Combining Heuristics and Constraint Programming for the Parallel Drone Scheduling Vehicle Routing Problem with Collective Drones

Roberto Montemanni(✉), Mauro Dell'Amico, and Andrea Corsini

University of Modena and Reggio Emilia, 42122 Reggio Emilia, Italy
{roberto.montemanni,mauro.dellamico,andrea.corsini}@unimore.it

Abstract. Last-mile delivery problems where trucks and drones collaborate to deliver goods to final customers are considered. We focus on settings where a fleet with several homogeneous trucks work in parallel to collaborative drones, able to combine with each other to optimize speed and power consumption for deliveries. A heuristic for the min-max vehicle routing problem is coupled with constraint programming models, leading to an effective method able to provide several state-of-the-art solutions for the instances commonly adopted in the literature.

Keywords: Vehicle Routing · Constraint programming · Heuristics

1 Introduction

The first optimization problems involving distribution with trucks and drones were introduced in Murray and Chu [5], where the concept of a new distribution problem where a truck and a drone make deliveries in a collaborative way, was introduced. In the Parallel Drone Scheduling Traveling Salesman Problem (PDSTSP) there is a truck making a tour to service some customers. In parallel, a set of drones is also employed, and each drone can leave the depot, serve a customer, return to the depot, and repeating several times for different customers. The objective of the optimization is to minimize the makespan required to service all the customers and having all the vehicles back to the depot. Models, exact and heuristic algorithms for the problem have been discussed, e.g., in [3] and [7]. Recently, Amazon Technologies Inc. filed a patent [8] where a new distribution paradigm, taking advantage of a so-called "Collective Drone" (c-drone), is introduced. Multiple drones can be connected and fly as a single drone, to serve a customer. The c-drone is able to transport larger and heavier goods with respect to the single drone, and can also modulate its speed more flexibly in order to increase its range [4]. In the work [6], the authors optimized a problem where collective drones and a truck are used to distribute goods, and the resulting problem is named the PDSTSP-c, where *c* stands for *collective*. An extension of the problem, where multiple trucks are considered, called PDSVRP-c, was

M. Sevaux et al. (Eds.): MIC 2024, LNCS 14753, pp. 136–142, 2024.
https://doi.org/10.1007/978-3-031-62912-9_13

recently introduced in [4]. An example of a PDSVRP-c instance is provided in Fig. 1. Without considering the use of the drones, the problem reduces to a classic Vehicle Routing Problems (VRP) [10] characterized by a *min-max* objective function calculated over the lengths of the different tours, which translates into completing all the deliveries in the shortest possible time. Both exact and heuristic methods have been presented for this problem, that is normally more difficult than a traditional VRP [1].

We investigate how two Constraint Programming models recently proposed for the PDSVRP-c perform once a solution only using trucks (VRP) is fed to the solver as a hint-solution.

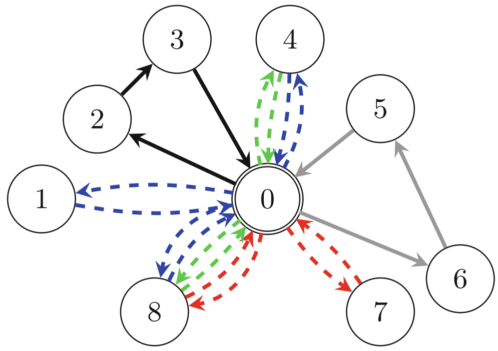

Fig. 1. Example of a PDSVRP-c instance. Node 0 is the depot, while the other nodes are customers. The black and grey continuous arcs represent the tours of the two trucks $(0, 2, 3, 0)$ and $(0, 6, 5, 0)$. The dashed arcs depict instead the missions of the drones, with each colour representing a different one. Notice how for some of the missions multiple drones are collaborating (Color figure online).

