Drone Routing with Energy Function: Formulation and Exact Algorithm

Chun Cheng[a], Yossiri Adulyasak[b], Louis-Martin Rousseau[a]

[a] *Polytechnique Montréal and CIRRELT, Montréal, H3C 3A7, Canada*
[b] *HEC Montréal and GERAD, Montréal, H3T 2A7, Canada*

Abstract

Drone delivery is known as a potential contributor in improving efficiency and alleviating last-mile delivery problems. For this reason, drone routing and scheduling has become a highly active area of research in recent years. Unlike the vehicle routing problem, however, designing drones' routes is challenging due to multiple operational characteristics including multi-trip operations, recharge planning, and energy consumption calculation. To fill some important gaps in the literature, this paper solves a multi-trip drone routing problem, where drones' energy consumption is modeled as a nonlinear function of payload and travel distance. We propose adding logical cuts and subgradient cuts in the solution process to tackle the more complex nonlinear (convex) energy function, instead of using the linear approximation method as in the literature, which can fail to detect infeasible routes due to excess energy consumption. We use a 2-index formulation to model the problem and develop a branch-and-cut algorithm for the formulation. Benchmark instances are first generated for this problem. Numerical tests indicate that even though the original model is nonlinear, the proposed approach can solve large problems to optimality. In addition, in multiple instances, the linear approximation model yields routes that under the nonlinear energy model would be energy infeasible. Use of a linear approximation for drone energy leads to differences in energy consumption of about 9% on average compared to the nonlinear energy model.

Keywords: drone routing, nonlinear energy function, logical cut, subgradient cut, branch-and-cut

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) or drones have attracted people's attention, especially since 2013 when Amazon announced their Prime Air UAV (Rose 2013). Other companies, like DHL, Google, and Alibaba also began developing their own drones, because they believe drones have the potential to reduce cost and waiting time for last-mile delivery. The development of technology has made this idea possible. For example, carbon fiber manufacturing costs have decreased dramatically during the past few years, which enable stronger and lighter air frames (Morgan 2005); lithium polymer batteries with high energy density are also now available, which help extend drones' flight range (Reddy 2010). Different companies have designed different drone models, notably, the multirotor drones used by UPS and DHL, and the hybrid drones developed by Amazon and Alphabet. Being similar to the multirotor helicopters, multirotor drones are lifted and propelled by rotors. Hybrid drones can take off and land vertically (like helicopters), but use wing or wing-like surfaces to generate lift. Meanwhile they can also perform horizontal maneuvers like airplanes. On October 18, 2019, Alphabet's drone unit Wing launched the first commercial drone delivery flight in the United States (Doherty 2019).

Compared to trucks, drones have some specific advantages: (i) They can save labor, because no drivers (or pilots) are needed. (ii) They can often travel faster than trucks. (iii) They are not restricted to road networks (Agatz et al. 2018). These merits enable logistics companies and on-line stores to use drones for rapid parcel delivery. Humanitarian organizations are also considering using drones in disaster scenarios. For example, in the immediate aftermath of a disaster, drones can provide support with risk assessment, mapping, and temporary communication network creation (Chowdhury et al. 2017). In situations where the transportation network is severely compromised by natural disasters, drones can deliver emergency supplies to affected regions. In addition, by taking traffic off the roads, drone might reduce negative implications on congestion, safety, and the environment (Heutger and Kückelhaus 2014).

On the other hand, some unique characteristics of drones have presented new operational challenges. Limited battery capacity influences a drone's flight duration, which can also be affected by payload, speed, and weather conditions (Dorling et al. 2017). Therefore, how should we represent the relationship between battery energy consumption and various factors which affect it? How to route drones so that they can safely return after visiting designated sites? Furthermore, drones' payload is also limited, which means that a drone can only visit a small number of customers during a trip. Thus, how should we schedule drones to serve more demands to maximize their use?

In this paper, we use the term *drone routing problem* (DRP) to refer to the problem where a fleet of drones visit a set of customer locations and each drone can visit multiple customers in a trip. In this case, drones can only be dispatched once from the depot. When drones can perform multiple trips (each trip starts and ends at the depot), this problem is referred to as the *multi-trip drone routing problem* (MTDRP). Existing research on drone operations normally assumes that drone flight duration is limited by a fixed amount of distance or time. However, flight duration is actually influenced by several factors such as battery energy capacity, battery weight, and payload. In addition, no benchmark instances and efficient exact algorithms are available for the DRP, which poses a limitation on algorithm evaluation. To fill some gaps in this area, this paper solves a MTDRP with time windows, where a fleet of homogeneous multirotor drones are dispatched to deliver packages to customers within stipulated time slots. The main contribution of this paper is to incorporate a nonlinear model of drone energy consumption that depends on payload and travel distance. We use a 2-index formulation to model the problem and develop a branch-and-cut algorithm to solve it. We also generate several benchmark instance sets, which are available to the research community.

The rest of this paper is organized as follows. Section 2 reviews related literature and states the contributions of our work. Section 3 describes our problem, presents the mathematical model, and introduces valid inequalities to strengthen it. Section 4 presents techniques for the calculation of energy consumption and provides details of our exact algorithm. Numerical tests and analyses are presented in Section 5. This is followed by the conclusions in Section 6.

2. Literature Review

This section reviews related literature on the drone delivery problem and the multi-trip vehicle routing problem. A summary of the papers on the drone delivery problem is given in Table 1. For more details about drones' civil applications, see the review paper by Otto et al. (2018).

2.1. Drone Delivery Problem

We divide literature on drone delivery problems reviewed here into two categories: drone-only problems and truck-drone problems. For the former, only drones are used in the delivery system. For the latter, both trucks (one or multiple) and drones are used simultaneously. A truck can be used either as a tool to carry drones (i.e., the truck does not have delivery tasks) or for both delivery tasks and as a temporary hub to launch/retrieve drones. Trucks and drones can also work in parallel making deliveries.

Drone-only problems. Studies on drone-only delivery systems normally assume that there are multiple drones and that each drone can cover one or several customers per trip. Choi and Schonfeld (2017) study an automated drone delivery system, where all customers' demands are the same. They use the relationship among battery capacity, payload, and flight range to optimize the drone fleet size. San et al. (2016) describe the implementation steps used to assign a fleet of heterogeneous UAVs to deliver items to target locations. Each order placed by a customer can include one or multiple items. Because of drones' limited payload, one order may not be completely fulfilled in one trip; thus, multiple deliveries might be required. They use a genetic algorithm to solve the problem, where a multi-dimensional chromosome representation is introduced. Dorling et al. (2017) propose two vehicle routing problem (VRP) variants for drone delivery. The first one minimizes the total operating cost subject to a delivery time limit, and the second one optimizes delivery time subject to a budget constraint. The costs include drone fleet cost and energy cost. Instead of dealing directly with the original form of the power function, which is nonlinear, they use a linear approximation function to calculate the power consumption which varies linearly with payload and battery weight. To save cost, each drone can perform multiple trips and visit multiple customers per trip. They use a simulated annealing (SA) heuristic to solve the models. Troudi et al. (2018) study a drone delivery problem with time windows and a trip duration limit. They minimize three different objectives: travel distance, the number of drones used, and the number of batteries required. When imposing the linear energy constraints, the battery capacity is reserved at 20% to be a buffer for unusual conditions.

Some works study the impacts of drone delivery on costs and carbon dioxide (CO_2) emissions.

D'Andrea (2014) analyze the feasibility of using drones for package delivery in terms of energy requirement and economics. They approximate power consumption as a linear function of payload and velocity. Figliozzi (2017) assess the potential of drones in reducing CO_2 emissions generated by the electricity supply chain and provide a comparison of this system with delivery using diesel vehicles and electric trucks/tricycles. They also consider the emissions from the vehicle production and disposal phases. Stolaroff et al. (2018) use the same battery reservation policy as in Troudi et al. (2018) when studying the energy use and environmental impacts of drones for last-mile delivery in comparison with medium-duty trucks. Their power function for hovering takes a similar form as that in Dorling et al. (2017).

There are also studies focusing on drone energy models, where drones' flying status are considered. Liu et al. (2017) derive a theoretical model to calculate the multirotor drone's power consumption. They identify the model's parameters by performing field tests. In their experiments, they consider different drone statuses in a flight path: ascend/descend, hover, and straight line fight. Kirschstein (2020) compare the energy demands of drone-based and ground-based (diesel trucks and electric trucks) parcel delivery services. Factors like drone weight, speed, head wind speed, and other drone parameters are taken into account for energy calculation. Zhang et al. (2020) review energy consumption models for drone delivery. They identify key factors that affect drone energy consumption and discuss similarities and differences among various models. For cruising flight, drone power consumption can be modeled as a convex function of a drone's total weight (e.g., Liu et al. (2017); Stolaroff et al. (2018); Kirschstein (2020)), while for hovering it is proportional to the weight to the power 1.5 (Dorling et al. 2017).

Truck-drone problems. The truck-drone tandem system is the most intensively studied area in drone delivery problems. Most papers in this area assume that during each trip a drone can visit only one customer. Murray and Chu (2015) consider two types of truck-drone delivery problems. The first is the flying sidekick traveling salesman problem (FSTSP), where one truck carries one drone to deliver parcels to a set of customers. As the driver performs deliveries, the UAV is launched from the truck, delivering a parcel for an individual customer, then the truck and the drone rendezvous at a new customer location. The second problem in Murray and Chu (2015) is the parallel drone scheduling traveling salesman problem (PDSTSP), where multiple drones make single-stop delivery trips from the depot while a single truck serves other customers without carrying any drone. The objective of both problems is to minimize the time required to service all customers and return to the depot. Simple heuristics are used to solve both problems. Ponza (2016) uses a SA heuristic to solve the FSTSP. Agatz et al. (2018) use a route first-cluster second heuristic to solve a variant of the FSTSP where the truck can wait at the start node for the drone to return. Bouman et al. (2018) and Poikonen et al. (2019) use a dynamic programming (DP) approach and a branch-and-bound (B&B) algorithm for the same variant, respectively. Marinelli et al. (2017) extend the FSTSP by allowing the launch and rendezvous operations to be performed not only at a node, but also along a route arc. A greedy randomized adaptive search procedure is developed for the problem. Jeong et al. (2019) extend the FSTSP by considering energy consumption and no-fly zones. The authors use the power consumption linear approximation from Dorling et al. (2017) and propose an evolutionary-based heuristic solution algorithm that integrates constructive and search heuristics. Moshref-Javadi and Lee (2017) use a truck-drone tandem system to minimize latency in a customer-oriented distribution system. They compare the benefits of using drones for a single trip versus multiple trips. Ham (2018) extends the PDSTSP by assuming that drones can perform two types of tasks: drop-off and pickup. A constraint programming method is applied. Ulmer and Thomas (2018) study a same-day delivery problem with trucks and drones, where customer orders come dynamically during a shift. The authors present a Markov decision model and an approximate DP algorithm to solve the problem.

Some studies consider multiple trucks where each is equipped with one or multiple drones. Wang et al. (2017) and Poikonen et al. (2017) consider a fleet of homogeneous trucks with multiple drones per truck. Their objective is to minimize the maximum duration of the routes, and they focus on the worst-case analysis. Pugliese and Guerriero (2017) extend the problem by considering time window constraints. Wang and Sheu (2019) allow docking hubs where trucks can drop off, and drones can pick up, parcels for delivery maintain backup drones. They present an arc-based model and develop a branch-and-price (B&P) algorithm. Raj and Murray (2020) study the multiple FSTSP with variable drone speeds. They assume that drone power consumption is a function of

Table 1: Summary of papers on drone delivery problem

	Problem					
Authors	# truck	# drone	# cust/trip	multi-trip	energy function	Solution method
Choi and Schonfeld (2017)	N/A	multiple	multiple			Mathematical analysis
San et al. (2016)	N/A	multiple	1	√		Genetic algorithm
Dorling et al. (2017)	N/A	multiple	multiple	√	√	Simulated annealing heuristic
Troudi et al. (2018)	N/A	multiple	multiple	√	√	Mixed-integer linear programming
Murray and Chu (2015)	1	1	1	√		TSP route and re-assign heuristic
	1	multiple	1	√		Partition and re-assign heuristic
Ponza (2016)	1	1	1	√		Simulated annealing heuristic
Agatz et al. (2018)	1	1	1	√		Route first-cluster second
Bouman et al. (2018)	1	1	1	√		Dynamic programming
Poikonen et al. (2019)	1	1	1	√		Branch-and-bound
Marinelli et al. (2017)	1	1	1	√		Greedy randomized adaptive search procedure
Jeong et al. (2019)	1	1	1	√	√	Evolutionary-based heuristic
Moshref-Javadi and Lee (2017)	1	multiple	1	√		Mixed-integer linear programming
Ham (2018)	multiple	multiple	multiple	√		Constraint programming, variable ordering heuristics
Ulmer and Thomas (2018)	multiple	multiple	1	√		Approximate dynamic programming
Wang et al. (2017)	multiple	multiple	1	√		Worst-case analysis
Poikonen et al. (2017)	multiple	multiple	1	√		Worst-case analysis
Pugliese and Guerriero (2017)	multiple	multiple	1	√		Mixed-integer linear programming
Wang and Sheu (2019)	multiple	multiple	multiple	√		Branch-and-price
Raj and Murray (2020)	1	multiple	1	√	√	Three-phased iterative heuristic
Mathew et al. (2015)	1	1	1	√		Reduce to TSP, then use TSP solver
Luo et al. (2017)	1	1	multiple	√		TSP route and split; route and re-assign
Carlsson and Song (2017)	1	1	1	√		Continuous approximation model
Campbell et al. (2017)	1	1/multiple	1	√		Continuous approximation model
This paper	N/A	multiple	multiple	√	√	Branch-and-cut

cust/trip: number of customers per drone trip. N/A: trucks are not used in the system.

speed and payload, which affects flight endurance and range.

Sometimes the truck is only used for carrying drones and packages without making any deliveries itself (Mathew et al. 2015; Luo et al. 2017). Carlsson and Song (2017) use continuous approximation techniques to derive the improvement of service quality (i.e., the completion time of all deliveries) by using a truck-drone system. Unlike other studies, they do not restrict the drone launch/retrieval locations to be customer sites. Campbell et al. (2017) also use a continuous approximation approach to derive general insights from the aspect of cost.

In the aforementioned literature, we find that only a few papers explicitly consider energy constraints, and many use an approximation that is linear in the payload. In addition, to the best of our knowledge, no benchmark instance is available for algorithm evaluation, and no efficient exact algorithm has been developed for the DRP.

2.2. Multi-trip Vehicle Routing Problem

The multi-trip vehicle routing problem (MTVRP) extends the classical VRP by allowing each truck to perform multiple trips. Fleischmann (1990) is the first to study this problem. The author develops a modification of the saving algorithm and uses a bin packing heuristic to assign routes to vehicles. Mingozzi et al. (2013) develop two set-partitioning-like formulations for the MTVRP.

Azi (2011) develops a B&P algorithm for the MTVRP with time windows (MTVRPTW). Their numerical tests focus on the type 2 instance sets in Solomon (1987). Macedo et al. (2011) propose a network flow model based on generated trips for the same problem. Hernandez et al. (2014) develop an exact two-phase algorithm. In the first phase, they enumerate all feasible trips; in the second phase, they use a B&P algorithm to select the best set of schedules. Azi et al. (2014) and Wang et al. (2014) develop an adaptive large neighborhood search and a route pool-based metaheuristic for the same problem, respectively. Hernandez et al. (2016) develop two set covering formulations for the MTVRPTW without the trip duration constraint and use B&P algorithms. They compare the two models on instances with the first 25 customers of Solomon's "C2", "R2", and "RC2" instances.

