newly designed objective function and $O(|T| + |D|)$ new constraints to account for budget limitations.

3.1.2. MC-2IDX: A model based on a giant truck tour

The variables used in this model are the same used for the model MT-2IDX in Section 2.1.2, with the addition of the new set of continuous variables β_i, $i \in V$ which is introduced to count the weight carried by each truck $k \in T$.

$$MC{-}2IDX: \quad \min \sum_{i \in V} \sum_{j \in V, j \neq i} c^T_{ij} y_{ij} + \sum_{d \in D} \sum_{i \in V} c^D_i x_{di} \tag{29}$$

$$s.t. \quad \text{MultipleCircuit}(y_{ij}, i, j \in V; i \neq 0 \vee j \neq 0) \tag{30}$$

$$\sum_{d \in D} x_{di} + \neg y_{ii} = 1 \qquad i \in C \tag{31}$$

$$\sum_{j \in V \setminus \{0\}} y_{0j} \leq |T| \tag{32}$$

$$\sum_{j \in C^D} t^D_j x_{dj} \leq \tau^D \qquad d \in D \tag{33}$$

$$\beta_0 = 0 \tag{34}$$

$$y_{ij} \implies \beta_j = \beta_i + w_j \qquad i \in V; j \in V \setminus \{0\} \tag{35}$$

$$\beta_i \leq Q^T \qquad i \in V \setminus \{0\} \tag{36}$$

$$\gamma_0 = 0 \tag{37}$$

$$y_{ij} \implies \gamma_j = \gamma_i + t^T_{ij} \qquad i \in V; j \in V \setminus \{0\} \tag{38}$$

$$y_{i0} \implies \gamma_i + t^T_{i0} \leq \tau^T \qquad i \in V \setminus \{0\} \tag{39}$$

$$y_{ij} \in \{0; 1\} \qquad i, j \in V \tag{40}$$

$$x_{di} \in \{0; 1\} \qquad d \in D, i \in C^D \tag{41}$$

$$\beta_i, \gamma_i \geq 0 \qquad i \in V \tag{42}$$

The objective function (29) minimizes the total cost, given by sum of the costs of each truck and drone, incurred to service all the customers. Constrains (30)

Journal Pre-proof

assures that the y variables take values in such a way to form a set of valid truck tours, and eventually self-loops for the variables associated to customers visited by the drones. Constrains (31) imposes that if each customer $i \in C$ must be visited either by a truck ($y_{ii} = 0$) or by one of the drones in D. Constrain (32) states that the number of truck tours (outgoing arcs from the depot 0) is limited to $|T|$. Constrains (33) limits the total operating time of each drone d to the maximum allowed value τ^D. Constraint (34) initializes to 0 the weight counter at the depot. Constraints (35) increase the weight transported by each truck. Inequalities (36) impose that the incremental weight β_i at each customer $i \in V$ can never exceed the maximum capacity Q^T of each truck. Constraint (37) initializes to 0 the time counter at the depot. Constraints (38) increase the operational time of each truck (by setting the time each customer i is visited). Inequalities (39) impose that the incremental time γ_i at each customer $i \in V$, plus the time required to the truck to go back to the depot, can never exceed the maximum operating time τ^T of each truck. Finally, constrains (40), (41) and (42) define the variables domain.

When comparing the number of variables and constraints of the new model with those of MC-3IDX, an analysis similar to that discussed in Section 2.1.2 for the Min-Time variation of the problem can be sketched: the number of variables is reduced from $O(|T| \cdot |V|^2 + |D| \cdot |V|)$ to $O(\cdot|V|^2 + |D| \cdot |V|)$, but extra constraints, in the order of $O(|V|^2)$ are in this case required to model the budget constraints for the trucks.

4. Computational experiments

The constraint programming models described in Sections 2 and 3 have been implemented in Python 3.9 and solved via the CP-SAT solver of Google OR-Tools 9.5.2237 [15]. The experiments have been run on a computer equipped with 32 GB of RAM and an Intel Core i7 12700F CPU with 12 cores (8 with a maximum frequency of 4.9 GHz and 4 with a maximum frequency of 3.6 GHz). Note that in the experiments reported in the remainder of this section, a completely fair comparison among the different methods proposed is not possible, due to the different experimental conditions (programming language used, length and multiplicity of the runs, hardware and number of cores used). However, and notwithstanding this premise, the potentialities of the different approaches clearly emerge from the experiments.

The outcome of the experimental champaign we run is discussed in the remainder of this section, and is organized according to the different problems attacked. All the tables present the information of the instances and for each method considered, the maximum computation time allowed is reported (in the label of the columns), although the hardware is different for each method. For each instance/method combination, the cost of the best heuristic solution retrieved, and eventually the lower bound found are reported. A dash means no result was retrieved in the given time (or - for some of the methods we compare with - the experiment was not attempted). Every time one of the new models

Journal Pre-proof

we propose improve a best-known bound, the relative entry is in bold. Analogously, every time one of the new models we propose does not match or improve a best-known bound, the relative entry is in italic. Finally, proven optimal solutions retrieved by any of the methods considered, are marked with an asterisk in the tables. All the new best-known solutions retrieved in the present work are available upon request to the corresponding author.