2 Problem Description

Given a graph $G(V, E)$ with a set of vertices $V = \{0, 1, \ldots, n\}$, where vertex 0 is the depot and the remaining vertices represent the customers (set $C = V \backslash \{0\}$). A customer i requests a parcel of weight w_i to be delivered to its address from the depot. A set S of s driver-operated delivery trucks, each with unlimited range and capacity, and a set D of m homogeneous drones form the fleet available for deliveries. All the vehicles are based at the depot and the drones have batteries of a given capacity that is installed before each mission. Each truck performs a single delivery tour and no collaboration among trucks is implemented. The deterministic travel time between two vertices $i, j \in V$ is given as t_{ij} for the trucks. The drones instead operate in a back-and-forth fashion from the depot, delivering one parcel at a time. Travel times and maximum ranges of drone missions depend on factors such as the number of drones cooperating and the traveling speed selected. The energy consumption model from [9] is adopted here to calculate battery draining and discharge peaks in order to estimate feasible mission settings. In the configuration considered, characterized by collaborative drones, the weight carried is evenly distributed among the k drones participating in the mission. As described in [6], given a number of drones involved k and a target customer j, the optimal cruise speed that minimizes the time required for the mission itself, while fulfilling the constraints on the batteries (power consumption is used here) can be pre-calculated by inspection. The time τ_j^k required to service customer j with k drones can therefore be pre-calculated as described in [6], with $\tau_i^k = +\infty$ if it is not possible to service customer i with k drones. The set of customers that cannot be serviced by drones is referred to as $\mathcal{C}_\mathcal{T} \subset \mathcal{C}$. Let $\mathcal{C}_\mathcal{F} = \mathcal{C} \backslash \mathcal{C}_\mathcal{T}$ be the set of customers that can be served by drones, and let q_j and p_j be the minimum and maximum number of drones that can be used to serve $j \in \mathcal{C}_{\mathcal{F}}$. The target of the PDSVRP-c is to find truck tours and drones scheduling that minimize the makespan, i.e. to complete all the deliveries in the shortest possible time, given the resources available.

3 Constraint Programming Models

We present two Constraint Programming models introduced in [4], based on the Google OR-Tools CP-SAT solver [2] and representing the state-of-the-art.

Model M2: The model revolves on the following variables: x_{ij} is 1 if edge (i, j), with $i, j \in V$, is traveled by a truck, 0 otherwise. If $x_{jj} = 1$ means that customer j is served by drones; z_j^k is 1 if k drones serve customer $j \in \mathcal{C}_{\mathcal{F}}$, 0 otherwise; y_{ij} is 1 if vertex a drone serves vertex i right before vertex j by one drone, 0 otherwise; $f_{ij} \in Z^+$ indicates the number of drones serving vertex i right before vertex j; $\overline{T}_j \in R^+$ is the time at which the mission at customer $j \in \mathcal{C}_{\mathcal{F}}$ is completed if the visit is operated by drones; it is the time the truck reaches the customer and the service is started in case the visit is operated by a truck.; α is the completion time of all missions. The model is the following one:

$$(M2)\ \min \alpha \tag{1}$$

$$s.t.\ \alpha \geq \overline{T}_j + t_{j0}x_{j0}, j \in \mathcal{C} \tag{2}$$

$$x_{jj} = \sum_{q_j \leq k \leq p_j} z_j^k, j \in \mathcal{C}_{\mathcal{F}} \tag{3}$$

$$\text{MultipleCircuit}\left(x_{ij} : \begin{matrix} i, j \in V, \\ i \neq 0 \vee j \neq 0, \\ i \in \mathcal{C}_{\mathcal{T}} \Rightarrow j \neq i \end{matrix} \right) \tag{4}$$

$$\sum_{j \in \mathcal{C}} x_{0j} \leq s \tag{5}$$

$$x_{ij} \Rightarrow \overline{T}_j \geq T_i + t_{ij}, i \in V, j \in \mathcal{C}, i \neq j \tag{6}$$

$$\sum_{j \in \mathcal{C}_{\mathcal{F}}} f_{0j} \leq m \tag{7}$$

$$\sum_{i \in \mathcal{C}_{\mathcal{F}} \cup \{0\}, i \neq j} f_{ij} = \sum_{q_j \leq k \leq p_j} k z_j^k, j \in \mathcal{C}_{\mathcal{F}} \tag{8}$$