In the review paper by Cattaruzza et al. (2016), they suggest that there are four ways to formulate the MTVRP. The first one is the 4-index formulation, which uses both the vehicle index and the trip index. Specifically, a binary variable x_{ij}^{vr} is defined to denote whether trip r of vehicle v travels through arc (i, j). The second and the third ones are the 3-index formulations with either a trip index, or with a vehicle index, respectively. That is, a variable x_{ij}^{r} (x_{ij}^{v}) is used to denote whether trip r (vehicle v) travels through arc (i, j). And the last one is the 2-index formulation using a variable x_{ij}, i.e., neither a vehicle nor a trip index is used. For the 3-index formulation with a trip index, since the number of trips performed by each vehicle is unknown, one has to set a sufficiently large cardinality for the trip set, resulting in a weak model with a large number of variables. Or, we can impose an upper bound on the maximal number of trips each vehicle can perform. For the 3-index formulation with a vehicle index, symmetries resulting from identical vehicles are introduced to the model, which make the formulation weak. Cattaruzza et al. (2016) indicate that the only compact formulation for the MTVRP is proposed by Karaoğlan (2015), where a 2-index formulation is applied. Rivera et al. (2013) also use a 2-index formulation for a multi-trip cumulative capacitated VRP, where the objective is to minimize the sum of arrival times at required nodes. For our problem, as there is no limit on the number of trips that each drone can perform, we do not consider the formulation with a trip index. Further, our preliminary tests also indicate that the 3-index formulation with a drone index provides worse results than the

2-index formulation. Therefore, in Section 3.2, we present a 2-index formulation for our MTDRP.

2.3. Our Contributions

The contributions of our study are fourfold. First, we explicitly represent drone's energy consumption as a nonlinear function of payload and travel time, instead of assuming that flight range (maximum distance or time) is a fixed number. To tackle the nonlinear energy function, instead of relying on a linear approximation (e.g., as in Dorling et al. (2017)), we propose adding two types of cuts in the solution process. Our results show that using a linear energy approximation can lead to routes that are energy infeasible under the nonlinear energy consumption model. Second, a 2-index formulation scheme is presented, which is solved by a branch-and-cut (B&C) algorithm. To the best of our knowledge, this paper is the first to formulate a MTDRP and use an exact algorithm for drone routing problems. Third, we generate several benchmark instance sets based on the realistic parameters and known instance sets in the literature, which will be available to the research community and allow for a better comparison of algorithms. Fourth, we provide extensive computational results of the formulation and the algorithm.

3. Formulation

This section presents the problem, constructs the mathematical model, and introduces valid inequalities to strengthen the model.

3.1. Problem Definition

The problem is defined on a directed graph $G = (N, A)$, where $N = \{0, \dots, n+1\}$ is the set of nodes. Node 0 represents the starting depot, and node $n+1$ is a copy of node 0 and it represents the returning depot. $N' = \{1, \dots, n\}$ is the set of customers. For notational convenience, we denote $N^+ = \{0, \dots, n\}$ and $N^- = \{1, \dots, n+1\}$. $A = \{(i, j) : i \in \{0\}, j \in N'$ and $i \in N', j \in N^-, i \neq j\}$ is the set of arcs. Sets $\delta^-(i)$ and $\delta^+(i)$ represent node i's predecessor and successor nodes, respectively.

Each customer is associated with a non-negative demand d_i, and a hard time window $[a_i, b_i]$. For the depots, $[a_0, b_0] = [a_{n+1}, b_{n+1}]$, where a_0 and b_0 are the earliest possible departure time and the latest possible arrival time, respectively. A fleet of K homogeneous multirotor drones are based at the depot. Q is the maximum payload of a drone and we assume that $d_i \leq Q, \forall i \in N'$. Each drone can perform several trips and during a trip it can visit several customers. Drone speed is assumed to be a constant number, and with each arc (i, j) is associated a travel time t_{ij} and a travel cost c_{ij}. Further, it is assumed that the triangle inequality is satisfied for t_{ij}. Without loss of generality, here we assume the service time at each customer is 0, because we can set t_{ij} to be the sum of travel time on arc (i, j) and the service time at node i. We consider multirotor drones in the study as these have been often used in drone delivery analyses and we use data from Dorling et al. (2017). Hybrid drones may have different performance characteristics and require a different energy model.

The problem consists in designing a set of drone routes, such that the objective function is optimized and the following constraints are satisfied: (1) Each route starts at depot 0 and ends at depot $n+1$. (2) Every customer is visited exactly once. (3) The sum of duration of trips assigned to the same drone does not exceed b_{n+1}. (4) The drone weight capacity constraint, battery energy constraint, and customers' time windows must be respected.

3.2. Mathematical Model

Decision variables. There are two sets of binary variables: $x_{ij} = 1$ if arc (i, j) is traversed by a drone, 0 otherwise. $z_{ij} = 1$ if a trip finishing with customer i is followed by another trip visiting j as the first customer (performed by the same drone), 0 otherwise. There are four sets of continuous variables: q_{ij} is the product weight carried through arc (i, j) (kg). τ_i is the start of service time at node $i \in N^-$ $(second)$. f_i is the accumulated energy consumption of a drone upon arrival at node i (kWh). e_{ij} is the energy consumption on arc (i, j) (kWh).

Constraints. We organize the constraints into five groups:

(i) *Route feasibility:*

$$\sum_{j \in \delta^+(i)} x_{ij} = 1 \qquad \forall i \in N', \tag{1}$$

$$\sum_{j \in \delta^-(i)} x_{ji} = 1 \qquad \forall i \in N', \tag{2}$$

$$\sum_{j \in \delta^+(0)} x_{0j} = \sum_{j \in \delta^-(n+1)} x_{j,n+1}. \tag{3}$$

Constraints (1) and (2) guarantee that each customer is visited exactly once. Constraints (3) indicate that the number of trips leaving the starting depot is equal to the number arriving at the ending depot.

(ii) *Weight related constraints:*

$$\sum_{i \in \delta^-(j)} q_{ij} - \sum_{i \in \delta^+(j)} q_{ji} = d_j \qquad \forall j \in N', \tag{4}$$

$$q_{ij} \leq Q x_{ij} \qquad \forall (i,j) \in A, \tag{5}$$

$$q_{i,n+1} = 0 \qquad \forall i \in N'. \tag{6}$$

Equations (4) impose that each customer's demand must be satisfied, and also eliminate subtours. Constraints (5) guarantee that drone weight capacity is respected. Equations (6) indicate that drones cannot carry any product from a customer to the ending depot.

(iii) *Drone energy constraints:*

We only consider drones' energy consumption during level flight in this study. Dorling et al. (2017) suggest that the average power during hover is an upper bound on the average power during flight. Since there are not available field tests of small drones making multiple deliveries or of actual delivery drones in *production mode*, in this study, we use the theoretical power consumption during hovering to approximate the horizontal power consumption for a delivery drone making multiple-stop trips. Leishman (2006) describes the energy consumption, $P(q)$, of a single rotor helicopter in hover as a convex function of payload q. Based on the assumption that each rotor shares the total weight of a drone equally, Dorling et al. (2017) derive the power consumption equation for a h-rotor drone as

$$P(q_{ij}) = (W + m + q_{ij})^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varsigma h}}, \tag{7}$$

where W is the frame weight (kg), m is the battery weight (kg), q_{ij} is the payload (kg), g is the force due to gravity (N), ρ is the fluid density of air (kg/m^3), ς is the area of spinning blade disc (m^2), h is the number of rotors, and the unit of P is *Watt*. In the experiments of Liu et al. (2017), the power consumption in hover also takes a similar form, i.e., $P(q_{ij}) = c_p[(W + m + q_{ij})g]^{\frac{3}{2}}$, where

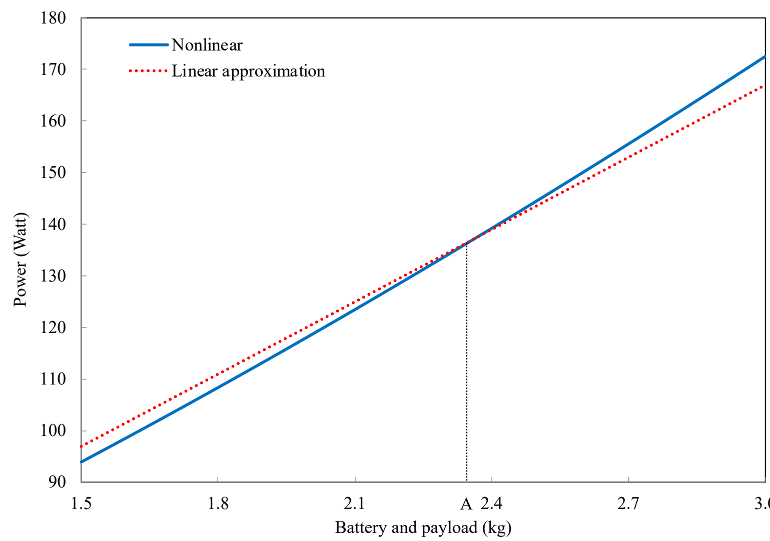

Figure 1: Energy calculation from linear and nonlinear functions (Figure 1 in Dorling et al. (2017))

c_p is a parameter. We rewrite Equation (7) as

$$P(q_{ij}) = k(W + m + q_{ij})^{\frac{3}{2}}, \tag{8}$$

where k depends on the details of the drone and the environmental parameters and it is a constant in our model. Based on field tests, Dorling et al. (2017) propose to approximate power consumption as

$$P(q_{ij}) = \alpha(m + q_{ij}) + \beta, \tag{9}$$

where $\alpha(kW/kg)$ and $\beta(kW)$ are two constant numbers obtained by a linear approximation.

As shown in Figure 1, when the sum of the battery weight and payload is smaller than A, the linear approximation function overestimates the energy consumption from the nonlinear model, and therefore drone routes calculated with the linear approximation will be "energy feasible" if the nonlinear model is used to calculate energy consumption. However, when the battery and payload weight is larger than A, then the linear approximation function underestimates the energy consumption from the nonlinear model. In this case, drone routes calculated with the linear approximation may be "energy infeasible" (i.e., exceed the battery's energy capacity) if the nonlinear model is used to calculate energy consumption. We use Equation (8) to compute power consumption in this study, and drones' energy consumption constraints are written as

$$f_0 = 0, \tag{10}$$

$$f_i + k'(W + m + q_{ij})^{\frac{3}{2}} t_{ij} \le M_{ij}(1 - x_{ij}) + f_j \quad \forall (i,j) \in A, \tag{11}$$

$$f_{n+1} \le \sigma. \tag{12}$$

Equations (10) indicate that at the beginning of each trip the accumulated energy consumption is 0, that is, every time a drone begins a new trip we swap it with a fully charged battery. This assumption is common in the literature (Murray and Chu 2015; Chowdhury et al. 2017; Ham 2018). Equations (11) establish the energy relationship between node i and its immediate successor j, where k' is a constant that includes k from earlier and the conversion from *Watt-second* to kWh and M_{ij} is an arbitrary large constant. We can observe that, when $x_{ij} = 0$, according to Equations (5), q_{ij} also equals 0, then we can set $M_{ij} = k'(W+m)^{\frac{3}{2}} t_{ij} + \sigma$ (σ is the battery energy available for a drone trip (kWh)). When $x_{ij} = 1$, the second term of the left-hand side of Equations (11) is the energy consumption on arc (i, j). Constraints (12) mean that battery's energy capacity constraint must be respected. Since constraints (11) are nonlinear, the model cannot be solved directly by a mixed-integer linear programming (MILP) solver. In Section 4.1, we introduce different types of cuts to tackle this group of constraints implicitly.

We also give the linear approximation version of constraints (11):

$$f_i + [\alpha(m + q_{ij}) + \beta] t_{ij}/3600 \le M'_{ij}(1 - x_{ij}) + f_j \qquad \forall (i,j) \in A, \tag{13}$$

where $M'_{ij} = (\alpha m + \beta) t_{ij}/3600 + \sigma$. In numerical tests, we will compare the difference in solution construction when using these two versions of the energy expressions.

(iv) *Time and trip related constraints:*

$$\tau_i + t_{ij} - M''_{ij}(1 - x_{ij}) \le \tau_j \qquad \forall i \in N', j \in N^-, \tag{14}$$

$$a_i \le \tau_i \le b_i \qquad \forall i \in N^-, \tag{15}$$

$$\tau_i + (t_{i,n+1} + t_{0j}) \le \tau_j + (1 - z_{ij}) M'''_{ij} \qquad \forall i, j \in N', i \ne j, \tag{16}$$

$$\sum_{\substack{i \in N' \\ i \ne j}} z_{ij} \le x_{0j} \qquad \forall j \in N', \tag{17}$$

$$\sum_{\substack{j \in N' \\ j \ne i}} z_{ij} \le x_{i,n+1} \qquad \forall i \in N', \tag{18}$$

$$\sum_{j\in N'} x_{0j} - \sum_{i\in N'} \sum_{\substack{j\in N'\\ j\neq i}} z_{ij} \leq K. \tag{19}$$

Constraints (14) establish the time relationship between customer i and its immediate successor j. We set the large constants $M''_{ij} = \max\{b_i + t_{ij} - a_j, 0\}$ (Desaulniers et al. 2014). Constraints (15) denote that the time window constraint must be respected. Here we impose the time window constraint instead of the deadline constraint, because the latter is a special case of the former with $a_i = 0, \forall i \in N$. This model fits best when drones land at customer sites for delivery, as we assume that drones can wait at customer locations until the opening of the time window and the energy consumption during this period is negligible. Note that in the case where the energy consumption during that period must be taken into account (e.g., in case when drones are equipped with cameras and sensors on to actively detect dangerous situations such as package or drone theft, or for a hovering while waiting), we can also incorporate the energy consumption of performing these activities in our model and our solution scheme can still be directly used. The detailed description on the modifications is presented in Appendix A. Equations (16) establish the time relationship between consecutive trips performed by the same drone, where $M'''_{ij} = t_{i,n+1} + t_{0j} + b_i$. These constraints take into account the time to return to the depot and replace the battery. Constraints (17)–(18) connect variables $\mathbf{x}$ and $\mathbf{z}$ (Karaoğlan 2015). Constraints (19) limit the number of drones that can be used in the system.

(v) *Variable domains:*

$$x_{ij} \in \{0,1\},\ q_{ij}, e_{ij} \geq 0 \qquad \forall (i,j) \in A, \tag{20}$$

$$f_i \geq 0 \qquad \forall i \in N, \tag{21}$$

$$\tau_i \geq 0 \qquad \forall i \in N^-, \tag{22}$$

$$z_{ij} \in \{0,1\} \qquad \forall i,j \in N'. \tag{23}$$

Objective function. We consider the applications of logistics companies who use drones for last-mile delivery, in order to reduce an overall transportation cost. Therefore, we consider a general form of the objective function which also incorporates the energy consumption

$$\min \sum_{(i,j)\in A} (c_{ij}x_{ij} + \delta e_{ij}), \tag{24}$$

where δ is the battery-related cost ($\$/kWh$) which includes the cost of electricity and the amortization of lithium-ion battery. We will show how variables $e_{ij}, \forall(i,j) \in A$ are incorporated in the constraints and linked to variables f_i and f_j in following sections. Note that the energy cost could be negligible in realistic applications, and we add it here for two reasons: First, to keep consistent with some existing works, which also include the energy cost in the objective function to incorporate the depreciation and operating cost of battery as a function of energy usage (Mathew et al. 2015; Dorling et al. 2017); Second, to demonstrate that our objective function is quite flexible. The model and approach can be used to solve a traditional VRP objective which minimizes the travel cost by dropping the second term, or a green supply chain related objective that minimizes the energy consumption/cost by dropping the first term. We analyze the impact of different objectives on computational efficiency and solution configurations in Section 5.3. For notational convenience, in the following sections we use R, E, and $R+E$ to represent the model that minimizes travel cost ($\delta = 0$), energy cost ($c_{ij} = 0, \forall(i,j) \in A$), and both travel and energy costs (as in the objective function (24)), respectively. For the energy calculation, we use a subscript e if the nonlinear energy function is used, and a subscript a if the linear approximation method is used.

We note that constraints in group (i), (ii), (iv), and (v) are adopted from studies on VRP and MTVRP (Desaulniers et al. 2014; Karaoğlan 2015; Cattaruzza et al. 2016). However, the nonlinear energy constraints and the objective function are newly introduced. Moreover, the time-window constraints, which are not considered in Karaoğlan (2015) and Dorling et al. (2017), are also considered in our study. Thus, our model generalizes the other models in the literature, such that it can capture important practical constraints. We further emphasize that our modeling and solution schemes (introduced in next section) simultaneously optimize multi-trip drone routing operations and energy consumption under time windows constraints. We also include a more complex nonlinear energy function.