4.1. MT-PDSVRP

The results are subdivided based on the source of the instances in the following subsections.

4.1.1. Instances from Mbiadou Saleu et al. [7]

A first set of benchmarks for the MT-PDSVRP was created in [7] starting from classic instances for the *Capacitated Vehicle Routing Problem*. A total of 20 instances with a number of customers ranging between 50 and 199 has been obtained. Note that for each instance considered here and in the remainder of the paper, the number of nodes of each instance can be inferred by its name (or it is written explicitly if this is not the case. We refer the interested reader to [7] for full details about the elements of the new instances and the complete sources for the instances. The outcome of the experiments is summarized in Table 1, where the results are those obtained by a branch and cut (BC) approach running on a MILP model and those of the best of nine variations of a hybrid metaheuristc approach (HM). Both the methods are discussed in [7]. The BC solver was run on a computer with an Intel Xeon CPU E5-2670 CPU with 2x8 cores running at 2.6 GHz and 62.5 GB of RAM, with a maximum computation time of 10800 seconds, while the HM variations was run on a computer with an Intel core i5-6200U CPU running at 2.4 GHz and 8 GB of RAM, with a maximum computation time of 1000 seconds for each variation. The solver was run on the models discussed in Section 2.1 for a maximum of 10800 seconds.

The results presented in Table 1 indicate that the new CP-based approaches are competitive with state-of-the-art methods. In particular, new improved lower bounds were provided for all the instances considered, and 2 new best-known heuristic solutions were also retrieved. The comparison between the new CP models indicate that the model MT-3IDX discussed in Section 2.1.1, and characterized by the set of variables z with 3 indices, performs better. In particular, it provides substantially tighter lower bounds. The given instances are not too large, so the advantage of MT-3IDX can be motivated by the intrinsical simplicity of the model, that does not require extra-variables and constraints to measure the length of each truck tour. However, MT-3IDX was not capable of handling the last instance, which by its nature allows a large number of possible drone missions, and consequently a large number of z variables. In this case, the model MT-2IDX, with its smaller memory footprint, is the only viable option to have a good lower bound. It is interesting also to observe that the quality of the heuristic solutions produced by the CP methods is constantly high, although not always matching the state-of-the-art provided by the best of nine purely heuristic methods, the results of which are those summarized in column HM.

Journal Pre-proof

Table 1: MT-PDSVRP - Instances originated from CVRPLIB in Mbiadou Saleu et al. [7].

Instances			Mbiadou Saleu et al. [7]			MT-3IDX		MT-2IDX	
			BC (10800 sec)		HM (9x1000 sec)	(10800 sec)		(10800 sec)	
Name	$\|T\|$	$\|D\|$	LB	UB	UB	LB	UB	LB	UB
CMT1 (51 nodes)	3	2	145.87	188.00	166.00	**158.27**	*168.00*	*103.25*	*174.00*
CMT2 (76 nodes)	5	5	101.66	3630.86	130.23	**108.00**	*158.00*	*65.33*	*144.00*
CMT3 (101 nodes)	4	4	161.07	4537.11	184.00	**165.98**	*222.00*	*98.07*	*210.00*
CMT4 (151 nodes)	6	6	-	-	160.38	**125.39**	*288.00*	*72.73*	*180.00*
CMT5 (200 nodes)	9	8	-	-	138.00	**82.84**	*312.00*	*55.67*	*158.00*
E-n51-k5	3	2	145.87	188.00	168.00	**159.34**	**166.00**	*103.22*	*174.00*
E-n76-k8	4	4	127.05	2975.51	154.00	**132.49**	*180.00*	*81.13*	*164.00*
E-n101-k8	4	4	161.07	4537.11	184.00	**165.64**	*244.00*	*98.03*	*224.00*
M-n151-k12	6	6	-	-	154.00	**126.57**	*270.00*	*72.78*	*188.00*
M-n200-k16	8	8	-	-	144.00	**95.21**	*302.00*	*59.19*	*166.00*
P-n51-k10	5	5	81.35	230.00	111.07	**90.13**	*118.00*	*52.60*	*124.00*
P-n55-k7	4	3	101.49	308.00	126.00	**111.00**	*136.00*	*70.40*	*130.00*
P-n60-k10	5	5	84.30	246.00	114.00	**95.00**	*124.00*	*54.89*	*120.00*
P-n65-k10	5	5	94.66	580.00	126.00	**105.41**	*144.00*	*61.46*	*138.00*
P-n70-k10	5	5	99.42	3166.25	128.00	**108.00**	*164.00*	*64.34*	*150.00*
P-n76-k5	3	2	181.36	280.00	200.00	**189.91**	*206.00*	*129.06*	*214.00*
P-n101-k4	2	2	321.80	4725.47	342.00	**329.90**	**340.00**	*198.00*	*362.00*
X-n110-k13	7	6	-	-	1864.00	**1365.68**	*2702.00*	*766.70*	*2048.00*
X-n115-k10	5	5	-	-	2258.00	**1850.38**	*2920.00*	*1031.21*	*2488.00*
X-n139-k10	5	5	-	-	2492.00	**-**	-	**1118.67**	*2744.00*