$$\sum_{i \in \mathcal{C}_{\mathcal{F}} \cup \{0\}, i \neq j} f_{ij} = \sum_{l \in \mathcal{C}_{\mathcal{F}} \cup \{0\}, l \neq j} f_{jl}, j \in \mathcal{C}_{\mathcal{F}} \cup \{0\} \tag{9}$$

$$f_{ij} \leq m y_{ij}, i, j \in \mathcal{C}_{\mathcal{F}} \cup \{0\}, i \neq j \tag{10}$$

$$y_{ij} \Rightarrow \overline{T}_j \geq \overline{T}_i + \sum_{q_j \leq k \leq p_j} \tau_j^k z_j^k, \begin{matrix} i \in \mathcal{C}_{\mathcal{F}} \cup \{0\}, \\ j \in \mathcal{C}_{\mathcal{F}}, i \neq j \end{matrix} \tag{11}$$

$$0 \leq f_{ij} \leq m, i, j \in \mathcal{C}_{\mathcal{F}} \cup \{0\}, i \neq j \tag{12}$$

$$x_{ij} \in \{0; 1\}, i, j \in V \tag{13}$$

$$z_j^k \in \{0; 1\}, j \in \mathcal{C}_{\mathcal{F}}, q_j \leq k \leq p_j \tag{14}$$

$$y_{ij} \in \{0; 1\}, i, j \in \mathcal{C}_{\mathcal{F}} \cup \{0\}, i \neq j \tag{15}$$

$$\overline{T}_j \geq 0, j \in V \tag{16}$$

$$m\alpha \geq \sum_{j \in \mathcal{C}_{\mathcal{F}}} \sum_{q_j \leq k \leq p_j} k \tau_j^k z_j^k \tag{17}$$

Constraints (2) state that the total time α to be minimized according to (1), has to be as large as to the time required by the truck and drone missions. Constraints (3) relate x and z variables for each drone-eligible customers; Constraint (4) uses *MultipleCircuit* command of CP-SAT to impose truck tours, while constraint (5) set to$s$ the maximum number of truck tours. Constraints (6) align timing ($\overline{T}$ variables) to tours. Constraints (7)-(9) regulate synchronization among drones (see [4] for details). Constraints (10) and (11) connect y variables with f and z variables, respectively. The remaining constraints (12)-(16) define the domain of the variables. The inequality (17) is not necessary for the validity of the model, but it contributes significantly to tighten the lower bounds, so it is added. The interested reader can refer to [4] for an explanation of the inequality and a formal proof of its validity.

Model M3: The variables of the model are defined starting from those of model M2. Here the x variables are changed to a set of variables w such that $w_{ij}^k = 1$ if edge (i, j) is traveled by truck $k \in S$, 0 otherwise. Moreover, $w_{00}^k = 1$ implies that truck k is not used in the current solution. Variables $\overline{T}$ are substituted by the following variables: T_j represents the completion of the drone-mission to customer $j \in \mathcal{C}_\mathcal{F}$. The resulting model is as follows:

$$(M3) \ \min \alpha \tag{18}$$

$$s.t. \ \alpha \geq \sum_{i \in V} \sum_{j \in V, i \neq j} t_{ij} w_{ij}^k, k \in S \tag{19}$$

$$\alpha \geq T_j, j \in \mathcal{C}_\mathcal{F} \tag{20}$$

$$\sum_{k=1}^{s} (1 - w_{jj}^k) + \sum_{q_j \leq k \leq p_j} z_j^k = 1, j \in \mathcal{C}_\mathcal{F} \tag{21}$$

$$\sum_{k=1}^{s} w_{jj}^k = s - 1, j \in \mathcal{C}_\mathcal{T} \tag{22}$$

$$\text{Circuit}(w_{ij}^k : i, j \in V), k \in S \tag{23}$$

$$w_{ij}^k \leq 1 - w_{00}^k, k \in S, i, j \in \mathcal{C} \tag{24}$$

$$\sum_{j \in \mathcal{C}_\mathcal{F}} f_{0j} \leq m \tag{25}$$

$$\sum_{i \in \mathcal{C}_\mathcal{F} \cup \{0\}, i \neq j} f_{ij} = \sum_{q_j \leq k \leq p_j} k z_j^k, j \in \mathcal{C}_\mathcal{F} \tag{26}$$