3.3. Valid Inequalities

We use constraints (25) to indicate the least number of trips needed to visit all the customers (Semet et al. 2014; Santos et al. 2014).

$$\sum_{j\in N'} x_{0j} \geq \left\lceil \frac{\sum_{i\in N'} d_i}{Q} \right\rceil. \tag{25}$$

Constraints (26) are derived from Equations (8) using the constant d_j to replace the variable q_{ij}, which yields linear equations and a lower bound of $P(q_{ij})$ since $q_{ij} \geq d_j$ when $x_{ij} = 1$. Constraints (26) mean that if arc (i, j) is traversed by a drone, the energy consumption is at least equal to the value of the right-hand side.

$$e_{ij} \geq k'(W + m + d_j)^{\frac{3}{2}} t_{ij} x_{ij} \qquad \forall (i, j) \in A. \tag{26}$$

4. Solution Method

In this section, we introduce the techniques to handle the nonlinear energy consumption, and develop a B&C algorithm for our model. We note that our solution method can also be applied to other applications with nonlinear energy functions.

4.1. Cuts for Nonlinear Energy Function

Logical cut (infeasibility cut). We first solve the model without constraints (10)–(12). When a feasible solution is generated, we check whether it satisfies the energy capacity constraint for each trip. For any violated trip $\{0, i_1, \ldots, i_l, n+1\}$, we add the logical cut

$$x_{i_1 i_2} + x_{i_2 i_3} + \ldots + x_{i_{l-1} i_l} \leq l - 2, \tag{27}$$

where i_{l-1} is the $(l-1)$th customer in the trip, and there are l customers in total in the trip. Equation (27) means that the customer sequence is not allowed to be performed.

Subgradient cut. In Equation (8), $P(q_{ij})$ is a convex function in q_{ij}. Thus, the tangent line at point $(\bar{q}_{ij}, \bar{P}(\bar{q}_{ij}))$ (we use a bar '-' to represent known values) is

$$P(q_{ij}) = \bar{P}(\bar{q}_{ij}) + \bar{\beta}_{ij}(q_{ij} - \bar{q}_{ij}), \tag{28}$$

where $\bar{\beta}_{ij} = \frac{3}{2}k(W + m + \bar{q}_{ij})^{\frac{1}{2}}$, and it is the derivative. Figure 2 is an illustration of the tangent line. Therefore, the subgradient cut derived for constraints (11) can be added using a conditional form as follows:

$$e_{ij} \geq [\bar{P}(\bar{q}_{ij})x_{ij} + \bar{\beta}_{ij}(q_{ij} - \bar{q}_{ij})]/1000 \times (t_{ij}/3600) \quad \forall (i,j) \in A. \tag{29}$$

When $x_{ij} = 0$, the right-hand side of Equation (29) is a negative number ($q_{ij} = 0$ because of constraints (5)) and the cut is inactive. When $x_{ij} = 1$, the cut is added and the right-hand side of (29) underestimates the energy from Equation (8).

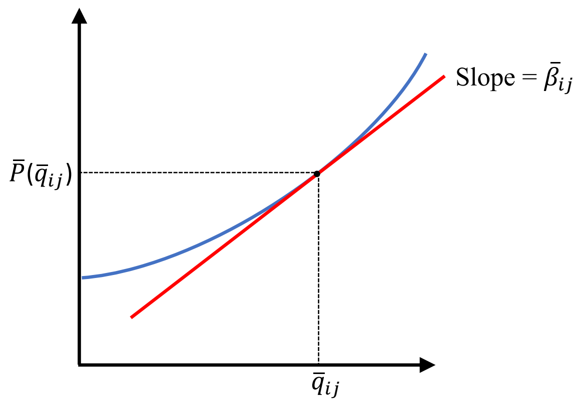

Figure 2: The tangent line of the power function

Remarks: (i) Being different from the logical cuts, constraints (10) and (12) are necessary when applying the subgradient cuts, and constraints (11) become $f_i + e_{ij} \leq M_{ij}(1 - x_{ij}) + f_j, \forall (i,j) \in A$. (ii) For the models *with* energy costs in the objective, i.e., the E and $R + E$ models, we must apply the subgradient cuts to ensure the involvement of energy cost. However, logical cuts are optional because the subgradient cuts can also guarantee that the energy capacity constraints are respected. (iii) For the models *without* energy costs, i.e., the R model, we can implement the cuts in three ways: only add logical cuts, only add subgradient cuts, or add both together. If there is only one customer in a trip, we do not add either logical or subgradient cuts for the R model, because we can guarantee that each customer is eligible to be serviced by a drone when generating the instance sets. Moreover, when only the logical cuts are used for model R, we do not need valid inequalities (26).

Our techniques can be applied for any energy function that is convex or piecewise convex in payload. If it is not a convex function, then our logical cut can be used. In other words, our method generalizes the ones presented in the literature.

4.2. Branch-and-Cut Algorithm

The B&C algorithm has been extensively used to solve MILP problems, and it is a combination of a cutting plane method with a B&B algorithm (Mitchell 2002). In our B&C scheme, we first add valid inequalities to the formulations at the root node of the search tree. We then solve the linear programming (LP) relaxation problem at each node of the tree. Each time a fractional solution is obtained, we detect and generate violated cuts in a cutting-plane fashion and the LP relaxation at the current B&B node is re-optimized. If all the cuts are respected and the solution still has fractional-valued integer variables, the branching process continues. If an integer solution is obtained and no cuts are generated, we consider updating the incumbent solution and pruning some nodes. This process continues until all nodes of the tree are evaluated.

4.2.1. Separation of Subtour Inequalities

Although constraints (4) can eliminate subtours, we introduce another group of subtour elimination constraints (SECs) which can help improve computational efficiency for the B&C scheme. The SECs are as follows (Laporte 1986):

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \leq |S| - q(S) \qquad \forall S \subseteq N', |S| \geq 2, \tag{30}$$

where $q(S) = \left\lceil \frac{\sum_{i\in S} d_i}{Q} \right\rceil$ is the minimum number of trips needed to visit customers in set S. The separation algorithm is performed by using the CVRPSEP package of Lysgaard et al. (2004).

4.2.2. Implementation of Cuts and SECs

For the logical and subgradient cuts, they are applied when an integer solution is obtained. For the SECs, we only generate them at the root node since they are redundant for our models due to the fact that subtours are eliminated by constraints (4) and it is time consuming to solve the separation problems at all nodes of the B&B tree.

5. Numerical Experiment

In this section, we present the instances and discuss our numerical tests for the MTDRP with the energy function presented in this paper. The B&C algorithms are coded in Python on Pycharm 2.7 using Gurobi 7.5.1 as the solver. All the parameters are set to their default values in the solver.

The experiments are performed on a cluster of Intel Xeon X5650 CPUs with 2.67 GHz and 24 GB RAM under Linux 6.3. Each experiment is conducted on a sing1e core of one node unless specified. The computing time limit is set to four hours.

5.1. Instance Sets

We introduce two sets of benchmark instances. The first set, named *Set A*, is created based on the instance generation frameworks presented in Solomon (1987) and Dorling et al. (2017). The second set, named *Set B*, is an extension of Solomon's instances, taking into account drones' specific characteristics. For Set A instances, we further consider two types of instances and each has 10–50 customers. For type 1 instances, named *Set* A_1, the depots are located at the lower left corner of the region. For type 2 instances, named *Set* A_2, the depots are in the middle of the region. We use Set A instances for preliminary tests and performance comparisons. We conduct experiments on Set B instances. The detailed instance generation procedures are presented in Appendix B. All the instances and solutions are also available at the following URL: `https://sites.google.com/view/chengchun/instances`.

We assume that 4-cell 14.8V lithium polymer batteries are used for drones. According to the field tests in Dorling et al. (2017), we set $\alpha = 0.217\ kW/kg$, $\beta = 0.185\ kW$, $m = 1.5\ kg$, $W = 1.5\ kg$, $Q = 1.5\ kg$, $g = 9.81\ N/kg$, $\rho = 1.204\ kg/m^3$, $\varsigma = 0.0064\ m^2$, $h = 6$, $\delta = 360\ \$/kWh$. For Set A instances, we set the battery energy capacity $\sigma = 0.27\ kWh$; For Set B instances, we set $\sigma = 0.027\ kWh$.

5.2. Enhancement Strategy Evaluation

This section analyzes the effect of valid inequalities and SECs. We conduct all the tests on instances with 10–30 customers in Set A. First, we only apply subgradient cuts to the model to evaluate the valid inequalities and SECs. After knowing the performances, we further compare different implementations of cuts. Results are provided in Table 2. For each model, we present detailed results of the largest instances (i.e., those with 30 customers) in the first six rows, and the results of all instances in the last two rows. The column *None* gives the results without any enhancement strategy. The remaining columns indicate that one (or all) valid inequalities or SECs are added to the model. *Opt* is the number of instances solved to optimality. *UP*, *LB*, and *RLB* are the best upper bound, the best lower bound, and the lower bound at the root node, respectively. *Gap* is the percentage difference between the best upper and lower bounds. *CPU* is the time in seconds consumed to solve the instance.

Table 2: Average results with different valid inequalities and SECs for Set A instances

			Only subgradient					Only logical	Subgradient +logical
	Cust		None	(25)	(26)	(25)+(26)	(25)+(26) +SECs	(25)+SECs	(25)+(26) +SECs
R_e	30	Opt	7/10⋆	7/10	5/10	7/10	7/10	7/10	**8/10**
		UB	11604.97	11608.23	11604.97	11616.74	11604.97	11604.97	11611.46
		LB	11539.12	11544.82	11520.78	11541.28	11553.25	11558.07	11575.70
		Gap	0.56	0.53	0.72	0.63	0.44	0.39	0.31
		CPU	6810.79	6038.29	10041.20	7436.01	6501.20	6033.06	6643.34
		RLB	11013.32	11041.65	11013.31	11033.22	11037.09	11055.78	11039.42
	All	Opt	46/50⋆	45/50	43/50	45/50	46/50	46/50	**47/50**
		Gap	0.14	0.16	0.22	0.19	0.13	0.11	0.10
E_e	30	Opt	0/10	0/10	3/10	3/10	4/10	Not Applicable	**6/10**
		UB	833.62	836.92	828.35	828.63	828.25		828.63
		LB	597.37	611.69	810.58	812.01	819.55		819.65
		Gap	28.24	26.82	2.24	2.07	1.05		1.07
		CPU	14400.00	14400.00	11515.46	12046.49	11437.83		9440.20
		RLB	123.92	80.82	705.74	704.07	708.00		708.00
	All	Opt	31/50	31/50	41/50	41/50	43/50		**46/50**
		Gap	8.83	8.06	0.54	0.50	0.22		0.22
$(R+E)_e$	30	Opt	6/10	5/10	6/10	6/10	**7/10**	Not Applicable	**7/10**
		UB	12451.89	12471.68	12437.36	12437.36	12437.36		12450.29
		LB	12321.26	12293.59	12343.51	12365.55	12369.68		12374.19
		Gap	1.03	1.40	0.75	0.57	0.54		0.60
		CPU	11631.34	9808.45	8577.97	8623.99	8512.23		8270.53
		RLB	11015.17	11181.01	11709.88	11742.26	11766.60		11766.70
	All	Opt	44/50	43/50	45/50	45/50	**46/50**		**46/50**
		Gap	0.30	0.38	0.19	0.18	0.16		0.18

⋆ indicates the number of instances (out of 10 and 50) that are solved to optimality.

Table 2 shows that different implementations of cuts yield different performances. In general, the simultaneous application of logical cuts, subgradient cuts, valid inequalities (25)–(26), and the SECs, gives the best performance for the three models. Specifically, a few more instances can be solved to optimality for the R_e and E_e models. For the $(R+E)_e$ model, the number of optimally solved instances is the same when only using the subgradient cut or using both cuts together; however, the average optimality gap is relatively close. We can also observe that, for instances with 30 customers, the average RLB is improved from 123.92 to 705.74 for model E_e when the valid inequalities based on the energy function (i.e., constraints (26)) are used. In addition, the E_e model consumes the most computation time on average, because its average RLB is not as tight as those of the other two models. In particular, RLB/LB= 0.86 for the E_e model whereas RLB/LB= 0.95

for the other two models. In the following sections, we use the 2-index formulation constructed in Section 3.2, together with constraints (25)–(26), SECs, and both logical and subgradient cuts, to perform our tests for each model.

5.3. *Details of Solutions for Set A Instances With Size 10–30*

Tables 3–4 give a summary details of results. *Cust* is the number of customers. *Log*, *Sub*, and *SECs* are the number of generated logical cuts, subgradient cuts, and SECs, respectively. In Table 4, *UAVs* is the number of drones used, and *Swap* represents the average number of battery swaps. When calculating *Swap*, we do not count the first trip performed by a drone. For example, if a drone has conducted 3 trips, then the value of *Swap* would be 2. *T/d* indicates the average number of trips performed by each drone. The last column in Table 4 is the proportion of energy cost to total cost. More detailed results for each instance are presented in Appendix C.

Table 3: Average results on cuts for Set A instances with size 10–30

		R_e						E_e						$(R+E)_e$					
	Cust	Opt	Gap	CPU	Log	Sub	SECs	Opt	Gap	CPU	Log	Sub	SECs	Opt	Gap	CPU	Log	Sub	SECs
Set A_1	10	5/5	0.0	0.5	0.2	92.2	21.4	5/5	0.0	0.9	0.4	138.2	31.4	5/5	0.0	0.6	0.0	134.8	25.2
	15	5/5	0.0	176.6	3.4	400.6	27.6	5/5	0.0	82.6	2.6	595.4	33.2	5/5	0.0	306.2	2.8	518.6	32.6
	20	5/5	0.0	162.2	1.6	349.8	35.4	5/5	0.0	72.3	1.6	558.2	39.4	5/5	0.0	224.0	0.6	615.6	35.6
	25	4/5	0.4	4666.1	3.0	552.4	34.2	5/5	0.0	4508.6	4.8	1282.0	39.8	4/5	0.6	4945.6	4.4	1512.0	38.2
	30	3/5	0.6	6860.3	3.8	1172.8	42.4	3/5	1.0	8989.8	3.4	1884.6	47.8	3/5	0.7	8865.7	2.8	1759.0	47.0
Set A_2	10	5/5	0.0	0.4	0.0	40.8	16.6	5/5	0.0	0.5	0.4	142.6	13.8	5/5	0.0	0.3	0.0	71.4	9.8
	15	5/5	0.0	5.0	2.4	152.2	34.0	5/5	0.0	4.1	1.4	274.4	27.6	5/5	0.0	5.7	1.8	297.8	29.4
	20	5/5	0.0	27.0	2.0	259.0	34.6	5/5	0.0	37.3	1.6	487.6	40.0	5/5	0.0	33.1	1.4	500.8	36.4
	25	5/5	0.0	202.3	2.2	469.4	36.6	5/5	0.0	301.0	4.0	921.0	37.2	5/5	0.0	313.1	2.4	853.0	39.4
	30	5/5	0.0	6426.4	3.2	927.2	41.2	3/5	1.2	9890.6	5.0	1815.8	35.4	4/5	0.5	7675.3	4.0	1437.4	39.6

Table 4: Average results on drones for Set A instances with size 10–30

		R_e			E_e			$(R+E)_e$			
	Cust	UAVs	Swap	T/d	UAVs	Swap	T/d	UAVs	Swap	T/d	Cost_e (%)
Set A_1	10	2.0	3.2	2.6	2.0	3.2	2.6	2.0	3.2	2.6	6.6
	15	2.2	4.0	2.9	2.2	4.2	3.0	2.2	4.0	2.9	6.6
	20	3.6	6.2	2.8	3.6	6.2	2.8	3.6	6.2	2.8	6.8
	25	3.8	7.4	3.0	3.8	7.6	3.0	3.8	7.6	3.0	6.8
	30	5.0	8.4	2.7	5.0	8.4	2.7	5.0	8.2	2.6	6.8
Set A_2	10	2.0	3.4	2.7	2.0	3.6	2.8	2.0	3.4	2.7	6.4
	15	2.8	5.6	3.1	2.8	6.2	3.3	2.8	5.6	3.1	6.4
	20	3.4	7.2	3.2	3.4	7.4	3.2	3.4	7.2	3.2	6.6
	25	4.0	8.6	3.2	4.0	9.2	3.3	4.0	8.6	3.2	6.7
	30	4.4	9.6	3.2	4.4	10.0	3.3	4.4	9.6	3.2	6.6
Average		3.3	6.4	2.9	3.3	6.6	3.0	3.3	6.4	2.9	6.6

Table 3 shows that the number of subgradient cuts is much larger than that of the logical cuts. This is because the subgradient cuts are produced for edges while the logical cuts are generated for trips. For the largest problems (30 customers), the R_e model consumes the least computing time and the E_e model consumes the most computing time. Further, instances in Set A_1 require more time than those in Set A_2, because the locations of the depot and customers are more geographically dispersed in Set A_1.