4.1.2. Instances from Raj et al. [16]

A second set of instances for the MT-PDSTSP was originated from TSPLIB [17] in [16], by defining the missing elements according to what done previously in [6] for the PDSTSP. Only the instances with more than one truck are reported here, since those with one truck boil down the problem to a PDSTSP, and all the optimal solutions can be found in [8]. The number of customers range between 48 and 229. Full details about the instances can be found in [16]. Three methods are considered for comparison purposes, all discussed in [16]: an arc-based MILP model; a branch and price (BP) method based on a set covering MILP model, and a math-heuristic method (MH) based on the same latter model. All the experiments for these methods were run on a computer equipped with an Intel i7-6700 CPU (with 8 cores running at 4 GHz) and 16 GB of RAM, with 3600 seconds for MILP and MH, and 2400 seconds for BC. The CP-based solvers we propose were run for a maximum time of 3600 seconds. The results are summarized in Tables 2 and 3.

The results of Table 2 indicates that the CP model MT-3IDX is very suitable for these instances, being able to provide all the state-of-the-art lower bounds and all the best-known heuristic solutions except one, with substantial improvements over previously known best results. It is remarkable also that five instances were closed here for the first time. The performance of the model MT-2IDX appears less brilliant, in particular on the larger instances considered, on which no solution or meaningful bound was produced in the given time. An explanation for this unwanted situation can be found in the use of the com-

Journal Pre-proof

Table 2: MT-PDSVRP - Instances with 2 trucks originated from TSPLIB in Raj et al. [16].

Instances			Raj et al. [16]			MT-3IDX (3600 s)		MT-2IDX (3600 s)	
			MILP (3600 s)	BP (2400 s)	MH (3600 s)				
Name	$\|T\|$	$\|D\|$	UB	UB	UB	LB	UB	LB	UB
att48_0_80	2	2	18206.00	17032.00	17056.00	**16940.18**	**16940.18***	9785.69	17288.00
att48_0_80	2	4	16730.00	16500.00	16500.00	**16500.00**	16500.00*	6562.48	16564.00
att48_0_80	2	6	16500.00	16500.00	16500.00	**16500.00**	16500.00*	4980.66	16500.00
berlin52_0_80	2	2	3328.20	3415.00	3415.00	**3285.00**	**3285.00***	2240.63	3480.00
berlin52_0_80	2	4	2995.00*	2995.00	2995.00	2995.00	2995.00*	1498.57	2995.00
berlin52_0_80	2	6	2995.00*	2995.00	2995.00	2995.00	2995.00*	1141.20	2995.00
eil101_0_80	2	2	392.00	316.00	305.40	**293.00**	**293.00***	197.44	331.00
eil101_0_80	2	4	408.00	266.80	253.70	**232.84**	**248.00**	131.60	*283.00*
eil101_0_80	2	6	320.00	219.00	215.30	**198.00**	**212.00**	99.15	*272.00*
gr120_0_80	2	2	1145.00	806.30	764.00	**703.50**	**735.00**	478.18	*866.00*
gr120_0_80	2	4	1092.00	676.00	646.00	**590.54**	*676.00*	318.78	*777.00*
gr120_0_80	2	6	1082.00	769.00	581.20	**518.00**	**549.00**	239.10	*688.00*
pr152_0_80	2	2	122961.00	55477.00	48967.00	**38873.80**	**41371.00**	18792.17	*46897.00*
pr152_0_80	2	4	-	54980.00	52353.00	**33593.10**	**41295.58**	12533.75	*44059.00*
pr152_0_80	2	6	-	50855.40	50854.40	**31020.74**	**37413.00**	9503.63	*43573.00*
gr229_0_80	2	2	-	1403.70	1403.70	**881.91**	**1064.97**	0.00	**-**
gr229_0_80	2	4	-	1346.20	1346.20	**818.33**	**1064.97**	0.00	-
gr229_0_80	2	6	-	1327.60	1327.60	**776.98**	**1056.10**	0.00	-

* Optimality proven.

mand "*MultipleCircuit*": it is designed to effectively handle instances up to 150 customers maximum and not too many trucks, while the largest instances have here up to 229 customers and up to 9 trucks.

The results of Table 3 confirm the previous impressions. Only, in this case the CP model MT-3IDX otperforms the other methods less clearly. This indicates a degradation in the performance when the number of trucks considered increases. This is due to the coordination required by the x variables corresponding to different tours (this function is completely embedded in the "*MultipleCircuit*" command at the basis of model MT-2IDX). On a few instances MT-3IDX found sub-optimal heuristic solutions, although it remarkably provided an optimality proof for the first instance. The model MT-2IDX was able to produce for these instances two new state-of-the-art heuristic solutions, indicating that this latter method might be indicated for problems with more trucks available, especially if the number of customers is not too large, and anyway within certain limits. If the number of trucks becomes too large, issues are however incurred by the model MT-2IDX (see Table 1).