$$\sum_{i \in \mathcal{C}_\mathcal{F} \cup \{0\}, i \neq j} f_{ij} = \sum_{l \in \mathcal{C}_\mathcal{F} \cup \{0\}, l \neq j} f_{jl}, j \in \mathcal{C}_\mathcal{F} \cup \{0\} \tag{27}$$

$$f_{ij} \leq m y_{ij}, i, j \in \mathcal{C}_\mathcal{F} \cup \{0\}, i \neq j \tag{28}$$

$$y_{ij} \Rightarrow T_j \geq \overline{T}_i + \sum_{q_j \leq k \leq p_j} \tau_j^k z_j^k, \quad i \in \mathcal{C}_\mathcal{F} \cup \{0\}, j \in \mathcal{C}_\mathcal{F}, i \neq j \tag{29}$$

$$0 \leq f_{ij} \leq m, i, j \in \mathcal{C}_\mathcal{F} \cup \{0\}, i \neq j \tag{30}$$

$$w_{ij}^k \in \{0; 1\}, k \in S, i, j \in V \tag{31}$$

$$z_j^k \in \{0; 1\}, j \in \mathcal{C}_\mathcal{F}, q_j \leq k \leq p_j \tag{32}$$

$$y_{ij} \in \{0; 1\}, i, j \in \mathcal{C}_\mathcal{F} \cup \{0\}, i \neq j \tag{33}$$

$$T_j \geq 0, j \in \mathcal{C}_\mathcal{F} \cup \{0\} \tag{34}$$

$$m\alpha \geq \sum_{j \in \mathcal{C}_\mathcal{F}} \sum_{q_j \leq k \leq p_j} k \tau_j^k z_j^k \tag{35}$$

The constraints follow the meaning already described for the model $M2$ in Sect. 3. The changes are as follows. Equalities (21) are used to force any each drone-eligible customer has to be services by drones or by a truck. The new constraints (22) is required to force customers in $\mathcal{C}_\mathcal{T}$ to be service by a truck. Constraints (23), adopting the *Circuit* command from CP-SAT, are defined for each truck k, since the concept of giant-tour seen in the model $CP2$ does not exist here. The constraints (24) state that a truck k can be used only if $w_{00}^k = 1$.

Hint-start: One of the features of CP-SAT is the possibility of passing a (partial) solution to the solver through some values for the variables of the model. The solver takes these settings as suggestions (hints) and potentially improves its

Table 1. Experimental results.