Table 4 indicates that more drones are used with an increasing number of customers and that, in most cases, each drone performs 2 or 3 trips. For the $(R + E)_e$ model, the energy cost only accounts for a small portion (around 6.6%) of the total cost. The average results seem similar for the three models; however, with different objectives, different schedules are indeed generated for some instances. An example is given in Table 5. It shows that two more trips are performed for the E_e model, leading to a greater travel distance and a lower energy consumption. Moreover, we find that for this instance the schedules generated by the R_e and $(R+E)_e$ models are quite similar, except that the travel direction of the second and fifth trips are opposite. Since we perform our tests on an undirected network, travel direction influences energy consumption because of different payloads on arcs. However, as the $(R + E)_e$ model includes the energy cost in the objective, it can always guarantee that drones travel in directions with minimal energy consumption. Thus, in realistic applications, for undirected networks, even though decision makers favor a VRP objective which minimizes the travel cost, they can still add energy cost in the objective and set a small value for energy price to save battery energy consumption and further reduce the recharging time.

Table 5: Schedules generated by different objectives for instance *Set_A2_Cust_15_2*

R_e		E_e		$(R+E)_e$	
Trips	Energy (kWh)	Trips	Energy (kWh)	Trips	Energy (kWh)
[0, 3, 1, 16]	0.1585	[0, 3, 1, 16]	0.1585	[0, 3, 1, 16]	0.1585
[0, 4, 2, 16]	**0.2389**	[0, 4, 2, 16]	0.2389	[0, 2, 4, 16]	**0.2344**
[0, 5, 10, 15, 16]	0.2099	[0, 10, 5, 16]	0.0937	[0, 5, 10, 15, 16]	0.2099
[0, 6, 12, 16]	0.2530	[0, 12, 16]	0.1645	[0, 6, 12, 16]	0.2530
[0, 7, 8, 16]	**0.1733**	[0, 8, 7, 16]	0.1637	[0, 8, 7, 16]	**0.1637**
[0, 9, 11, 16]	0.2418	[0, 9, 16]	0.1354	[0, 11, 9, 16]	0.1835
[0, 13, 16]	0.1690	[0, 13, 16]	0.1690	[0, 13, 16]	0.1690
[0, 14, 16]	0.1341	[0, 14, 16]	0.1341	[0, 14, 16]	0.1341
		[0, 11, 16]	0.0462		
		[0, 15, 6, 16]	0.1794		
Total energy (kWh)	1.5785		1.4834		1.5061
Total travel distance	7995.39		8153.26		7995.39

5.4. Performance Comparison Between Models with Nonlinear and Linear Energy Functions

In this section, we compare the model performance with our nonlinear and linear energy models. Table 6 presents a summary of results, with some detailed results in Appendix C. For solutions generated by the linear approximation models, after obtaining the trips, we calculate the energy consumption using the nonlinear model (8) for each trip and report the average results in the last two rows of Table 6. *Infeasible* is the number of instances for which the linear approximation models yield trips that when energy is calculated with (8) exceed the energy capacity. *Energy gap* is the percentage difference in energy calculation, which is computed as *(energy from (8) − energy from (9))/energy from (9)*.

Table 6: Statistics of solutions generated by models R and $R+E$ with nonlinear and linear energy functions

	model R		model $R+E$	
Energy function	Nonlinear	Linear	Nonlinear	Linear
Opt	47/50	49/50	46/50	50/50
Optimality gap	0.10	0.04	0.18	0.00
CPU	1852.69	911.59	2236.96	743.95
Travel distance	8276.33	8227.72	8278.71	8227.68
Infeasible	0/50	**20/50**	0/50	**18/50**
Energy gap (%)	0.00	**9.45**	0.00	**9.32**

From Table 6, we get two observations: (1) ***Computational efficiency***. For both models, the computational efficiency with the linear approximation (Equation (9)) is better than that of the nonlinear method (Equation (8)). By using the approximation method, more instances can be solved in a shorter time frame. For the R and $R+E$ models with the nonlinear energy function, the average computation times are 1852.69 and 2236.96 seconds, respectively. Thus, we can conclude that, even though our original models are nonlinear, the use of logical and subgradient cuts can help solve large problems to optimality. (2) ***Feasibility and solution quality***. In multiple instances, the approximation models yield "energy infeasible" trips when energy is calculated based on the nonlinear model (8). For the R and $R+E$ models, the approximation method produces infeasible trips for 20 and 18 instances respectively. In addition, the energy gap is around 9% on average between the two methods.

To further display the importance of how energy is calculated, we give an example in Table 7 to show the different schedules generated by the two methods. It demonstrates that the first trip given by the two approximation models consumes 0.2925 kWh energy, which violates the battery's

Table 7: Detailed solutions of models with nonlinear and linear energy functions for instance *Set_A1_Cust_25_2*

	Nonlinear energy function		Linear energy function			
				Energy consumption		
	Trips	Energy	Trips	Linear	Nonlinear	Energy gap(%)
R	[0, 1, 14, 2, 6, 26]	0.2685	**[0, 1, 14, 2, 6, 16, 26]**	0.2656	0.2925	10.13
	[0, 4, 7, 8, 26]	0.1348	[0, 7, 8, 26]	0.1147	0.1240	8.11
	[0, 5, 20, 26]	0.2056	[0, 5, 20, 26]	0.1857	0.2056	10.72
	[0, 13, 11, 26]	0.1501	[0, 13, 11, 26]	0.1378	0.1501	8.93
	[0, 15, 3, 26]	0.1471	[0, 15, 3, 26]	0.1346	0.1471	9.29
	[0, 17, 10, 24, 12, 25, 26]	0.2528	[0, 12, 24, 10, 17, 4, 26]	0.1965	0.2175	10.69
	[0, 18, 9, 16, 26]	0.2138	[0, 25, 18, 9, 26]	0.1944	0.2146	10.39
	[0, 19, 21, 26]	0.1551	[0, 21, 19, 26]	0.1320	0.1447	9.62
	[0, 22, 26]	0.0134	[0, 22, 26]	0.0127	0.0134	5.51
	[0, 23, 26]	0.0684	[0, 23, 26]	0.0622	0.0684	9.97
$R+E$	[0, 1, 14, 2, 6, 26]	0.2685	**[0, 1, 14, 2, 6, 16, 26]**	0.2656	0.2925	10.13
	[0, 4, 7, 8, 26]	0.1348	[0, 8, 7, 26]	0.1099	0.1177	7.10
	[0, 20, 5, 26]	0.2039	[0, 20, 5, 26]	0.1845	0.2039	10.51
	[0, 13, 11, 26]	0.1501	[0, 13, 11, 26]	0.1378	0.1501	8.93
	[0, 15, 3, 26]	0.1471	[0, 15, 3, 26]	0.1346	0.1471	9.29
	[0, 17, 10, 24, 12, 25, 26]	0.2528	[0, 4, 17, 10, 24, 12, 26]	0.1903	0.2091	9.88
	[0, 18, 9, 16, 26]	0.2138	[0, 25, 18, 9, 26]	0.1944	0.2146	10.39
	[0, 21, 19, 26]	0.1447	[0, 21, 19, 26]	0.1320	0.1447	9.62
	[0, 22, 26]	0.0134	[0, 22, 26]	0.0127	0.0134	5.51
	[0, 23, 26]	0.0684	[0, 23, 26]	0.0622	0.0684	9.97

energy capacity (0.27 kWh). However, if the linear approximation method is used, it will consider these trips as feasible ones. Therefore, care is needed when modeling energy consumption to ensure energy feasibility of routes.

5.5. *Impact of Time Windows*

Here, we first consider new instances with tighter time windows at customers. We generate the width of customers' time windows according to a new normal distribution whose mean is $0.15(b_{n+1} - t_{j,n+1} - t_{0j})$, and keep other data unchanged. Next, we remove the time constraints (14)–(16) and solve a multi-trip drone routing problem. Summary results are reported in Table 8 and detailed results are presented in Appendix C.

Table 8: Average results for models with tighter time windows and without time windows

		model R_e						model E_e						model $(R+E)_e$					
		Tighter time windows			Without time windows			Tighter time windows			Without time windows			Tighter time windows			Without time windows		
	Cust	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU	Opt	Gap	CPU
Set A_1	10	5/5	0.0	0.2	5/5	0.0	0.5	5/5	0.0	0.3	5/5	0.0	1.0	5/5	0.0	0.3	5/5	0.0	0.4
	15	5/5	0.0	32.2	5/5	0.0	527.4	5/5	0.0	47.8	5/5	0.0	44.6	5/5	0.0	41.7	5/5	0.0	129.5
	20	5/5	0.0	82.1	5/5	0.0	387.1	5/5	0.0	106.8	5/5	0.0	200.7	5/5	0.0	87.8	5/5	0.0	287.5
	25	5/5	0.0	2377.8	4/5	0.7	5948.0	3/5	1.6	6319.1	4/5	0.4	5641.6	5/5	0.0	3179.9	4/5	0.8	5061.6
	30	3/5	0.6	6565.7	1/5	0.5	12339.4	3/5	1.1	8552.8	2/5	1.3	12211.0	3/5	0.7	6425.4	1/5	0.8	12468.2
Set A_2	10	5/5	0.0	0.1	5/5	0.0	1.5	5/5	0.0	0.2	5/5	0.0	0.9	5/5	0.0	0.2	5/5	0.0	1.0
	15	5/5	0.0	1.0	5/5	0.0	8.8	5/5	0.0	1.3	5/5	0.0	4.4	5/5	0.0	1.2	5/5	0.0	9.6
	20	5/5	0.0	10.1	5/5	0.0	97.4	5/5	0.0	19.3	5/5	0.0	47.2	5/5	0.0	13.4	5/5	0.0	73.4
	25	5/5	0.0	140.2	5/5	0.0	1183.1	5/5	0.0	240.7	5/5	0.0	119.2	5/5	0.0	121.8	5/5	0.0	1006.6
	30	5/5	0.0	646.5	3/5	0.9	8849.3	5/5	0.0	2458.6	3/5	0.5	7969.1	5/5	0.0	507.5	4/5	0.5	6988.8

From Table 3 and Table 8, when the time windows are tighter, one more instance can be solved to optimality for the R_e model, and two more instances for the $(R+E)_e$ model. Moreover, for problems of the same size, the average computation time is generally reduced for the three models. However, when the time constraints are absent, instances become much more difficult to handle. Fewer instances can be solved to optimality within the time limit and the average computation time also increases. These observations are consistent with the results provided by Azi (2011), where a B&P algorithm is used for the MTVRPTW.

5.6. Algorithm Performance on Extended Solomon's Instances

In this section we test our algorithm on Set B instances based on the well-known Solomon's instances. All the experiments are performed on 4 core processors with a 12-hour (43200 seconds) time limit. Summarized results are shown in Table 9 and detailed results on each instance are provided in Appendix C. In Table 9, column *Inst* is the instance label.

Table 9: Algorithm performance on Solomon's instances of type 2

	25 customers						40 customers					
	model R_e		model E_e		model $(R+E)_e$		model R_e		model E_e		model $(R+E)_e$	
Inst	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU	Gap	CPU
c201	0.00	1.4	0.00	11.5	0.00	2.5	0.00	48.3	0.00	229.9	0.00	43.8
c202	0.00	13.4	0.00	48.5	0.00	19.7	0.00	372.9	0.00	14073.8	0.00	955.4
c203	0.00	56.4	0.00	86.5	0.00	74.8	0.00	3881.6	0.95	43200.0	0.00	10685.1
c204	0.00	48.2	0.00	240.8	0.00	78.6	0.20	43200.0	1.55	43200.0	0.00	16940.6
c205	0.00	13.0	0.00	29.5	0.00	9.0	0.00	871.2	0.00	23630.3	0.00	881.7
c206	0.00	21.1	0.00	40.8	0.00	21.7	0.00	2011.9	1.83	43200.0	0.00	8368.8
c207	0.00	32.0	0.00	55.3	0.00	43.4	0.00	5757.3	0.00	5567.9	0.00	8358.9
c208	0.00	23.2	0.00	34.7	0.00	34.3	0.00	1936.7	0.63	43200.0	0.00	3547.3
r201	0.00	4.0	0.00	8.7	0.00	5.6	0.00	437.7	0.00	2730.5	0.00	348.1
r202	0.00	32.9	0.00	26.9	0.00	29.2	0.00	11666.6	1.22	43200.0	0.00	8183.1
r203	0.00	132.1	0.00	40.2	0.00	58.0	0.00	33302.4	0.00	17013.0	0.00	40920.9
r204	0.00	134.2	0.00	125.4	0.00	120.3	0.58	43200.0	0.00	37471.4	1.22	43200.0
r205	0.00	38.5	0.00	21.8	0.00	27.1	0.00	5921.7	0.00	7086.4	0.00	8098.5
r206	0.00	54.8	0.00	29.8	0.00	84.6	0.96	43200.0	0.00	5167.2	0.42	43200.0
r207	0.00	83.5	0.00	37.4	0.00	87.8	1.18	43200.0	0.00	22529.7	0.00	39388.0
r208	0.00	75.7	0.00	47.9	0.00	106.7	0.96	43200.0	0.36	43200.0	0.67	43200.0
r209	0.00	42.8	0.00	41.5	0.00	50.0	0.00	42044.0	5.91	43200.0	1.40	43200.0
r210	0.00	46.0	0.00	24.8	0.00	60.2	0.00	14821.3	0.00	2832.5	0.68	43200.0
r211	0.00	136.1	0.00	53.4	0.00	102.5	1.44	43200.0	0.62	43200.0	1.16	43200.0
rc201	0.00	28.0	0.00	31.7	0.00	60.0	0.00	959.5	7.42	43200.0	0.00	2540.3
rc202	0.00	366.9	0.00	315.0	0.00	125.1	0.79	43200.0	2.86	43200.0	0.62	43200.0
rc203	0.00	15.7	0.00	29.9	0.00	56.8	0.00	1542.6	0.00	509.2	0.00	5064.3
rc204	0.00	5.4	0.00	64.3	0.00	952.8	0.00	110.3	0.00	2663.9	0.00	9900.8
rc205	0.00	63.4	0.00	269.0	0.00	100.6	0.00	27642.5	13.59	43200.0	1.58	43200.0
rc206	0.00	65.4	0.00	59.5	0.00	755.0	0.00	4719.6	10.42	43200.0	0.00	39338.2
rc207	0.00	1253.2	0.00	58.3	0.00	7306.9	0.39	43200.0	1.10	43200.0	0.91	43200.0
rc208	0.00	207.4	0.00	23.0	0.00	157.5	0.00	1684.8	0.00	237.6	0.00	7961.0
Average	$0.00^{(0)}$	110.9	$0.00^{(0)}$	68.7	$0.00^{(0)}$	390.0	$0.24^{(8)}$	18716.0	$1.80^{(13)}$	26049.8	$0.32^{(9)}$	22234.3

$^{(-)}$ indicates the number of instances (out of 27) that are not solved to optimality.