4.2. MC-PDSVRP

The instances considered in this section are those originally proposed in Nguyen et al. [11]. They are based on several statistical data shared by logistics service providers and common practices, so although being generated as random, they can be considered as realistic. The numbers of customers considered are 30, 50, 100, 200 and 400 (the first element of the name of each instance reflects

Journal Pre-proof

Table 3: MT-PDSVRP - Instances with 3 trucks originated from TSPLIB in Raj et al. [16].

Instances			Raj et al. [16]			MT-3IDX (3600 s)		MT-2IDX (3600 s)	
			MILP (3600 s)	BP (2400 s)	MH (3600 s)				
Name	$\|T\|$	$\|D\|$	UB	UB	UB	LB	UB	LB	UB
att48_0_80	3	2	19824.00	15218.00	14062.00	**13452.00**	**13452.00**[*]	7811.15	*14652.00*
att48_0_80	3	4	17194.00	13394.00	14652.00	**10605.21**	13394.00	5690.68	13394.00
att48_0_80	3	6	16226.00	13756.00	13394.00	**10369.78**	13394.00	4441.85	13394.00
berlin52_0_80	3	2	3205.00	3403.10	2995.00	**2471.35**	**2935.00**	1777.81	2995.00
berlin52_0_80	3	4	2665.00	2635.00	2625.00	**2013.82**	2625.00	1285.27	*2925.00*
berlin52_0_80	3	6	2625.00	2625.00	2625.00	**2132.20**	2625.00	1015.52	*2925.00*
eil101_0_80	3	2	-	260.00	235.00	**209.00**	*252.00*	157.04	*278.00*
eil101_0_80	3	4	-	221.00	200.10	**177.00**	**198.00**	112.23	*227.00*
eil101_0_80	3	6	-	228.00	181.00	**155.20**	*218.00*	87.91	*204.00*
gr120_0_80	3	2	-	611.00	597.00	**509.50**	*688.00*	381.29	*734.00*
gr120_0_80	3	4	-	536.00	528.00	**444.00**	*688.00*	272.58	*601.00*
gr120_0_80	3	6	-	590.90	527.20	**397.50**	**526.00**	212.84	*547.00*
pr152_0_80	3	2	-	48135.00	48135.00	**25154.41**	38395.00	15033.25	**36443.00**
pr152_0_80	3	4	-	43024.00	43024.00	**21913.95**	**36940.00**	10799.23	37183.00
pr152_0_80	3	6	-	46403.00	46403.00	**20584.69**	36940.00	8526.30	**36281.00**
gr229_0_80	3	2	-	1145.40	1145.40	**579.62**	**981.59**	0.00	-
gr229_0_80	3	4	-	1132.30	1132.30	**561.47**	**969.92**	0.00	-
gr229_0_80	3	6	-	1130.00	1130.00	**517.37**	**1004.89**	0.00	-

[*] Optimality proven.

this information). We refer the reader to [11] for the full details of the instances (loading capacities, speeds, battery endurance for the drones, etc).

The results of our experiments are summarized in Tables 4, 5, 6, 7 and 8, divided according the size of the instances. On top of the CP models discussed in Section 3.1, that are run for a maximum of 1800 seconds, the methods considered in the comparison are those presented in [11], namely a MILP model and *Ruin and Recreate* heuristic algorithm (RR). The experiments for the latter methods were run on a computer equipped with an AMD Ryzen 3700X CPU running at 4.0 GHz and 16 GB of RAM, with a time limit of 3600 seconds. Note that the RR method was run 30 times, and the best result is reported here.

The results reported in Tables 4 and 5 suggest that the MILP model is the most viable method to solve the small problems with 30 and 50 customers: several instances are solved to optimality within the given time. However, on some of the instances with 50 customers, MILP is not able to provide any solution. The RR heuristic also obtains remarkably good results, always matching or improving the best known figures. We believe that taking the best of 30 runs was an important factor in these results, because this allows a good random exploration of the search space. The CP models performed reasonably well on these instances in terms of heuristic solution found, and were able to provide the first non-trivial lower bounds (no lower bound was provided in previous publications). It is worth noticing that the model MC-2IDX seems to perform better than MC-3IDX on these small *Minimum Cost* instances, especially on the instances with 50 customers. This observation is discordant with respect

Journal Pre-proof

Table 4: MC-PDSVRP - Instances with 30 customers from Nguyen et al. [11].