	2 Trucks						3 Trucks					
	VRP	Model M2		Model M3		New bounds	VRP	Model M2		Model M3		New bounds
		[4]	Hint-start	[4]	Hint-start			[4]	Hint-start	[4]	Hint-start	
Instances	UB	[LB, UB]	[LB, UB]	[LB, UB]	[LB, UB]	[LB, UB]	UB	[LB, UB]	[LB, UB]	[LB, UB]	[LB, UB]	[LB, UB]
50-r-e	128	[65, 116]	[61, 116]	[63, 120]	[**69**, **112**]	[**69**, **112**]	112	[48, 112]	[**62**, **104**]	[47, 112]	[52, 108]	[**62**, **104**]
53-r-e	128	[77, 112]	[65, 116]	[82, 128]	[78, 112]	[82, 112]	112	[56, 96]	[**64**, 96]	[51, 112]	[56, 104]	[**64**, 96]
66-rc-e	128	[72, 112]	[47, 124]	[73, 136]	[63, 116]	[73, 112]	112	[53, 108]	[47, **100**]	[38, 116]	[**57**, 104]	[**57**, **100**]
67-c-c	56	[38, 52]	[20, **48**]	[31, 52]	[19, 52]	[38, **48**]	56	[27, 52]	[21, 52]	[9, 52]	[12, 52]	[27, 52]
68-rc-c	120	[50, 56]	[36, 104]	[52, 104]	[47, 84]	[52, 56]	84	[39, 56]	[36, 60]	[34, 104]	[**42**, 76]	[**42**, 56]
76-c-c	40	[26, 36]	[16, 36]	[16, 40]	[16, 36]	[26, 36]	28	[18, 24]	[12, 24]	[12, 52]	[16, 24]	[18, 24]
82-c-e	64	[32, 64]	[26, 64]	[17, 64]	[12, 64]	[32, 64]	64	[22, 64]	[**26**, 64]	[8, 64]	[10, 64]	[**26**, 64]
82-rc-c	108	[62, 116]	[32, **100**]	[56, 132]	[54, **100**]	[**62**, **100**]	88	[47, 80]	[34, 84]	[38, 128]	[**48**, 84]	[**48**, 80]
88-c-e	84	[54, 84]	[18, 84]	[58, 112]	[40, 84]	[58, 84]	80	[36, 76]	[18, 76]	[32, 104]	[**39**, 80]	[**39**, 76]
91-r-c	152	[75, 152]	[33, 140]	[75, 160]	[63, **124**]	[75, **124**]	108	[56, 120]	[34, 104]	[42, 148]	[54, **100**]	[56, **100**]
99-rc-c	152	[63, 96]	[26, 136]	[51, 144]	[47, 136]	[63, 96]	96	[47, 64]	[26, 92]	[29, 128]	[41, 88]	[47, 64]
101-r-c	176	[71, 164]	[20, 160]	[53, 152]	[55, 152]	[71, 152]	104	[52, 128]	[20, **96**]	[36, 144]	[44, 100]	[52, **96**]
103-rc-c	128	[69, 124]	[34, 124]	[52, 128]	[49, 128]	[69, 124]	88	[49, 96]	[35, **88**]	[32, 136]	[41, **88**]	[49, **88**]
105-rc-e	140	[65, 136]	[27, **128**]	[57, 148]	[52, **128**]	[65, **128**]	112	[49, 120]	[27, **104**]	[34, 132]	[40, 108]	[49, **104**]
108-rc-e	160	[79, 172]	[22, **152**]	[70, 160]	[57, 160]	[79, **152**]	128	[58, 184]	[23, **124**]	[37, 160]	[54, **124**]	[58, **124**]
114-rc-c	120	[58, 124]	[27, **104**]	[49, 140]	[47, **104**]	[58, **104**]	88	[44, 80]	[26, **72**]	[32, 112]	[40, 80]	[44, **72**]
121-rc-e	144	[70, 156]	[25, 140]	[56, 152]	[57, **132**]	[70, **132**]	96	[52, 124]	[28, **96**]	[40, 152]	[44, **96**]	[52, **96**]
126-rc-e	160	[87, 220]	[18, **152**]	[67, 184]	[63, **152**]	[87, **152**]	116	[63, 136]	[20, 108]	[44, 164]	[48, **100**]	[63, **100**]
126-r-c	172	[78, 160]	[26, **136**]	[56, 156]	[58, **136**]	[78, **136**]	108	[56, 140]	[24, **116**]	[38, 148]	[48, **116**]	[56, **116**]
144-rc-c	132	[67, 272]	[21, 128]	[47, 168]	[43, **116**]	[67, **116**]	120	[50, 132]	[23, **116**]	[35, 160]	[41, 120]	[50, **116**]
154-c-c	40	[35, -]	[14, **40**]	[8, 72]	[12, **40**]	[35, **40**]	36	[24, 36]	[14, 36]	[8, 68]	[8, 36]	[24, 36]
165-r-c	200	[88, -]	[16, 192]	[67, 224]	[68, **184**]	[88, **184**]	140	[68, -]	[15, 136]	[50, 212]	[50, **132**]	[68, **132**]
167-r-e	196	[100, -]	[16, 188]	[74, 256]	[75, **180**]	[100, **180**]	140	[73, -]	[16, **132**]	[54, 204]	[56, **132**]	[73, **132**]
173-r-c	196	[85, 204]	[16, 188]	[59, 240]	[60, **176**]	[85, **176**]	136	[65, -]	[16, 132]	[45, 212]	[47, **128**]	[65, **128**]
173-rc-c	152	[79, -]	[21, 148]	[48, 180]	[50, **144**]	[79, **144**]	120	[58, 172]	[20, **116**]	[37, 168]	[40, 120]	[58, **116**]
181-r-e	192	[112, -]	[18, 188]	[78, 252]	[79, **180**]	[112, **180**]	152	[82, -]	[18, **152**]	[55, 216]	[62, **152**]	[82, **152**]
185-c-c	60	[48, -]	[20, **60**]	[24, 96]	[24, **60**]	[48, **60**]	48	[32, -]	[22, **48**]	[14, 96]	[24, **48**]	[32, **48**]
187-rc-e	176	[100, 308]	[27, 172]	[65, 212]	[67, **160**]	[100, **160**]	132	[74, -]	[26, **124**]	[46, 212]	[50, 128]	[74, **124**]
198-c-c	36	[32, -]	[16, **36**]	[12, 64]	[12, **36**]	[32, **36**]	36	[22, 36]	[16, 36]	[8, 68]	[13, 36]	[22, 36]
200-r-e	224	[105, -]	[16, **216**]	[68, 324]	[69, 220]	[105, **216**]	152	[77, -]	[16, 152]	[48, 252]	[50, **148**]	[77, **148**]
Average	131.5	[68.1, 138.0]	[26.0, 124.0]	[52.8, 150.0]	[50.2, 120.3]	[68.6, 117.2]	99.7	[49.9, 97.2]	[26.2, 94.7]	[34.4, 137.9]	[40.9, 95.9]	[51.1, 92.7]