We can observe that all instances with 25 customers are solved to optimality within the time limit. When the number of customers increases to 40, 19 out of 27 instances are optimally solved for model R_e, and this number decreases to 14 and 18 for model E_e and model $(R+E)_e$, respectively. The CPU time also varies widely, ranging from a few minutes to many hours. In terms of computational performance as opposed to the MTVRPTW which is relatively similar to the MTDRP considered in this paper, our algorithms could generally solve larger instance sizes compared to those considered in exact algorithms for the MTVRPTW despite the fact that our original models are nonlinear and more complex.

5.7. Results for Large Instances of Set A

Here, we report the results of Set A instances with 35–50 customers in Table 10. More detailed results are given in Appendix C. All the experiments are performed on 4 core processors with a 12-hour time limit. The instances with 10–30 customers that were not optimally solved in previous experiments are also solved again with the longer time limit. Our results show that all the previous instances, except *Set_A1_Cust_30_5* for model E_e, are solved to optimality under the new experiment setting. The optimality gap of this instance for model E_e is 1.77%. For some instances, when we directly solve the E_e model or the $(R+E)_e$ model, we find that the optimality gap is over 5% within the time limit, mainly resulting from the poor lower bound. Considering the R_e model is relatively easier than the other two models, for these specific instances, we first solve the R_e model to get a feasible solution and then use this solution as a start for the other two models. The results of these instances are marked by a square in Table 10.

Table 10 shows that the average gap ranges from 1.81% to 2.28% for instances in Set A_1 and from 1.29% to 1.50% for instances in Set A_2, which further confirms our previous observation that generally instances in Set A_2 are easier than those in Set A_1. For the R_e model, 13 out of 35 instances are solved to optimality. For the E_e and $(R+E)_e$ models, the number of optimally solved instances are 12 and 10 respectively. We also note that it is effective to use the first solution of the R_e model as a start for the other two models. In particular, for the E_e model, 5 instances can be solved to optimality by using this method. We further use this idea to model E_e for Solomon's r209, rc201, rc205, and rc206 instances with 40 customers (i.e., instances whose optimality gap is over 5% in Table 9). The results show that all these instances can be solved to optimality now.

Table 10: Results using multicore processors for Set A instances with 35–50 customers

			model R_e		model E_e		model $(R+E)_e$	
	Cust	Inst	Gap	CPU	Gap	CPU	Gap	CPU
Set A$_1$	35	1	4.11	43200.0	4.55$^{\square}$	43200.0	3.24$^{\square}$	43200.0
		2	1.80	43200.0	3.58$^{\square}$	43200.0	2.78	43200.0
		3	0.00	20642.0	0.00$^{\square}$	21866.9	0.98	43200.0
		4	0.00	30214.8	0.00$^{\square}$	36895.7	0.21	43200.0
		5	0.00	29126.0	0.00	15121.7	0.00	20446.6
	40	1	3.38	43200.0	2.92	43200.0	3.95	43200.0
		2	0.00	13947.5	0.90	43200.0	0.59	43200.0
		3	3.74	43200.0	4.74	43200.0	3.78$^{\square}$	43200.0
		4	0.44	43200.0	0.00$^{\square}$	23208.1	0.39	43200.0
		5	0.73	43200.0	1.30$^{\square}$	43200.0	2.32	43200.0
	45	1	4.24	43200.0	3.65$^{\square}$	43200.0	3.96	43200.0
		2	2.15	43200.0	2.66$^{\square}$	43200.0	2.06$^{\square}$	43200.0
		3	1.51	43200.0	3.76	43200.0	2.46$^{\square}$	43200.0
		4	3.05	43200.0	4.42$^{\square}$	43200.0	3.15$^{\square}$	43200.0
		5	1.95	43200.0	1.79$^{\square}$	43200.0	2.29	43200.0
	Average		1.81	37942.0	2.28	38152.8	2.14	41683.1
Set A$_2$	35	1	2.53	43200.0	0.00$^{\square}$	31873.4	2.65	43200.0
		2	0.00	1755.4	0.00	18306.0	0.00	3397.5
		3	0.00	8732.7	2.83	43200.0	0.00	11645.0
		4	0.00	9765.1	0.00	38648.4	0.00	25076.3
		5	0.00	9491.5	0.00	42292.8	0.00	18041.8
	40	1	0.00	32162.3	0.00	6219.5	0.00	21628.1
		2	1.12	43200.0	1.44	43200.0	0.00	41897.4
		3	0.00	3298.6	0.00	30554.3	0.00	5495.7
		4	2.13	43200.0	0.00$^{\square}$	7308.4	2.16	43200.0
		5	5.05	43200.0	6.39$^{\square}$	43200.0	4.81	43200.0
	45	1	0.00	6142.2	0.00	1802.5	0.00	8093.6
		2	0.00	41018.0	1.30	43200.0	1.32	43200.0
		3	1.20	43200.0	2.15	43200.0	0.00	7452.0
		4	2.02	43200.0	1.85$^{\square}$	43200.0	2.68	43200.0
		5	0.00	37956.5	2.11	43200.0	1.00	43200.0
	50	1	1.80	43200.0	0.74$^{\square}$	43200.0	2.36	43200.0
		2	3.81	43200.0	2.24$^{\square}$	43200.0	3.80	43200.0
		3	2.92	43200.0	4.20$^{\square}$	43200.0	2.72	43200.0
		4	1.32	43200.0	3.29	43200.0	0.72$^{\square}$	43200.0
		5	1.80	43200.0	1.55$^{\square}$	43200.0	2.01$^{\square}$	43200.0
	Average		1.29	31276.1	1.50	34770.27	1.31	30896.4

$^{\square}$ We use the first feasible solution of the R model as an initial solution for this model.

6. Conclusions

This paper solves a MTDRP with time windows. A 2-index formulation is introduced and a B&C algorithm is developed. We propose two types of cuts to tackle the nonlinear energy function. We demonstrate the differences between using a complex nonlinear energy consumption function and a linear approximation, which can result in higher energy use and energy infeasible drone routes. We generate benchmark instances for the drone routing problem and conduct extensive numerical experiments to evaluate the effects of valid inequalities and user cuts. The effectiveness of our modeling scheme and the B&C algorithm is confirmed by solving generated instances and Solomon's type 2 instances.

The limitations of the energy function used in our work include: (1) We did not consider the energy consumption of other flight status like taking off and landing. (2) Some other factors such as drone speed and wind speed were neglected in our power function. (3) The parameters associated with the drones considered in our paper are small drones with limited payloads and a low travel speed; however, the recently developed drone models by Amazon and UPS can carry payloads of up to five pounds and fly at speeds up to 50 *mph*. Thus, we consider future research which can extend this work in the following aspects: (1) More complex power models with additional influence factors can be used for drone energy calculation. In this case, the energy consumption of other flight status should also be explicitly incorporated into the mathematical model. (2) Numerical tests can be conducted by using the parameters collected from production level delivery drones to provide operational insights for decision-makers.

Acknowledgements

The first author wishes to acknowledge the supporters of this study: the Institute for Data Valorization (IVADO) and the Interuniversity Research Center on Enterprise Networks, Logistics and Transportation (CIRRELT). We thank Calcul Québec for providing computing facilities for the experiments. We also thank the associate editor and two anonymous referees for their helpful comments.

References

Agatz, N., Bouman, P., Schmidt, M., 2018. Optimization approaches for the traveling salesman problem with drone. Transportation Science, 52 (4), 965–981.

Azi, N., 2011. Méthodes exactes et heuristiques pour le problème de tournées de véhicules avec fenêtres de temps et réutilisation de véhicules. Ph.D. thesis, Université de Montréal.

Azi, N., Gendreau, M., Potvin, J.-Y., 2014. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Computers & Operations Research 41, 167–173.

Bouman, P., Agatz, N., Schmidt, M., 2018. Dynamic programming approaches for the traveling salesman problem with drone. Networks 72 (4), 528–542.

Campbell, J. F., Sweeney, D., II, Z. J., 2017. Strategic design for delivery with trucks and drones. Technical Report.

Carlsson, J. G., Song, S., 2017. Coordinated logistics with a truck and a drone. Management Science, 64 (9), 4052–4069.

Cattaruzza, D., Absi, N., Feillet, D., 2016. Vehicle routing problems with multiple trips. 4OR 14 (3), 223–259.

Choi, Y., Schonfeld, P. M., 2017. Optimization of multi-package drone deliveries considering battery capacity. In: 96th Annual Meeting of the Transportation Research Board, Washington, DC (Paper No. 17–05769).

Chowdhury, S., Emelogu, A., Marufuzzaman, M., Nurre, S. G., Bian, L., 2017. Drones for disaster response and relief operations: A continuous approximation model. International Journal of Production Economics 188, 167–184.

D'Andrea, R., 2014. Guest editorial can drones deliver? IEEE Transactions on Automation Science and Engineering 11 (3), 647–648.

Desaulniers, G., Madsen, O. B., Ropke, S., 2014. Chapter 5: The vehicle routing problem with time windows. In: Vehicle Routing: Problems, Methods, and Applications, 2nd edition. SIAM, Philadelphia, pp. 119–159.

Doherty, J., 2019. Alphabet's Wing begins first commercial drone delivery service in U.S., beating Amazon, Uber. Accessed December 6, 2019, `https://www.newsweek.com/wing-drone-first-commercial-delivery-1466471`.

Dorling, K., Heinrichs, J., Messier, G. G., Magierowski, S., 2017. Vehicle routing problems for drone delivery. IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (1), 70–85.

Figliozzi, M. A., 2017. Lifecycle modeling and assessment of unmanned aerial vehicles (Drones) CO_2e emissions. Transportation Research Part D: Transport and Environment 57, 251–261.

Fleischmann, B., 1990. The vehicle routing problem with multiple use of the vehicles. Working Paper, Facbereich Wirtschaftswissenschafte, Universität Hamburg.

Ham, A. M., 2018. Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. Transportation Research Part C: Emerging Technologies 91, 1–14.

Hernandez, F., Feillet, D., Giroudeau, R., Naud, O., 2014. A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. 4OR 12 (3), 235–259.

Hernandez, F., Feillet, D., Giroudeau, R., Naud, O., 2016. Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. European Journal of Operational Research 249 (2), 551–559.

Heutger, M., Kückelhaus, M., 2014. Unmanned aerial vehicle in logistics: a DHL perspective on implications and use cases for the logistics industry. Technical Report, DHL Trend Research, http://www.dhl.com/content/dam/downloads/g0/about_us/logistics_insights/dhl_trend_report_uav.pdf.

Jeong, H. Y., Song, B. D., Lee, S., 2019. Truck-drone hybrid delivery routing: Payload-energy dependency and No-Fly zones. International Journal of Production Economics 214, 220–233.

Karaoğlan, İ., 2015. A branch-and-cut algorithm for the vehicle routing problem with multiple use of vehicles. International Journal of Lean Thinking 6 (1), 21–46.

Kirschstein, T., 2020. Comparison of energy demands of drone-based and ground-based parcel delivery services. Transportation Research Part D: Transport and Environment 78, 102209.

Laporte, G., 1986. Generalized subtour elimination constraints and connectivity constraints. Journal of the Operational Research Society 37 (5), 509–514.

Leishman, G. J., 2006. Principles of helicopter aerodynamics, 2nd edition. Cambridge university press, New York.

Liu, Z., Sengupta, R., Kurzhanskiy, A., 2017. A power consumption model for multi-rotor small unmanned aircraft systems. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, pp. 310–315.

Luo, Z., Liu, Z., Shi, J., 2017. A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. Sensors 17 (5), 1144.

Lysgaard, J., Letchford, A. N., Eglese, R. W., 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming 100 (2), 423–445.

Macedo, R., Alves, C., de Carvalho, J. V., Clautiaux, F., Hanafi, S., 2011. Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. European Journal of Operational Research 214 (3), 536–545.

Marinelli, M., Caggiani, L., Ottomanelli, M., Dell'Orco, M., 2017. En-route truck-drone parcel delivery for optimal vehicle routing strategies. IET Intelligent Transport Systems 12 (4), 253–261.

Mathew, N., Smith, S. L., Waslander, S. L., 2015. Planning paths for package delivery in heterogeneous multirobot teams. IEEE Transactions on Automation Science and Engineering 12 (4), 1298–1308.

Mingozzi, A., Roberti, R., Toth, P., 2013. An exact algorithm for the multitrip vehicle routing problem. INFORMS Journal on Computing 25 (2), 193–207.

Mitchell, J. E., 2002. Branch-and-cut algorithms for combinatorial optimization problems. Handbook of applied optimization, 65–77.

Morgan, P., 2005. Carbon fibers and their composites. CRC press, Florida.

Moshref-Javadi, M., Lee, S., 2017. Using drones to minimize latency in distribution systems. In: IIE Annual Conference. Proceedings. Institute of Industrial and Systems Engineers (IISE), pp. 235–240.

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

Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E., 2018. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. Networks 0 (0), 1–48.

Poikonen, S., Golden, B., Wasil, E. A., 2019. A branch-and-bound approach to the traveling salesman problem with a drone. INFORMS Journal on Computing 31 (2), 335–346.

Poikonen, S., Wang, X., Golden, B., 2017. The vehicle routing problem with drones: Extended models and connections. Networks 70 (1), 34–43.

Ponza, A., 2016. Optimization of drone-assisted parcel delivery. Master's thesis, University of Padova, Italy.

Pugliese, L. D. P., Guerriero, F., 2017. Last-mile deliveries by using drones and classical vehicles. In: International Conference on Optimization and Decision Science. Springer, pp. 557–565.

Raj, R., Murray, C. C., 2020. Fly slower, deliver faster: The multiple flying sidekicks traveling salesman problem with variable drone speeds. `https://ssrn.com/abstract=3549622`.

Reddy, T. B., 2010. Chapter 14: Lithium primary batteries. In: Linden's Handbook of Batteries, 4th edition. McGraw-Hill Professional, New York, pp. 14.1–14.90.

Rivera, J. C., Afsar, H. M., Prins, C., 2013. Multistart evolutionary local search for a disaster relief problem. In: International Conference on Artificial Evolution (Evolution Artificielle). Springer, pp. 129–141.

Rose, C., 2013. Amazon's Jeff Bezos Looks to the Future. Accessed December 6, 2019, `https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/`.

San, K. T., Lee, E. Y., Chang, Y. S., 2016. The delivery assignment solution for swarms of uavs dealing with multi-dimensional chromosome representation of genetic algorithm. In: Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual. IEEE, pp. 1–7.

Santos, F. A., Mateus, G. R., da Cunha, A. S., 2014. A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. Transportation Science 49 (2), 355–368.

Semet, F., Toth, P., Vigo, D., 2014. Chapter 2: Classical exact algorithms for the capacitated vehicle routing problem. In: Vehicle Routing: Problems, Methods, and Applications, Second Edition. SIAM, Philadelphia, pp. 37–57.

Solomon, M. M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35 (2), 254–265.

Stolaroff, J. K., Samaras, C., ONeill, E. R., Lubers, A., Mitchell, A. S., Ceperley, D., 2018. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. Nature Communications 9 (1), 409.

Troudi, A., Addouche, S.-A., Dellagi, S., Mhamedi, A., 2018. Sizing of the drone delivery fleet considering energy autonomy. Sustainability 10 (9), 3344.

Ulmer, M. W., Thomas, B. W., 2018. Same-day delivery with a heterogeneous fleet of drones and vehicles. Networks 72 (4), 475–505.

Wang, X., Poikonen, S., Golden, B., 2017. The vehicle routing problem with drones: Several worst-case results. Optimization Letters 11 (4), 679–697.

Wang, Z., Liang, W., Hu, X., 2014. A metaheuristic based on a pool of routes for the vehicle routing problem with multiple trips and time windows. Journal of the Operational Research Society 65 (1), 37–48.

Wang, Z., Sheu, J.-B., 2019. Vehicle routing problem with drones. Transportation Research Part B: Methodological 122, 350–364.

Zhang, J., Campbell, J. F., Sweeney, D., II, Hupman, A. C., 2020. Energy consumption models for delivery drones: A comparison and assessment.