Instances			Nguyen et al. [11]		MC-3IDX		MC-2IDX	
			MILP (3600 sec)	RR (30x3600 sec)	(1800 sec)		(1800 sec)	
Name	$\|T\|$	$\|D\|$	UB	UB	LB	UB	LB	UB
30-r-0-c	10	4	128.67*	128.67	101.69	128.67	*103.11*	128.67
30-r-0-e	10	4	300.85	300.85	104.85	300.85	**150.61**	300.85
30-r-0-r	10	4	199.76	199.76	110.71	199.76	**125.00**	199.76
30-r-1-c	10	4	58.30*	58.30	58.30	58.30*	58.30	58.30*
30-r-1-e	10	4	243.11	243.11	117.48	243.11	**148.43**	243.11
30-r-1-r	10	4	164.75	164.75	106.82	164.75	**116.50**	164.75
30-r-2-c	10	4	91.57*	91.57	69.04	91.57	*80.43*	91.57
30-r-2-e	10	4	235.25	235.25	111.62	235.25	**151.10**	235.25
30-r-2-r	10	4	211.26	211.26	101.11	211.26	**135.88**	211.26
30-c-0-c	10	4	82.15*	82.15	*60.08*	82.15	82.15	82.15*
30-c-0-e	10	4	150.44*	150.44	*103.78*	150.44	*110.23*	150.44
30-c-0-r	10	4	112.72*	112.72	*83.08*	112.72	*91.44*	112.72
30-c-1-c	10	4	39.98*	39.98	39.98	39.98*	39.98	39.98*
30-c-1-e	10	4	172.49	172.49	**97.03**	172.49	83.94	172.49
30-c-1-r	10	4	61.28*	61.28	61.28	61.28*	61.28	61.28*
30-c-2-c	10	4	77.00*	77.00	77.00	77.00*	77.00	77.00*
30-c-2-e	10	4	166.82	166.82	**98.49**	166.82	95.89	166.82
30-c-2-r	10	4	88.81*	88.81	*59.65*	88.81	*80.22*	88.81
30-rc-0-c	10	4	81.06*	81.06	81.06	81.06*	*74.47*	81.06
30-rc-0-e	10	4	181.89	181.89	111.08	181.89	**115.78**	181.89
30-rc-0-r	10	4	146.73	146.73	88.40	146.73	**98.85**	146.73
30-rc-1-c	10	4	56.48*	56.48	56.48	56.48*	56.48	56.48*
30-rc-1-e	10	4	176.57	176.57	115.70	176.57	**118.09**	176.57
30-rc-1-r	10	4	97.91*	97.91	97.91	97.91*	*71.96*	97.91
30-rc-2-c	10	4	77.00*	77.00	*63.74*	77.00	77.00	77.00*
30-rc-2-e	10	4	187.18	187.18	71.50	187.18	**104.96**	187.18
30-rc-2-r	10	4	94.21*	94.21	*77.67*	94.21	*86.67*	94.21

* Optimality proven.

to what was observed in Section 4.1 for the *Minimum Time* instances, but an explanation can be given. For the *Minimum Cost* problem, both the models require variables and constraints to handle capacity and load balancing constraints. This makes the MC-3IDX model less attractive, since the simplicity that characterized it for the *Minimum Time* problem (see Section 2) is partially compromised here. For this reason, the command "*MultipleCircuit*" becomes attractive on these small/medium instances, and the results show that the choice pays off.

The results summarized in Tables 6, 7 and 8 do not include the results of the MILP model, since these larger instances with 100, 200 and 400 customers were out of reach for it. The RR heuristic produces good-quality solutions, as we were able to certify here with the first lower bounds ever produced on these instances. On the other hand, the relatively small gaps between upper and lower bounds on some of the instances, also suggest that the quality provided by the CP models are good. In details, MC-2IDX scaled up well, while MC-3IDX

Journal Pre-proof

Table 5: MC-PDSVRP - Instances with 50 customers from Nguyen et al. [11].