	4 Trucks						5 Trucks					
	VRP	Model M2		Model M3		New	VRP	Model M2		Model M3		New
		[4]	Hint-start	[4]	Hint-start	bounds		[4]	Hint-start	[4]	Hint-start	bounds
Instances	UB	[LB, UB]	[LB, UB]	[LB, UB]	[LB, UB]	[LB, UB]	UB	[LB, UB]	[LB, UB]	[LB, UB]	[LB, UB]	[LB, UB]
50-r-e	116	[46, 104]	[**62**, **100**]	[35, 112]	[37, **100**]	[**62**, **100**]	116	[47, 100]	[**61**, 100]	[30, 112]	[34, 112]	[**61**, 100]
53-r-e	112	[50, 96]	[**64**, 96]	[38, 112]	[39, 100]	[**64**, 96]	112	[50, 92]	[**64**, 92]	[32, 112]	[36, 112]	[**64**, 92]
66-rc-e	108	[41, 104]	[**44**, **100**]	[34, 108]	[35, 104]	[**44**, **100**]	100	[35, 100]	[**46**, 100]	[24, 120]	[32, 100]	[**46**, 100]
67-c-c	56	[21, 48]	[20, 52]	[8, 52]	[11, 52]	[21, 48]	56	[18, 52]	[**21**, 52]	[8, 52]	[11, 52]	[**21**, 52]
68-rc-c	64	[32, 52]	[**36**, 60]	[29, 88]	[30, 60]	[**36**, 52]	60	[28, 44]	[**35**, 56]	[23, 80]	[27, 56]	[**35**, 44]
76-c-c	28	[14, 24]	[12, 24]	[12, 56]	[**16**, 24]	[**16**, 24]	28	[12, 24]	[12, 24]	[12, 40]	[**14**, 24]	[**14**, 24]
82-c-e	64	[18, 64]	[**26**, 64]	[8, 64]	[10, 64]	[**26**, 64]	64	[15, 64]	[**26**, 64]	[6, 64]	[9, 64]	[**26**, 64]
82-rc-c	72	[38, 68]	[32, 68]	[31, 124]	[32, **60**]	[38, **60**]	64	[32, 68]	[**33**, **60**]	[24, 112]	[28, **60**]	[**33**, **60**]
88-c-e	76	[28, 76]	[18, **72**]	[32, 108]	[**39**, 76]	[**39**, **72**]	76	[23, 72]	[18, 72]	[32, 108]	[**38**, 76]	[**38**, 72]
91-r-c	92	[45, 96]	[33, 88]	[32, 156]	[35, **76**]	[45, **76**]	72	[38, 88]	[32, **68**]	[28, 124]	[32, **68**]	[38, **68**]
99-rc-c	76	[37, 68]	[26, 72]	[24, 120]	[26, **64**]	[37, **64**]	60	[32, 64]	[27, **60**]	[20, 108]	[26, 64]	[32, **60**]
101-r-c	84	[42, 76]	[21, 80]	[30, 144]	[31, 80]	[42, 76]	72	[36, 112]	[22, **68**]	[26, 144]	[28, **68**]	[36, **68**]
103-rc-c	76	[39, 80]	[35, **76**]	[26, 140]	[30, **76**]	[39, **76**]	72	[32, 80]	[37, **68**]	[22, 120]	[25, **68**]	[37, **68**]
105-rc-e	116	[39, 116]	[27, **104**]	[26, 132]	[26, 108]	[39, **104**]	104	[33, 112]	[26, **104**]	[21, 124]	[24, **104**]	[33, **104**]
108-rc-e	120	[46, 124]	[22, **120**]	[28, 152]	[36, **120**]	[46, **120**]	128	[39, 120]	[23, 120]	[24, 136]	[31, 124]	[39, 120]