Appendix A. Drone Energy Consumption for Waiting Time at Customer Locations

We can incorporate a non-zero energy consumption for waiting time at customer locations in our model. Specifically, we introduce new variables $w_i, \forall i \in N'$ to represent the waiting time for the opening of the time window at customer $i \in N'$. Variables $\tau_i, \forall i \in N^-$ now denote drones' arrival time at node $i \in N^-$. Correspondingly, constraints (14)–(16) are modified as follows to include w_i in the model

$$\tau_i + w_i + t_{ij} - M''_{ij}(1 - x_{ij}) \leq \tau_j \qquad \forall i \in N', j \in N^-, \tag{A.1}$$

$$a_i \leq \tau_i + w_i \leq b_i \qquad \forall i \in N', \tag{A.2}$$

$$a_{n+1} \leq \tau_{n+1} \leq b_{n+1}, \tag{A.3}$$

$$\tau_i + w_i + (t_{i,n+1} + t_{0j}) \leq \tau_j + (1 - z_{ij})M'''_{ij} \qquad \forall i, j \in N', i \neq j, \tag{A.4}$$

$$w_i \geq 0 \qquad \forall i \in N'. \tag{A.5}$$

Now we set $M''_{ij} = b_i + t_{ij}$, and M'''_{ij} take the same values as before. We assume the unit energy consumption for waiting (e.g., performing sensing activities, hovering, etc.) as γ (kWh/s). Then, constraints (11) are replaced by

$$f_0 + k'(W + m + q_{0j})^{\frac{3}{2}} t_{0j} \leq M_{0j}(1 - x_{0j}) + f_j \quad \forall j \in N', \tag{A.6}$$

$$f_i + \gamma w_i + k'(W + m + q_{ij})^{\frac{3}{2}} t_{ij} \leq M_{ij}(1 - x_{ij}) + f_j \quad \forall i \in N', j \in N^-, \tag{A.7}$$

where constraints (A.6) establish the energy relationship between the starting depot 0 and customer i, and constraints (A.7) are the energy relationship between customer i and node j (which can be a customer node or the ending depot $n + 1$). M_{ij} take the same values as before. The objective function becomes

$$\min \sum_{(i,j) \in A} (c_{ij} x_{ij} + \delta e_{ij}) + \sum_{i \in I} \delta \gamma w_i. \tag{A.8}$$

Then, our solution schemes can be directly applied for this extension.

Appendix B. Instance Generation Procedures

This section presents the detailed procedures for instance generation.

Appendix B.1. New Benchmark Instances (Set A)

In this set, we consider two types of instances and each has 10–50 customers. For type 1 instances, named *Set* A_1, the depots are located at the lower left corner of the region. For type 2 instances, named *Set* A_2, the depots are in the middle of the region. For a fixed number of customers in each type, we generate 5 instances. Our instances are labeled as $Set\_A_x\_Cust\_Y\_Z$, which represents that this is the Zth instance of Y customers in Set A_x.

Based on the size of drones, we consider the delivery of relatively lightweight items (including those like medicines). The demand of the first 40% of customers is drawn uniformly from $[0.1, 0.7]$ and the demand of the remaining customers is drawn uniformly from $[0.1, 1.5]$. We set $K = \lceil \frac{\sum_{i \in N'} d_i}{3Q} \rceil$, that is, we expect that each drone can perform 3 or more trips on average. For Set A_1, the coordinate of the depot is (0, 0). The $x-$coordinate and $y-$coordinate of each customer is drawn uniformly from $[0, 480]$. Since we assume travel distance and travel time are the same, if a customer is located at (0, 480), then the travel time from the depot to this customer would be 480 seconds. Meanwhile, we let $c_{ij} = t_{ij}\ \forall (i, j) \in A$. For the depots, we set $a_0 = a_{n+1} = 0$ and generate the right-hand side of the time window as follows: We first compute the travel time between the depot and each customer, i.e., t_{0j}, and rank them in a non-increasing order; we then sum up the first hth numbers in order, where $h = \lceil \frac{\lfloor N' \rfloor}{K} \rceil$ and the sum is denoted as s. Finally, we set $b_0 = b_{n+1} = \lceil 2s \rceil$. This generation scheme is based on the idea that, in an extreme situation, each drone trip only involves one customer and each drone performs at most h trips. And all the deliveries can be finished within $\lceil 2s \rceil$ time limit. As travel time satisfies triangle inequality, the earliest time that a customer j can be serviced is t_{0j}, and the latest time that a drone must leave j is $b_{n+1} - t_{j,n+1}$. To create customers' time windows, we refer to the method in Solomon (1987). We first randomly generate the center of the time window o_j from $[t_{0j}, b_{n+1} - t_{j,n+1}]$ using uniform distribution, then we generate the time window's width w_j as a normally distributed random number whose mean is $0.25(b_{n+1} - t_{j,n+1} - t_{0j})$ and standard deviation is $0.05(b_{n+1} - t_{j,n+1} - t_{0j})$. We set $a_j = \max(\lceil t_{0j} \rceil, \lfloor o_j - 0.5w_j \rfloor)$, $b_j = \min\{\lfloor b_{n+1} - t_{j,n+1} \rfloor, \lfloor o_j + 0.5w_j \rfloor\}$. For Set A_2, the coordinate of the depot is (480, 480). The x-coordinate and y-coordinate of each customer is drawn uniformly from $[0, 960]$. The method of generating the time windows is the same as that of Set A_1.

Appendix B.2. Instances Extended From Solomon's Instances (Set B)

We generate this set of instances based on the principle of minimal modifications to the original data. To fit Solomon's instances, we need to add a service time $s_i, \forall i \in N'$ to constraints (14) and (16) when conducting our numerical tests. We also make some modifications to customers' demands to fit the drone's payload and to allow multi-trip operations. In particular, for type C2 and RC2 instances with the first 25 and 40 customers, demands are multiplied by 0.03, because the minimal and the maximal demands are 10 and 40, respectively. For type R2 instances, demands are multiplied by 0.05 for those with the first 25 customers, because the minimal and the maximal demand are 2 and 29, respectively; demands are multiplied by 0.045 for those with the first 40 customers, because the maximal demand now becomes 31. We determine the number of drones as described in the former section.

Appendix C. Detailed Results

This section provides the detailed results of our numerical tests in Section 5, which are also available at `https://sites.google.com/view/chengchun/instances`.

Table C.11: Results for Set A instances with 10–30 customers

			R_e										E_e										$(R+E)_e$									
	Cust	Inst	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A_1	10	1	2930.4	2930.4	0.0	0.7	2816.4	0	130	17	2	5	203.3	203.3	0.0	0.8	187.0	0	152	21	2	5	3133.6	3133.6	0.0	0.9	3011.8	0	200	21	2	5
		2	4426.1	4426.1	0.0	0.1	4337.4	1	24	17	2	6	314.6	314.6	0.0	0.4	313.1	0	96	25	2	6	4740.7	4740.7	0.0	0.2	4695.6	0	80	16	2	6
		3	4252.3	4252.3	0.0	0.2	4205.5	0	68	13	2	5	305.5	305.5	0.0	0.5	303.9	0	125	59	2	5	4557.8	4557.8	0.0	0.5	4555.4	0	77	43	2	5
		4	4105.3	4105.3	0.0	1.2	3692.5	0	148	44	2	5	287.4	287.4	0.0	1.9	252.6	1	179	41	2	5	4392.7	4392.7	0.0	1.3	3886.2	0	180	22	2	5
		5	4225.0	4225.0	0.0	0.2	4070.9	0	91	16	2	5	301.0	301.0	0.0	1.1	288.8	1	139	11	2	5	4526.0	4526.0	0.0	0.4	4349.3	0	137	24	2	5
	15	1	6601.6	6601.6	0.0	14.7	6286.3	3	363	38	3	7	473.1	473.1	0.0	18.0	452.2	1	472	35	3	7	7074.7	7074.2	0.0	12.8	6742.0	0	316	38	3	7
		2	4114.7	4114.7	0.0	6.2	3831.2	2	155	20	2	4	284.5	284.5	0.0	18.5	255.0	6	787	35	2	4	4399.8	4399.8	0.0	22.5	4051.3	7	550	33	2	4
		3	5575.0	5574.7	0.0	761.0	4816.1	5	686	19	2	6	387.7	387.7	0.0	318.3	291.8	1	777	32	2	6	5970.1	5969.5	0.0	1391.5	5165.0	2	761	28	2	6
		4	5125.9	5125.4	0.0	31.3	4811.3	0	202	29	2	7	363.6	363.6	0.0	22.9	319.2	0	333	39	2	7	5493.1	5492.8	0.0	40.4	5124.8	0	376	29	2	7
		5	6900.0	6899.4	0.0	70.0	6210.1	7	597	32	2	7	484.3	484.3	0.0	35.6	402.7	5	608	25	2	8	7385.9	7385.5	0.0	63.8	6655.9	5	590	35	2	7
	20	1	7720.1	7720.1	0.0	2.3	7638.3	0	118	38	4	11	566.0	566.0	0.0	6.2	552.7	0	289	44	4	11	8287.7	8287.7	0.0	2.5	8247.6	0	384	44	4	11
		2	8912.6	8912.3	0.0	44.1	8574.7	2	404	33	4	10	638.3	638.2	0.0	35.4	554.8	1	678	36	4	10	9550.9	9550.8	0.0	75.7	9105.8	1	775	29	4	10
		3	8219.7	8219.7	0.0	18.2	8112.8	0	162	45	4	10	597.0	596.9	0.0	93.6	525.3	1	540	34	4	10	8819.3	8819.3	0.0	24.7	8636.7	0	541	37	4	10
		4	6229.5	6229.0	0.0	29.3	6073.8	0	454	34	3	9	459.2	459.2	0.0	141.4	379.5	2	479	35	3	9	6697.0	6696.5	0.0	34.0	6455.1	0	460	36	3	9
		5	7269.4	7268.7	0.0	717.3	6796.6	6	611	27	3	9	505.5	505.4	0.0	84.9	481.9	4	805	48	3	9	7785.3	7784.5	0.0	983.3	7207.8	2	918	32	3	9
	25	1	9962.5	9961.5	0.0	59.7	9814.6	0	136	37	4	12	712.3	712.2	0.0	76.9	636.3	1	891	33	4	12	10683.5	10682.5	0.0	217.0	10436.5	5	1454	31	4	12
		2	8064.6	8063.8	0.0	7817.4	7611.9	10	1002	22	3	10	575.1	575.1	0.0	10468.2	496.1	14	1495	50	3	10	8639.7	8638.9	0.0	8600.1	8105.8	13	2458	33	3	10
		3	9422.6	9421.7	0.0	609.4	9182.2	0	263	28	4	10	675.2	675.2	0.0	1244.2	562.4	5	1741	31	4	10	10097.8	10096.8	0.0	1186.0	9774.8	0	666	26	4	10
		4	9460.6	9459.8	0.0	443.8	9135.6	3	509	45	4	13	689.2	689.2	0.0	736.1	581.5	3	854	38	4	13	10149.8	10148.8	0.0	324.8	9702.3	2	1199	58	4	13
		5	10398.1	10213.6	1.8	14400.0	9786.8	2	852	39	4	11	747.8	747.7	0.0	10017.9	614.9	1	1429	47	4	12	11209.3	10899.6	2.8	14400.0	10428.0	2	1783	43	4	12
	30	1	9165.8	9164.9	0.0	2440.8	8841.7	7	1922	42	5	13	667.9	667.8	0.0	3225.0	554.2	3	1804	48	5	13	9836.3	9835.3	0.0	8584.4	9419.8	3	1947	46	5	13
		2	11811.4	11810.6	0.0	360.6	11564.9	1	484	40	5	14	858.0	857.9	0.0	942.4	734.0	0	1113	39	5	14	12669.3	12668.1	0.0	1732.8	12361.7	1	1476	45	5	14
		3	11530.4	11289.6	2.1	14400.0	10932.4	2	1243	34	5	14	831.8	807.8	2.9	14400.0	706.7	2	2226	33	5	14	12435.7	12077.9	2.9	14400.0	11681.0	2	1984	35	5	13
		4	11670.9	11669.7	0.0	2699.8	11281.7	0	949	45	5	13	845.5	845.4	0.0	11981.8	755.7	1	1829	55	5	13	12517.1	12515.9	0.0	5211.6	12073.9	0	1828	49	5	13
		5	11330.9	11222.6	1.0	14400.0	10924.6	9	1266	51	5	13	814.7	798.3	2.0	14400.0	729.8	11	2451	64	5	13	12090.4	12020.0	0.6	14400.0	11644.5	8	1560	60	5	13
Set A_2	10	1	4668.6	4668.6	0.0	0.2	4475.2	0	62	9	2	5	324.2	324.2	0.0	0.5	292.5	2	174	15	2	6	5000.2	5000.2	0.0	0.2	4772.6	0	75	2	2	5
		2	5447.2	5447.2	0.0	0.1	5385.9	0	44	2	2	6	379.9	379.9	0.0	0.1	378.7	0	128	8	2	6	5827.1	5827.1	0.0	0.1	5825.6	0	96	2	2	6
		3	4946.7	4946.7	0.0	0.4	4910.0	0	48	30	2	5	324.9	324.9	0.0	0.2	323.6	0	77	12	2	5	5271.5	5271.5	0.0	0.2	5267.8	0	45	15	2	5
		4	5775.2	5775.2	0.0	0.9	5046.2	0	26	18	2	5	382.9	382.9	0.0	1.3	311.9	0	222	20	2	5	6158.6	6158.6	0.0	0.8	5401.1	0	109	15	2	5
		5	5178.5	5178.5	0.0	0.6	4785.7	0	24	24	2	6	356.9	356.9	0.0	0.4	309.0	0	112	14	2	6	5535.3	5535.3	0.0	0.3	5112.1	0	32	15	2	6
	15	1	6427.6	6427.6	0.0	1.9	6295.3	0	88	44	3	8	444.2	444.2	0.0	1.2	428.6	0	140	25	3	8	6871.8	6871.8	0.0	1.0	6868.4	0	211	22	3	8
		2	7995.4	7995.4	0.0	10.9	7072.5	3	337	41	3	8	534.0	534.0	0.0	7.6	446.4	1	295	33	3	10	8537.6	8537.6	0.0	10.1	7542.8	2	351	43	3	8
		3	6173.9	6173.9	0.0	0.7	6165.3	2	40	33	3	9	436.3	436.3	0.0	3.1	398.6	0	246	29	3	9	6614.5	6614.5	0.0	1.5	6465.5	1	240	28	3	9
		4	8238.9	8238.9	0.0	5.5	7249.5	6	180	14	2	8	541.3	541.3	0.0	6.5	463.7	4	345	19	2	8	8780.2	8780.0	0.0	10.0	7878.0	5	322	25	2	8
		5	8114.1	8114.1	0.0	6.0	7565.1	1	116	38	3	9	544.0	544.0	0.0	2.1	508.1	2	346	32	3	10	8674.8	8674.5	0.0	5.8	7978.3	1	365	29	3	9
	20	1	10684.6	10684.6	0.0	46.5	9677.3	5	243	28	4	13	741.3	741.3	0.0	57.1	643.6	1	666	31	4	13	11425.9	11425.0	0.0	43.0	10524.4	0	323	34	4	13
		2	9093.2	9093.2	0.0	20.0	8408.8	1	181	26	3	10	631.2	631.2	0.0	36.4	540.8	2	428	42	3	10	9733.1	9733.1	0.0	27.1	9045.0	0	392	41	3	10
		3	9444.0	9444.0	0.0	14.5	9046.6	2	240	43	3	11	652.9	652.9	0.0	21.1	577.3	2	342	41	3	11	10096.9	10096.9	0.0	15.4	9590.1	2	313	34	3	11
		4	8858.5	8858.5	0.0	12.6	8573.3	0	244	42	4	10	619.5	619.5	0.0	4.6	595.0	1	438	44	4	11	9495.5	9495.5	0.0	22.0	9136.3	2	712	42	4	10
		5	7750.7	7750.4	0.0	41.6	7341.9	2	387	34	3	9	551.4	551.4	0.0	67.2	463.9	2	564	42	3	9	8302.2	8302.0	0.0	57.9	7805.1	3	764	31	3	9
	25	1	10667.2	10666.2	0.0	182.1	10044.2	1	394	32	4	12	752.2	752.1	0.0	570.9	646.2	5	1216	30	4	13	11440.1	11439.1	0.0	229.7	10733.9	2	408	50	4	12
		2	11623.1	11622.4	0.0	78.7	11094.8	0	52	35	4	13	805.3	805.2	0.0	186.4	707.5	6	1110	27	4	13	12430.0	12429.0	0.0	159.0	11747.4	4	798	34	4	13
		3	10243.9	10243.0	0.0	609.9	9584.1	5	769	39	4	12	717.8	717.7	0.0	668.6	601.7	3	1100	39	4	12	10977.3	10976.2	0.0	971.0	10216.3	3	1468	43	4	12
		4	11468.4	11468.4	0.0	44.4	11015.6	3	333	46	4	13	805.7	805.7	0.0	24.4	778.8	1	544	56	4	14	12279.3	12279.2	0.0	66.5	11705.7	2	736	39	4	13
		5	10986.4	10985.3	0.0	96.7	10415.8	2	799	31	4	13	784.1	784.0	0.0	54.8	668.3	5	635	34	4	14	11791.6	11790.5	0.0	139.2	11101.9	1	855	31	4	13
	30	1	14004.6	14003.2	0.0	6400.9	13082.9	4	525	43	5	16	983.4	983.3	0.0	1125.6	852.0	5	1232	39	5	17	15002.0	15000.5	0.0	5548.2	13939.4	3	1674	40	5	16
		2	11958.2	11957.0	0.0	5707.0	11162.0	0	572	49	4	14	840.2	814.1	3.1	14400.0	701.9	3	1256	40	4	14	12798.4	12797.1	0.0	10161.1	11913.3	2	573	55	4	14
		3	11432.8	11431.8	0.0	6124.3	10552.0	5	1332	33	5	14	802.6	802.5	0.0	12470.0	660.5	7	2277	33	5	14	12239.0	12237.8	0.0	7670.6	11183.5	4	1767	29	5	14
		4	10827.6	10826.6	0.0	373.0	10491.1	5	1174	38	4	13	763.6	763.5	0.0	7057.4	645.4	6	1438	30	4	13	11591.2	11590.0	0.0	596.6	11128.8	6	1788	35	4	13
		5	12382.2	12380.9	0.0	13527.0	11561.0	2	1033	43	4	13	878.7	855.7	2.6	14400.0	739.8	4	2876	35	4	14	13323.6	12999.4	2.4	14400.0	12321.1	5	1385	39	4	13

Trip: the number of trips performed by UAVs.