Instances			Nguyen et al. [11]		MC-3IDX (1800 sec)		MC-2IDX (1800 sec)	
			MILP (3600 sec)	RR (30x3600 sec)				
Name	$\|T\|$	$\|D\|$	UB	UB	LB	UB	LB	UB
50-r-0-c	10	4	161.13*	161.13	*110.62*	161.13	*131.27*	*161.56*
50-r-0-e	10	4	-	310.50	129.68	310.50	**151.57**	310.50
50-r-0-r	10	4	-	246.53	110.02	246.53	**137.85**	246.53
50-r-1-c	10	4	177.21	177.21	103.69	177.21	**135.27**	177.21
50-r-1-e	10	4	216.97	216.97	129.97	216.97	**153.12**	216.97
50-r-1-r	10	4	-	228.11	116.45	228.11	**148.57**	228.11
50-r-2-c	10	4	182.09	181.49	108.59	182.62	**145.83**	*182.62*
50-r-2-e	10	4	-	375.05	127.88	375.05	**153.27**	375.05
50-r-2-r	10	4	-	313.32	109.95	313.32	**162.11**	313.32
50-c-0-c	10	4	89.04	89.04	47.73	89.04	**78.64**	89.04
50-c-0-e	10	4	183.26	182.91	78.01	182.91	**105.82**	182.91
50-c-0-r	10	4	112.17	112.17	85.09	112.17	**89.09**	112.17
50-c-1-c	10	4	81.76*	81.76	*39.38*	81.76	*75.97*	81.76
50-c-1-e	10	4	133.57	133.57	81.44	133.57	**92.35**	133.57
50-c-1-r	10	4	149.24	148.86	56.78	*149.06*	**83.95**	*149.06*
50-c-2-c	10	4	77.16*	77.16	*54.26*	77.16	*75.31*	77.16
50-c-2-e	10	4	214.25	214.25	74.24	*214.66*	**112.86**	*214.66*
50-c-2-r	10	4	136.60	136.51	67.30	136.51	**90.66**	136.51
50-rc-0-c	10	4	116.80*	116.80	*70.01*	116.80	*107.92*	116.80
50-rc-0-e	10	4	-	242.02	74.24	242.02	**124.28**	242.02
50-rc-0-r	10	4	217.87	217.87	95.88	217.87	**109.28**	217.87
50-rc-1-c	10	4	110.44*	110.44	*82.98*	110.44	*104.16*	110.44
50-rc-1-e	10	4	201.62	201.62	112.97	201.62	**124.27**	201.62
50-rc-1-r	10	4	173.16	173.16	90.57	*173.61*	**116.16**	*173.61*
50-rc-2-c	10	4	129.82	129.65	64.33	129.82	**104.74**	129.65
50-rc-2-e	10	4	-	383.49	109.51	383.49	**129.50**	383.49
50-rc-2-r	10	4	197.20	186.31	108.11	196.97	**108.82**	186.31

* Optimality proven.

Journal Pre-proof

Journal Pre-proof

Table 6: MC-PDSVRP - Instances with 100 customers from Nguyen et al. [11].

Instances			Nguyen et al. [11]		MC-3IDX (1800 sec)		MC-2IDX (1800 sec)	
			MILP (3600 sec)	RR (30x3600 sec)				
Name	$\|T\|$	$\|D\|$	UB	UB	LB	UB	LB	UB
100-r-0-c	10	5	-	272.77	143.76	-	**222.27**	*277.90*
100-r-0-e	10	5	-	506.07	206.21	-	**250.30**	-
100-r-0-r	10	5	-	325.29	140.62	-	**226.44**	-
100-r-1-c	10	5	-	302.46	167.13	-	**240.03**	-
100-r-1-e	10	5	-	470.06	198.41	-	**251.18**	-
100-r-1-r	10	5	-	350.97	179.63	-	**230.43**	-
100-r-2-c	10	5	-	284.77	169.39	*288.00*	**232.12**	-
100-r-2-e	10	5	-	435.91	208.16	-	**265.24**	-
100-r-2-r	10	5	-	320.98	162.92	*361.47*	**226.69**	-
100-c-0-c	10	5	-	218.59	92.16	*221.59*	**170.73**	-
100-c-0-e	10	5	-	315.84	93.02	-	**175.48**	-
100-c-0-r	10	5	-	252.39	81.02	*276.95*	**161.81**	-
100-c-1-c	10	5	-	154.61	87.41	*179.75*	**112.94**	-
100-c-1-e	10	5	-	222.15	103.33	*236.44*	**119.22**	-
100-c-1-r	10	5	-	110.95	64.71	110.95	**98.77**	110.95
100-c-2-c	10	5	-	108.35	73.34	108.35	**95.81**	-
100-c-2-e	10	5	-	118.71	70.71	*132.28*	**103.69**	-
100-c-2-r	10	5	-	86.28	49.08	*102.47*	**86.00**	*88.01*
100-rc-0-c	10	5	-	252.37	128.09	*254.03*	**199.69**	-
100-rc-0-e	10	5	-	472.95	168.08	-	**233.41**	-
100-rc-0-r	10	5	-	337.61	157.65	*339.09*	**205.28**	-
100-rc-1-c	10	5	-	276.78	152.69	*305.51*	**209.72**	-
100-rc-1-e	10	5	-	432.44	190.81	*490.66*	**246.80**	-
100-rc-1-r	10	5	-	423.33	163.48	*426.76*	**220.13**	-
100-rc-2-c	10	5	-	260.45	125.31	*286.73*	**198.66**	-
100-rc-2-e	10	5	-	358.24	174.37	*406.12*	**228.51**	-
100-rc-2-r	10	5	-	331.97	138.87	*345.25*	**207.27**	-

was not able to produce significant bounds, due to the very large number of z variables necessary to model these instances with 10 trucks and many customers. Unfortunately, the good performance of the model MC-2IDX in terms of lower bounds did not get reflected on the quality of the heuristic solutions: very often the method was not capable of producing any feasible solution at all in the given computation time. We believe this is due to the troubles encountered by the "*MultipleCircuit*" command to handle instances approaching (and going beyond) the limits in terms of number of customers and trucks for which it has been designed.