114-rc-c	72	[35, 88]	[28, **72**]	[26, 120]	[26, **72**]	[35, **72**]	80	[30, 64]	[28, 64]	[22, 96]	[22, 72]	[30, 64]
121-rc-e	96	[42, 104]	[34, **92**]	[29, 144]	[31, 96]	[42, **92**]	96	[34, 116]	[30, **92**]	[24, 128]	[24, 96]	[34, **92**]
126-rc-e	112	[50, 132]	[19, **84**]	[35, 164]	[34, **84**]	[50, **84**]	124	[41, 120]	[17, **76**]	[29, 148]	[30, **76**]	[41, **76**]
126-r-c	88	[45, 116]	[23, **112**]	[28, 140]	[30, **112**]	[45, **112**]	80	[37, 116]	[26, **112**]	[24, 144]	[24, **112**]	[37, **112**]
144-rc-c	80	[40, 128]	[27, 76]	[25, 144]	[26, **72**]	[40, **72**]	172	[34, 104]	[29, **92**]	[22, 136]	[23, 148]	[34, **92**]
154-c-c	36	[18, 40]	[14, **36**]	[8, 72]	[8, **36**]	[18, **36**]	36	[15, 36]	[14, 36]	[6, 68]	[12, 36]	[15, 36]
165-r-c	108	[54, 192]	[14, **108**]	[40, 192]	[40, **108**]	[54, **108**]	88	[47, 220]	[14, 88]	[34, 212]	[34, **84**]	[47, **84**]
167-r-e	124	[58, 176]	[16, **120**]	[42, 196]	[43, 124]	[58, **120**]	120	[49, 204]	[16, **116**]	[34, 204]	[35, 120]	[49, **116**]
173-r-c	104	[54, 352]	[16, **96**]	[36, 192]	[37, 104]	[54, **96**]	100	[43, -]	[16, **92**]	[32, 196]	[31, 96]	[43, **92**]
173-rc-c	88	[46, 116]	[21, 88]	[29, 164]	[29, **84**]	[46, **84**]	88	[39, 116]	[21, **80**]	[24, 164]	[24, 88]	[39, **80**]
181-r-e	128	[65, 268]	[19, **124**]	[42, 208]	[42, 128]	[65, **124**]	132	[54, 204]	[20, **120**]	[35, 204]	[36, 132]	[54, **120**]
185-c-c	48	[24, 48]	[20, **44**]	[14, 100]	[24, 48]	[24, **44**]	48	[20, 48]	[20, **44**]	[12, 60]	[**24**, 48]	[**24**, **44**]
187-rc-e	248	[58, 216]	[27, **148**]	[37, 204]	[37, 180]	[58, **148**]	248	[48, 128]	[28, 136]	[32, 192]	[32, 176]	[48, 128]
198-c-c	36	[16, -]	[16, **36**]	[8, 68]	[13, **36**]	[16, **36**]	36	[16, 36]	[16, 36]	[8, 68]	[13, 36]	[16, 36]
200-r-e	124	[60, 308]	[16, **120**]	[38, 228]	[39, 124]	[60, **120**]	120	[52, 288]	[17, **120**]	[32, 216]	[34, **120**]	[52, **120**]
Average	91.7	[40.0, 120.0]	[26.3, 84.4]	[27.7, 133.5]	[29.7, 85.7]	[42.0, 82.7]	91.7	[34.3, 103.2]	[26.5, 80.4]	[23.4, 126.4]	[26.4, 86.4]	[37.2, 79.6]