Table C.12: Results for linear approximation models

			R_a											$(R+E)_a$										
	Cust	Inst	UP	LB	Gap	CPU	RLB	SECs	UAVs	Trip	Viol▲	Linear energy	Nonlinear energy	UP	LB	Gap	CPU	RLB	SECs	UAVs	Trip	Viol	Linear energy	Nonlinear energy
Set A_1	10	1	2930.4	2930.4	0.0	0.8	2791.5	12	2	5	F▲	0.5204	0.5656	3117.4	3117.4	0.0	1.0	2996.2	23	2	5	F	0.5196	0.5645
		2	4426.1	4426.1	0.0	0.1	4385.3	10	2	6	F	0.7996	0.8740	4713.9	4713.9	0.0	0.1	4675.2	9	2	6	F	0.7996	0.8740
		3	4252.3	4252.3	0.0	0.2	4202.1	15	2	5	F	0.7789	0.8537	4531.3	4531.3	0.0	0.4	4528.0	20	2	5	F	0.7751	0.8486
		4	3948.8	3948.8	0.0	0.7	3655.7	16	2	4	T▲	0.7475	0.8268	4218.0	4218.0	0.0	1.3	3891.4	34	2	4	T	0.7475	0.8268
		5	4225.0	4225.0	0.0	0.3	4058.0	16	2	5	F	0.7638	0.8363	4499.9	4499.9	0.0	0.4	4368.2	19	2	5	F	0.7638	0.8363
	15	1	6529.0	6528.9	0.0	9.1	6276.7	22	3	7	T	1.2070	1.3284	6960.3	6960.3	0.0	10.4	6714.9	31	3	7	T	1.1981	1.3163
		2	4100.2	4100.2	0.0	9.6	3791.5	34	2	4	T	0.7629	0.8389	4373.4	4373.0	0.0	9.8	4018.0	22	2	4	T	0.7572	0.8314
		3	5315.6	5315.6	0.0	191.7	4861.4	31	2	5	T	0.9765	1.0718	5664.9	5664.4	0.0	196.8	5244.0	29	2	5	T	0.9705	1.0637
		4	5125.9	5125.9	0.0	66.0	4837.1	38	2	7	F	0.9368	1.0273	5461.2	5461.2	0.0	38.0	5145.9	30	2	7	F	0.9313	1.0199
		5	6722.6	6722.6	0.0	12.4	6196.3	25	2	6	T	1.2568	1.3845	7165.7	7165.5	0.0	17.1	6635.4	29	2	7	T	1.2131	1.3246
	20	1	7720.1	7720.1	0.0	1.6	7675.0	42	4	11	F	1.4327	1.5766	8235.7	8235.7	0.0	1.7	8178.0	37	4	11	F	1.4327	1.5766
		2	8773.9	8773.9	0.0	20.4	8546.9	29	4	10	T	1.6400	1.8091	9357.1	9357.1	0.0	26.8	9140.0	30	4	10	T	1.6202	1.7824
		3	8219.7	8219.7	0.0	24.6	8111.5	44	4	10	F	1.5268	1.6814	8765.1	8765.1	0.0	9.9	8650.2	42	4	10	F	1.5148	1.6653
		4	6229.5	6229.4	0.0	48.8	6075.1	33	3	9	F	1.1762	1.2989	6652.9	6652.8	0.0	25.6	6495.1	40	3	9	F	1.1762	1.2989
		5	7174.4	7173.7	0.0	159.8	6785.5	27	3	9	T	1.3128	1.4391	7640.7	7640.3	0.0	240.6	7257.9	45	3	9	T	1.2953	1.4156
	25	1	9962.5	9961.6	0.0	55.6	9799.6	31	4	12	F	1.8249	2.0027	10619.5	10619.5	0.0	63.3	10460.3	51	4	12	F	1.8249	2.0027
		2	7818.1	7817.4	0.0	71.8	7648.2	26	3	10	T	1.4362	1.5779	8330.7	8330.7	0.0	56.1	8165.8	35	3	10	T	1.4240	1.5615
		3	9422.6	9421.7	0.0	581.3	9191.1	41	4	10	F	1.7205	1.8866	10039.0	10039.0	0.0	522.4	9800.3	30	4	10	F	1.7123	1.8756
		4	9460.6	9459.6	0.0	321.9	9120.1	42	4	13	F	1.7414	1.9145	10087.5	10086.5	0.0	266.6	9738.7	42	4	13	F	1.7414	1.9145
		5	10398.1	10397.1	0.0	14287.6	9814.1	50	4	11	T	1.9580	2.1631	11097.5	11096.4	0.0	12119.9	10470.7	47	4	11	F	1.9427	2.1425
	30	1	9091.6	9091.0	0.0	544.5	8845.7	38	5	13	T	1.6714	1.8374	9691.7	9691.3	0.0	330.1	9479.7	54	5	13	T	1.6668	1.8311
		2	11811.4	11810.3	0.0	356.3	11601.2	40	5	14	F	2.1760	2.3937	12592.0	12591.0	0.0	403.7	12360.8	31	5	14	F	2.1682	2.3834
		3	11543.0	11358.3	1.6	14400.0	10938.3	45	5	14	T	2.1197	2.3291	12289.7	12288.5	0.0	11300.9	11695.5	33	5	14	F	2.1094	2.3154
		4	11670.9	11669.9	0.0	1537.7	11359.6	37	5	13	F	2.1578	2.3745	12441.4	12440.3	0.0	1834.2	12029.1	38	5	13	F	2.1395	2.3498
		5	11221.9	11220.7	0.0	2553.1	10897.0	39	5	13	T	2.0783	2.2881	11968.4	11967.3	0.0	1573.8	11683.8	56	5	13	T	2.0724	2.2805
Set A_2	10	1	4668.6	4668.6	0.0	0.2	4475.2	4	2	5	F	0.8428	0.9212	4972.0	4972.0	0.0	0.2	4770.1	7	2	5	F	0.8428	0.9212
		2	5447.2	5447.2	0.0	0.1	5447.2	1	2	6	F	0.9694	1.0552	5796.1	5796.1	0.0	0.1	5773.8	1	2	6	F	0.9694	1.0552
		3	4946.7	4946.7	0.0	0.3	4842.9	18	2	5	F	0.8483	0.9136	5249.1	5249.1	0.0	0.2	5155.9	7	2	5	F	0.8400	0.9024
		4	5775.2	5775.2	0.0	1.3	5053.4	27	2	5	F	0.9934	1.0714	6131.1	6131.1	0.0	0.8	5384.6	25	2	5	F	0.9888	1.0651
		5	5178.5	5178.5	0.0	0.3	4711.4	17	2	6	F	0.9139	0.9913	5507.5	5507.5	0.0	0.4	5073.5	31	2	6	F	0.9139	0.9913
	15	1	6427.6	6427.6	0.0	1.1	6424.2	47	3	8	F	1.1361	1.2340	6836.6	6836.6	0.0	0.7	6836.6	35	3	8	F	1.1361	1.2340
		2	7762.6	7762.6	0.0	7.4	7038.7	40	3	9	T	1.3402	1.4462	8244.7	8244.7	0.0	4.5	7679.7	31	3	9	T	1.3390	1.4447
		3	6173.9	6173.9	0.0	0.9	6132.3	61	3	9	F	1.1361	1.2453	6577.1	6577.1	0.0	0.7	6415.8	28	3	9	F	1.1202	1.2239
		4	8238.9	8238.9	0.0	8.0	7277.4	17	2	8	F	1.4012	1.5068	8742.4	8742.4	0.0	5.0	7898.3	23	2	8	F	1.3987	1.5038
		5	8114.1	8114.1	0.0	6.4	7525.2	36	3	9	F	1.4349	1.5575	8630.6	8630.6	0.0	7.1	7957.6	26	3	9	F	1.4349	1.5575
	20	1	10684.6	10683.7	0.0	34.4	9689.7	41	4	13	F	1.8992	2.0692	11365.6	11365.1	0.0	22.2	10454.5	50	4	13	F	1.8916	2.0590
		2	9093.2	9092.7	0.0	33.7	8492.2	37	3	10	F	1.6296	1.7775	9679.9	9679.0	0.0	31.7	8901.2	31	3	10	F	1.6296	1.7775
		3	9444.0	9444.0	0.0	23.1	9167.5	35	3	11	F	1.6753	1.8212	10045.1	10045.1	0.0	17.4	9728.2	41	3	11	F	1.6696	1.8137
		4	8858.5	8858.5	0.0	14.5	8539.0	37	4	10	F	1.6307	1.7899	9440.1	9440.1	0.0	8.1	9056.6	30	4	10	F	1.6154	1.7692
		5	7708.5	7708.0	0.0	24.8	7323.1	25	3	9	T	1.4241	1.5645	8216.4	8216.4	0.0	24.3	7832.3	32	3	9	T	1.4109	1.5466
	25	1	10667.2	10666.4	0.0	289.0	10062.3	40	4	12	F	1.9581	2.1469	11366.1	11365.4	0.0	161.4	10735.0	32	4	12	T	1.9338	2.1138
		2	11623.1	11623.1	0.0	136.7	11028.5	36	4	13	F	2.0624	2.2431	12365.5	12365.5	0.0	76.5	11793.3	42	4	13	F	2.0624	2.2431
		3	9984.1	9984.1	0.0	83.3	9571.5	37	4	12	T	1.8217	1.9937	10634.6	10634.6	0.0	39.5	10242.6	41	4	12	T	1.8068	1.9737
		4	11468.4	11468.4	0.0	58.2	10998.9	46	4	13	T	2.1005	2.3022	12211.3	12211.3	0.0	72.0	11725.9	40	4	13	F	2.0637	2.2527
		5	10854.2	10853.4	0.0	50.8	10465.3	28	4	14	T	2.0058	2.2046	11571.2	11571.2	0.0	29.0	11106.6	34	4	14	T	1.9916	2.1856
	30	1	13998.6	13997.3	0.0	1292.2	13068.9	42	5	16	T	2.5527	2.7955	14909.4	14908.0	0.0	1466.8	13960.3	39	5	16	T	2.5299	2.7653
		2	11958.2	11957.2	0.0	2828.7	11183.9	43	4	14	F	2.1636	2.3643	12728.9	12727.7	0.0	2617.0	11921.9	41	4	14	F	2.1410	2.3338
		3	11144.0	11144.0	0.0	1012.5	10558.9	29	5	13	T	2.0439	2.2413	11878.9	11878.2	0.0	1065.3	11232.4	32	5	13	T	2.0414	2.2380
		4	10827.6	10826.6	0.0	291.7	10481.7	38	4	13	F	1.9436	2.1211	11527.3	11526.5	0.0	179.7	11138.1	28	4	13	F	1.9435	2.1209
		5	12244.9	12243.8	0.0	4124.7	11640.1	44	4	12	T	2.2363	2.4497	13050.0	13048.7	0.0	2316.2	12319.9	37	4	12	T	2.2363	2.4497

▲: Whether there exist trips which violate the energy capacity constraint. If yes, a 'T' is used, 'F' otherwise.