5. Conclusions

In this paper we have discussed some different Constraint Programming models to describe two versions of the Parallel Drone Scheduling Vehicle Routing Problem that were recently proposed in the literature.

Experimental results suggest that solving these models can lead to improved state-of-the-art results. In particular, the new models appear to be the new

Journal Pre-proof

Table 7: MC-PDSVRP - Instances with 200 customers from Nguyen et al. [11].

Instances			Nguyen et al. [11]		MC-3IDX (1800 sec)		MC-2IDX (1800 sec)	
			MILP (3600 sec)	RR (30x3600 sec)				
Name	$\|T\|$	$\|D\|$	UB	UB	LB	UB	LB	UB
200-r-0-c	10	5	-	430.42	134.99	-	**353.51**	-
200-r-0-e	10	5	-	728.00	0.00	-	**368.88**	-
200-r-0-r	10	5	-	528.29	102.71	-	**355.73**	-
200-r-1-c	10	5	-	434.66	228.69	-	**348.04**	-
200-r-1-e	10	5	-	601.06	0.13	-	**369.88**	-
200-r-1-r	10	5	-	481.83	128.30	-	**349.86**	-
200-r-2-c	10	5	-	446.15	0.00	-	**357.04**	-
200-r-2-e	10	5	-	606.05	0.00	-	**374.19**	-
200-r-2-r	10	5	-	497.72	0.00	-	**350.49**	-
200-c-0-c	10	5	-	234.40	0.00	-	**184.05**	-
200-c-0-e	10	5	-	442.74	0.00	-	**189.35**	-
200-c-0-r	10	5	-	270.76	0.00	-	**173.94**	-
200-c-1-c	10	5	-	225.55	78.21	-	**186.77**	-
200-c-1-e	10	5	-	461.57	0.00	-	**211.23**	-
200-c-1-r	10	5	-	239.91	0.00	-	**198.22**	-
200-c-2-c	10	5	-	263.26	0.00	-	**193.47**	-
200-c-2-e	10	5	-	486.05	0.00	-	**212.55**	-
200-c-2-r	10	5	-	322.61	0.00	-	**189.90**	-
200-rc-0-c	10	5	-	354.23	0.00	-	**280.87**	-
200-rc-0-e	10	5	-	575.94	0.00	-	**297.83**	-
200-rc-0-r	10	5	-	484.66	0.00	-	**293.68**	-
200-rc-1-c	10	5	-	370.01	0.00	-	**280.68**	-
200-rc-1-e	10	5	-	591.58	0.00	-	**296.25**	-
200-rc-1-r	10	5	-	411.34	0.00	-	**267.01**	-
200-rc-2-c	10	5	-	372.30	0.00	-	**299.35**	-
200-rc-2-e	10	5	-	534.45	0.00	-	**330.13**	-
200-rc-2-r	10	5	-	432.17	0.00	-	**299.16**	-

reference for producing high quality lower bounds on the optimal solution costs, but they are also able to produce several new best-known heuristic solutions, and even to close for the first time several of the instances considered. Some scalability issues however emerge on the larger instances, for which the tools provided by the solver to represent truck tours likely reach the limits for which they have been designed. Future work should try to overcome this phenomenon, and to make the models we propose more robust.

In our opinion the flourishing literature on specialization of Parallel Drone Scheduling Vehicle Routing Problems, aiming at introducing more and more realistic aspects could be greatly benefit from our contributions, given the high flexibility provided by Constraint Programming, and the likely assumption that the results we have obtained here could be replicated. This is material for future work.

Journal Pre-proof

Table 8: MC-PDSVRP - Instances with 400 customers from Nguyen et al. [11].

Instances			Nguyen et al. [11]		MC-3IDX (1800 sec)		MC-2IDX (1800 sec)	
			MILP (3600 sec)	RR (30x3600 sec)				
Name	$\|T\|$	$\|D\|$	UB	UB	LB	UB	LB	UB
400-r-0-c	10	5	-	604.94	0.00	-	**500.91**	-
400-r-0-e	10	5	-	801.45	0.00	-	**518.74**	-
400-r-0-r	10	5	-	626.46	0.00	-	**518.62**	-
400-r-1-c	10	5	-	578.77	0.00	-	**487.76**	-
400-r-1-e	10	5	-	772.30	0.00	-	**445.80**	-
400-r-1-r	10	5	-	584.07	0.00	-	**477.13**	-
400-r-2-c	10	5	-	587.88	0.00	-	**483.98**	-
400-r-2-e	10	5	-	771.16	0.00	-	**510.02**	-
400-r-2-r	10	5	-	692.91	0.00	-	**454.20**	-
400-c-0-c	10	5	-	271.82	0.00	-	**207.27**	-
400-c-0-e	10	5	-	378.98	0.00	-	**253.55**	-
400-c-0-r	10	5	-	403.85	0.00	-	**248.69**	-
400-c-1-c	10	5	-	187.20	0.00	-	**136.34**	-
400-c-1-e	10	5	-	249.01	0.00	-	**142.46**	-
400-c-1-r	10	5	-	219.23	0.00	-	**146.12**	-
400-c-2-c	10	5	-	152.79	0.00	-	**141.57**	-
400-c-2-e	10	5	-	245.54	0.00	-	**145.22**	-
400-c-2-r	10	5	-	231.89	0.00	-	**154.86**	-
400-rc-0-c	10	5	-	497.90	0.00	-	**420.36**	-
400-rc-0-e	10	5	-	700.96	0.00	-	**456.69**	-
400-rc-0-r	10	5	-	604.38	0.00	-	**437.29**	-
400-rc-1-c	10	5	-	518.40	0.00	-	**395.26**	-
400-rc-1-e	10	5	-	784.02	0.00	-	**424.46**	-
400-rc-1-r	10	5	-	531.44	0.00	-	**396.43**	-
400-rc-2-c	10	5	-	510.64	0.00	-	**424.09**	-
400-rc-2-e	10	5	-	723.63	0.00	-	**449.39**	-
400-rc-2-r	10	5	-	625.31	0.00	-	**448.43**	-