performance if the information received is valuable. In the experiments reported in [4], it emerges that both the model have scalability issues on large instances, likely due to the difficulties of the *Circuit* and *Multicircuit* commands of CP-SAT of dealing with VRP problems with more than a few tens of customers. In this paper, we evaluate whether hinting a solution can make the models more effective.

In the solution considered, we will ignore the drones and solve each instance as a min-max VRP problem. The solution, using only trucks is then passed to the solver, that might benefit from this because solutions using drones can in principle be obtained by taking away some customers from the tours of the truck.

4 Experimental Results

All the models presented in previous sections have been coded in Python 3.11.2. The Constraint Programming models discussed in Sect. 3 have been solved via the CP-SAT solver of Google OR-Tools 9.6 [2], while the heuristic method adopted for retrieving min-max VRP solutions was the *Route* solver, again from OR-Tools. All the experiment reported have been carried out on a computer equipped with A CPU Intel Core i7 12700F, and 32 GB of RAM and with a maximum computation time of 1 h. The instances originally introduced in [6] for the PDSTSP-c, and available at http://orlab.com.vn/home/download are considered. The number n of customers varies from 50 to 200, the number m of drones available is between 5 and 10 and the number s of trucks is between 2 and 5. The interested reader can find all the details of the instances in [6].

The models M2 and M3, without hint-start (from [4], state-of-the-art at the time of writing) and with hint-start, are considered in Table 1. The upper and lower bounds (when available) found in the given time by each method are reported.

The experiments suggest that hint-starting the solver with a solution is beneficial when considering both lower bounds and (especially) heuristic solutions. Given that the hinted solution is only based on trucks, this was not obvious. Passing an initial solution optimized externally – even without drones – shapes up the truck tours. The CP-SAT solver appears to benefit from such information and seems more effective in taking customers out of the truck tours to assign them to drones, then to design tours from scratch. Finally, a consideration about the use of (collaborative) drones is that they allow an average time-saving in the order of 10% (comparison against the column VRP).

References

1. Bertazzi, L., Golden, B., Wang, X.: Min-max vs. min-sum vehicle routing: a worst-case analysis. Eur. J. Oper. Res. **240**(2), 372–381 (2015)
2. Google: OR-Tools (2023). https://developers.google.com/optimization/. Accessed 03 Mar 2023
3. Montemanni, R., Dell'Amico, M.: Solving the parallel drone scheduling traveling salesman problem via constraint programming. Algorithms **16**(1), 40 (2023)

4. Montemanni, R., Dell'Amico, M., Corsini, A.: Parallel drone scheduling vehicle routing problems with collective drones. Comput. Oper. Res. **163**, 106514 (2024)
5. Murray, C.C., Chu, A.G.: The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. Transp. Res. Part C: Emerg. Technol. **54**, 86–109 (2015)
6. Nguyen, M.A., Hà, M.H.: The parallel drone scheduling traveling salesman problem with collective drones. Transp. Sci. **4**(57), 866–888 (2023)
7. Nguyen, M.A., Luong, H.L., Hà, M.H., Ban, H.B.: An efficient branch-and-cut algorithm for the parallel drone scheduling traveling salesman problem. 4OR **21**, 609–637 (2023)
8. Paczan, N.M., Elzinga, M.J., Hsieh, R., Nguyen, L.K.: Collective unmanned aerial vehicle configurations (2022). Patent US 11,480,958 B2
9. Raj, R., Lee, D., Lee, S., Walteros, J., Murray, C.: A branch-and-price approach for the parallel drone scheduling vehicle routing problem. SSRN Electron. J. 1–47 (2021)
10. Toth, P., Vigo, D.: The Vehicle Routing Problem. SIAM (2002)