Acknowledgements

The authors are grateful to Prof. Minh Hoàng Hà for the useful discussions and hints, as well as to the anonymous Reviewers and the Editors for their useful suggestions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Dell'Amico, R. Montemanni, and S. Novellani. Matheuristic algorithms for the parallel drone scheduling traveling salesman problem. *Annals of Operations Research*, 289:211–226, 2020.

Journal Pre-proof

[2] M. Dell'Amico, R. Montemanni, and S. Novellani. Algorithms based on branch and bound for the flying sidekick traveling salesman problem. *Omega*, 104:102493, 2021.

[3] Q. T. Dinh, D. D. Do, and M. H. Há. Ants can solve the parallel drone scheduling traveling salesman problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 14–21, 2021.

[4] Forbes. Drone explosion: $5B investment in 2 years, 129 startups, 170 new craft. `https://www.forbes.com/sites/johnkoetsier/2022/02/07/drone-innovation-check-up-5b-investment-129-companies-170-craft/`, 2022. [Accessed: 2023-02-05].

[5] D. Lei and X. Chen. An improved variable neighborhood search for parallel drone scheduling traveling salesman problem. *Applied Soft Computing*, 127:109416, 2022.

[6] R. G. Mbiadou Saleu, L. Deroussi, D. Feillet, N. Grangeon, and A. Quilliot. An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks*, 72(4):459–474, 2018.

[7] R. G. Mbiadou Saleu, L. Deroussi, D. Feillet, N. Grangeon, and A. Quilliot. The parallel drone scheduling problem with multiple drones and vehicles. *European Journal of Operational Research*, 300:571–589, 2022.

[8] R. Montemanni and M. Dell'Amico. Solving the parallel drone scheduling traveling salesman problem via constraint programming. *Algorithms*, 16(1):40, 2023.

[9] R. Montemanni and L.M. Gambardella. An ant colony system for team orienteering problems with time windows. *Foundations of Computing and Decision Sciences*, 34(4):287–306, 2009.

[10] C. C. Murray and A. G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109, 2015.

[11] M. A. Nguyen, G. T.-H. Dang, M. H. Há, and M.-T. Pham. The min-cost parallel drone scheduling vehicle routing problem. *European Journal of Operational Research*, 299:910–930, 2022.

[12] M. A. Nguyen, H. L. Luong, M. H. Hà, and H. B. Ban. An efficient branch-and-cut algorithm for the parallel drone scheduling traveling salesman problem. *4OR*, 2022.

[13] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018.

Journal Pre-proof

[14] J. Pasha, Z. Elmi, S. Purkayastha, A. M. Fathollahi-Fard, Y.-E. Ge, Y.-Y. Lau, and M. A. Dulebenets. The drone scheduling problem: A systematic state-of-the-art review. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14224–14247, 2022.

[15] L. Perron and V. Furnon. Google OR-Tools, 2023. https://developers.google.com/optimization/ [Accessed: 2023-03-03].

[16] R. Raj, D. Lee, S. Lee, J. Walteros, and C. Murray. *A Branch-and-Price Approach for the Parallel Drone Scheduling Vehicle Routing Problem*. SSRN Electronic Journal, 2021.

[17] G. Reinelt. TSPLIB – A Traveling Salesman Problem Library. *ORSA Journal on Computing*, 3(4):376–384, 1991.

[18] Statista. E-commerce. `https://www.statista.com/markets/413/e-commerce/`, 2022. [Accessed: 2023-02-05].

[19] R. Wolleswinkel, V. Lukic, W. Jap, R. Chan, J. Govers, and S. Banerjee. *An onslaught of new rivals in parcel and express*, volume Travel, Transport and Logistics. Boston Consulting Group, 2018.

Journal Pre-proof

Journal Pre-proof

Constraint Programming models for the parallel drone scheduling vehicle routing problem
R Montemanni, M. Dell'Amico

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Journal Pre-proof