Table C.13: Results for Set A instances with tighter time windows

			R_e										E_e										$(R+E)_e$									
	Cust	Inst	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A$_1$	10	1	3135.3	3135.3	0.0	0.2	2952.4	0	72	17	2	5	212.7	212.7	0.0	0.6	191.0	0	120	33	2	5	3351.5	3351.5	0.0	0.5	3169.0	0	152	24	2	5
		2	5499.8	5499.8	0.0	0.2	4679.7	1	26	12	2	7	364.6	364.6	0.0	0.2	317.7	1	103	8	2	7	5864.5	5864.5	0.0	0.4	4989.3	0	51	11	2	7
		3	4175.2	4175.2	0.0	0.3	4118.3	0	68	19	2	5	304.2	304.2	0.0	0.2	291.2	0	77	13	2	5	4479.4	4479.4	0.0	0.2	4425.9	0	77	23	2	5
		4	4315.4	4315.4	0.0	0.4	3956.9	0	36	34	2	5	296.6	296.6	0.0	0.6	291.5	1	120	55	2	5	4615.9	4615.9	0.0	0.3	4155.8	1	210	14	2	5
		5	4184.3	4184.3	0.0	0.1	4177.9	0	18	6	2	5	300.8	300.8	0.0	0.1	298.6	0	128	4	2	5	4486.2	4486.2	0.0	0.1	4433.1	1	45	4	2	5
	15	1	6636.2	6636.2	0.0	6.9	6335.3	3	393	30	3	7	469.8	469.8	0.0	5.2	426.1	2	540	26	3	7	7106.1	7105.8	0.0	7.1	6756.4	2	291	42	3	7
		2	3994.4	3994.4	0.0	1.8	3777.1	3	215	22	2	4	292.1	292.1	0.0	9.7	262.1	6	397	23	2	4	4286.5	4286.5	0.0	1.5	4001.6	0	120	20	2	4
		3	5583.5	5583.0	0.0	109.7	4955.7	3	353	30	2	6	398.3	398.2	0.0	186.7	298.6	2	758	35	2	6	5981.9	5981.6	0.0	146.9	5292.6	2	441	32	2	6
		4	5237.8	5237.6	0.0	13.1	4816.9	0	230	22	2	7	383.1	383.0	0.0	25.4	303.3	0	508	35	2	7	5620.9	5620.7	0.0	13.7	5248.2	0	485	44	2	7
		5	6956.9	6956.5	0.0	29.5	6248.5	4	300	29	2	7	481.7	481.7	0.0	11.9	402.9	4	361	35	2	8	7467.7	7467.0	0.0	39.5	6643.6	4	446	30	2	7
	20	1	7729.5	7729.5	0.0	0.7	7717.7	0	42	24	4	11	559.0	559.0	0.0	3.5	536.5	0	219	36	4	11	8288.5	8288.5	0.0	0.8	8277.3	0	281	27	4	11
		2	9253.7	9252.9	0.0	62.6	8741.4	0	565	37	4	10	666.4	666.3	0.0	333.0	549.8	2	744	23	4	11	9932.2	9931.4	0.0	175.3	9308.3	2	707	37	4	10
		3	8363.5	8363.5	0.0	16.7	8166.9	0	230	33	4	10	603.7	603.7	0.0	27.4	533.7	2	552	42	4	10	8976.2	8976.2	0.0	11.6	8726.1	0	240	47	4	10
		4	6429.8	6429.4	0.0	199.5	6049.3	1	728	20	3	9	446.2	446.1	0.0	68.8	377.3	0	810	28	3	9	6876.0	6876.0	0.0	116.4	6433.9	0	725	24	3	9
		5	7284.8	7284.1	0.0	131.1	6868.7	3	404	31	3	9	506.4	506.4	0.0	101.5	424.3	2	855	30	3	9	7791.2	7790.6	0.0	134.8	7304.7	4	646	36	3	9
	25	1	9992.0	9991.1	0.0	44.1	9824.5	4	465	25	4	12	714.5	714.4	0.0	37.5	690.5	3	603	30	4	12	10714.3	10713.7	0.0	49.7	10470.6	1	226	38	4	12
		2	8311.8	8310.9	0.0	2566.6	7616.0	6	1276	28	3	10	601.9	568.2	5.6	14400.0	460.0	11	1328	31	3	10	8918.8	8917.9	0.0	2700.7	8110.6	5	1206	53	3	10
		3	9644.4	9643.4	0.0	1034.5	9246.8	5	797	27	4	10	695.2	695.2	0.0	1121.3	566.7	1	979	27	4	10	10339.6	10338.6	0.0	1684.2	9840.3	1	816	33	4	10
		4	9699.3	9698.3	0.0	2054.5	9129.2	5	1157	49	4	13	698.5	698.5	0.0	1636.5	591.9	4	1606	45	4	13	10397.8	10396.8	0.0	1866.6	9732.4	2	1638	49	4	13
		5	10502.7	10501.6	0.0	6189.6	9848.1	3	1075	49	4	12	744.0	726.6	2.3	14400.0	614.2	4	1969	39	4	12	11255.9	11254.8	0.0	9598.6	10455.0	2	1113	45	4	12
	30	1	9314.1	9313.2	0.0	1912.9	8936.0	1	1684	40	5	13	680.2	680.1	0.0	8464.1	603.5	1	2688	44	5	13	9996.3	9995.3	0.0	1503.5	9534.8	1	1496	44	5	13
		2	12099.3	12098.1	0.0	1022.8	11784.0	1	1145	38	5	14	871.2	871.1	0.0	3270.8	755.2	0	2205	39	5	14	12970.5	12969.3	0.0	710.3	12559.7	0	1835	39	5	14
		3	11560.9	11403.8	1.4	14400.0	10922.5	1	697	38	5	14	831.8	803.8	3.4	14400.0	691.7	1	1588	33	5	14	12452.0	12225.3	1.8	14400.0	11667.0	3	1419	28	5	14
		4	11679.0	11677.9	0.0	1092.6	11337.4	2	1191	47	5	13	852.2	852.1	0.0	2229.4	710.3	0	1756	37	5	13	12533.7	12532.5	0.0	1113.3	12060.5	0	2149	40	5	13
		5	11508.9	11330.3	1.6	14400.0	11019.1	9	2249	56	5	13	824.8	808.4	2.0	14400.0	734.4	6	1863	57	5	13	12333.8	12123.0	1.7	14400.0	11687.5	2	2029	45	5	13
Set A$_2$	10	1	5287.2	5287.2	0.0	0.1	5239.2	0	20	10	2	6	355.7	355.7	0.0	0.4	355.2	0	32	25	2	6	5642.9	5642.9	0.0	0.2	5527.3	0	32	5	2	6
		2	6180.3	6180.3	0.0	0.1	6161.0	1	21	9	2	7	415.1	415.1	0.0	0.1	415.1	1	51	8	2	7	6595.4	6595.4	0.0	0.1	6592.6	1	51	5	2	7
		3	5675.9	5675.9	0.0	0.2	5596.2	0	20	21	2	6	358.4	358.4	0.0	0.1	354.2	0	64	5	2	6	6034.4	6034.4	0.0	0.2	5997.5	0	64	27	2	6
		4	5943.9	5943.9	0.0	0.2	5775.4	0	66	13	2	6	385.1	385.1	0.0	0.3	381.3	0	64	17	2	6	6329.0	6329.0	0.0	0.2	6133.7	0	32	15	2	6
		5	5656.0	5656.0	0.0	0.1	5473.3	1	84	10	2	7	370.1	370.1	0.0	0.2	351.7	1	84	8	2	7	6026.1	6026.1	0.0	0.3	5824.9	2	84	10	2	7
	15	1	6198.2	6198.2	0.0	0.7	6022.9	1	48	33	3	8	429.1	429.1	0.0	0.8	424.9	0	186	34	3	8	6627.3	6627.3	0.0	0.8	6420.4	0	115	21	3	8
		2	7759.7	7759.7	0.0	1.0	7415.4	0	132	32	3	8	533.1	533.1	0.0	1.1	506.5	0	166	27	3	9	8301.2	8301.2	0.0	0.9	7921.5	0	190	27	3	8
		3	6825.2	6825.2	0.0	1.0	6524.9	0	128	31	3	9	470.2	470.2	0.0	1.1	422.7	1	171	21	3	9	7295.4	7295.4	0.0	1.2	6920.2	0	178	25	3	9
		4	8627.0	8627.0	0.0	1.7	7999.3	3	148	34	2	8	565.3	565.3	0.0	2.7	499.5	3	360	21	2	9	9207.1	9207.1	0.0	2.2	8604.5	3	256	37	2	8
		5	8187.3	8187.3	0.0	0.7	8035.0	1	140	33	3	9	544.7	544.7	0.0	0.7	544.0	1	178	31	3	10	8743.9	8743.9	0.0	0.9	8599.1	0	246	44	3	9
	20	1	10542.8	10542.8	0.0	2.9	10058.5	3	378	22	4	12	732.4	732.4	0.0	2.8	672.8	2	461	34	4	13	11283.8	11282.8	0.0	2.5	10794.8	0	163	35	4	12
		2	9184.3	9184.3	0.0	15.4	8489.8	0	112	39	3	10	648.7	648.7	0.0	54.7	545.0	3	565	36	3	11	9834.2	9834.2	0.0	32.3	9072.8	2	277	35	3	10
		3	10565.5	10565.5	0.0	13.5	9821.2	4	256	23	3	11	714.0	713.9	0.0	11.4	654.4	4	380	30	3	11	11279.4	11279.4	0.0	14.6	10542.4	3	732	42	3	11
		4	9278.4	9278.4	0.0	9.1	8655.4	1	384	40	4	11	644.7	644.7	0.0	9.7	606.5	0	379	38	4	12	9925.9	9924.9	0.0	7.6	9234.8	1	314	32	4	11
		5	7929.8	7929.8	0.0	9.7	7610.9	1	614	38	3	9	562.7	562.7	0.0	17.7	492.5	1	365	29	3	10	8502.9	8502.9	0.0	9.8	8105.9	0	438	34	3	9
	25	1	10860.4	10859.6	0.0	432.8	10101.1	5	760	34	4	12	768.3	768.2	0.0	865.1	658.0	6	1445	38	4	12	11640.6	11639.5	0.0	223.9	10881.6	6	819	43	4	12
		2	11957.9	11956.9	0.0	52.5	11302.5	6	553	40	4	12	833.9	833.8	0.0	132.7	722.4	9	882	30	4	12	12794.5	12793.6	0.0	84.4	11962.1	2	374	25	4	12
		3	11106.8	11106.8	0.0	163.4	10176.9	3	225	25	4	13	754.9	754.9	0.0	158.0	648.4	3	494	37	4	13	11875.6	11874.6	0.0	262.0	10825.7	3	609	33	4	13
		4	11529.3	11529.3	0.0	16.8	11161.8	0	468	38	4	13	812.7	812.6	0.0	26.5	759.8	0	703	34	4	13	12342.3	12341.0	0.0	11.8	12037.1	0	382	54	4	13
		5	11174.2	11173.3	0.0	35.8	10622.9	0	719	34	4	14	801.5	801.5	0.0	21.2	758.0	1	718	47	4	15	11980.5	11980.5	0.0	26.9	11250.6	1	358	33	4	14
	30	1	14270.9	14269.5	0.0	1636.5	13189.4	5	741	47	5	17	1004.9	1004.8	0.0	2835.6	859.6	2	856	58	5	17	15283.9	15282.5	0.0	880.1	14064.2	2	1089	46	5	17
		2	11651.3	11650.3	0.0	132.5	11115.2	2	687	40	4	13	834.5	834.4	0.0	345.3	771.2	3	1035	58	4	14	12493.1	12493.1	0.0	130.2	11842.9	1	523	37	4	13
		3	11548.4	11547.2	0.0	697.8	10869.4	10	1074	46	5	14	818.3	818.2	0.0	2257.1	691.1	2	1742	33	5	14	12370.3	12369.2	0.0	678.8	11548.6	9	842	39	5	14
		4	11357.9	11356.8	0.0	526.9	10839.6	2	508	38	4	13	810.3	810.2	0.0	3828.6	679.4	8	1665	46	4	13	12178.5	12178.2	0.0	335.4	11488.5	4	303	35	4	13
		5	11998.2	11997.4	0.0	239.0	11581.4	3	822	34	4	13	843.1	843.0	0.0	3026.2	731.8	1	1395	32	4	13	12844.9	12844.1	0.0	513.2	12246.8	2	775	34	4	13

Table C.14: Results for Set A instances without time windows

			R_e										E_e										$(R+E)_e$									
	Cust	Inst	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A_1	10	1	2816.4	2816.4	0.0	0.4	2815.8	0	118	38	2	5	193.7	193.7	0.0	0.6	191.6	0	181	26	1	5	3011.9	3011.9	0.0	0.4	3008.9	0	158	27	2	5
		2	4426.1	4426.1	0.0	0.9	4228.5	0	36	50	2	6	310.2	310.2	0.0	0.9	284.3	0	221	14	2	6	4736.3	4736.3	0.0	0.5	4509.0	1	96	36	2	6
		3	4002.2	4002.2	0.0	0.2	4001.6	0	22	25	2	5	284.0	284.0	0.0	1.0	276.6	0	355	21	2	5	4288.9	4288.9	0.0	0.4	4288.2	0	155	34	2	5
		4	3547.3	3547.3	0.0	0.4	3491.5	0	46	14	2	4	258.0	258.0	0.0	1.9	246.9	1	288	32	2	4	3806.0	3806.0	0.0	0.5	3749.1	0	145	32	2	4
		5	4164.7	4164.7	0.0	0.6	4162.6	1	89	31	1	5	298.4	298.4	0.0	0.7	267.2	1	169	15	2	5	4463.1	4463.1	0.0	0.3	4449.2	1	76	30	1	5
	15	1	6417.5	6417.0	0.0	30.3	6282.6	2	260	36	3	7	458.8	458.8	0.0	20.0	440.2	2	621	39	3	7	6877.1	6876.4	0.0	35.0	6649.8	4	619	39	3	7
		2	3805.2	3805.2	0.0	1.1	3803.2	3	138	41	2	4	268.8	268.8	0.0	69.6	243.7	6	751	33	2	4	4076.3	4076.0	0.0	55.8	3730.7	4	558	32	2	4
		3	4925.5	4925.5	0.0	15.5	4839.1	1	534	33	2	5	356.2	356.1	0.0	31.4	337.8	3	999	34	2	5	5284.2	5283.7	0.0	13.6	5184.9	2	347	37	2	5
		4	5106.7	5106.2	0.0	2578.8	4707.2	0	785	29	2	7	358.0	358.0	0.0	70.4	326.7	0	829	25	2	7	5464.7	5464.3	0.0	521.1	5121.0	0	1025	41	2	7
		5	6153.8	6153.8	0.0	11.4	6059.8	7	288	32	2	6	444.2	444.1	0.0	31.5	413.5	14	655	45	2	6	6598.5	6598.0	0.0	22.1	6455.3	11	753	30	2	6
	20	1	7652.9	7652.6	0.0	23.2	7470.0	0	246	39	4	11	547.9	547.9	0.0	22.4	506.9	0	567	30	4	11	8201.7	8201.7	0.0	22.5	8001.3	0	731	33	4	11
		2	8773.9	8773.1	0.0	349.4	8389.1	1	612	24	4	10	628.6	628.5	0.0	214.4	551.2	0	1162	22	4	10	9407.2	9406.3	0.0	154.1	8973.3	1	1163	35	4	10
		3	8113.2	8113.2	0.0	11.5	7990.0	0	478	42	4	10	583.9	583.8	0.0	38.9	562.8	1	1165	50	4	10	8698.5	8698.1	0.0	12.6	8604.2	0	283	48	4	10
		4	6177.6	6177.0	0.0	636.0	5864.3	1	939	38	3	9	430.6	430.5	0.0	470.3	381.8	0	1304	32	3	9	6608.2	6607.7	0.0	307.5	6346.0	0	849	41	3	9
		5	6909.9	6909.3	0.0	915.4	6589.7	14	878	34	3	9	485.5	485.4	0.0	257.5	464.4	7	1864	40	3	9	7403.0	7402.3	0.0	941.0	7054.8	6	675	30	3	9
	25	1	9807.5	9806.6	0.0	642.7	9631.9	0	762	31	3	12	703.8	703.8	0.0	791.1	660.2	2	1463	48	4	12	10512.2	10511.1	0.0	989.1	10277.1	1	1808	35	4	12
		2	7810.7	7809.9	0.0	11970.3	7505.2	10	1361	35	3	10	558.3	558.3	0.0	11311.1	468.1	15	2533	46	3	10	8372.8	8372.0	0.0	8181.0	8039.9	11	1950	48	3	10
		3	9228.6	9227.7	0.0	423.5	9084.6	1	784	28	4	10	658.2	658.2	0.0	459.7	574.6	3	1687	27	4	10	9888.3	9887.3	0.0	837.8	9731.1	3	1117	32	4	10
		4	9411.5	9410.5	0.0	2303.5	8975.4	2	780	45	3	13	674.2	674.2	0.0	1246.0	600.1	1	1034	29	4	13	10086.9	10085.9	0.0	899.9	9576.4	3	1418	56	4	13
		5	10383.3	10016.5	3.5	14400.0	9704.5	1	1291	47	4	12	734.5	719.2	2.1	14400.0	646.7	3	1604	36	3	11	11123.4	10673.9	4.0	14400.0	10366.7	0	1491	43	4	12
	30	1	9014.0	9013.1	0.0	4096.4	8618.5	0	1071	37	5	13	653.8	635.2	2.9	14400.0	576.1	0	3209	41	5	13	9669.6	9668.7	0.0	4740.8	9140.5	0	2514	41	5	13
		2	11779.5	11758.9	0.2	14400.5	11479.2	0	1299	44	4	14	856.1	845.6	1.2	14400.0	723.2	3	2242	24	5	14	12635.6	12554.2	0.6	14400.0	12216.2	1	2799	37	4	14
		3	11199.9	11114.0	0.8	14400.0	10742.4	1	631	42	5	14	804.5	804.4	0.0	8639.3	747.0	0	1638	43	5	14	12005.2	11882.8	1.0	14400.0	11375.2	0	1061	29	5	14
		4	11606.9	11584.3	0.2	14400.1	11258.2	0	1360	38	5	13	831.9	831.9	0.0	9215.6	740.0	0	1860	36	5	13	12444.3	12414.6	0.2	14400.0	11934.3	0	1680	37	5	13
		5	11249.0	11077.3	1.5	14400.0	10710.9	5	1415	35	5	13	799.6	782.0	2.2	14400.0	705.5	11	1884	34	5	13	12022.3	11798.3	1.9	14400.0	11406.5	5	2763	35	5	13
Set A_2	10	1	4475.2	4475.2	0.0	0.1	4464.8	1	82	6	2	5	307.2	307.2	0.0	0.2	306.8	1	80	7	2	5	4782.4	4782.4	0.0	0.1	4780.3	2	78	9	2	5
		2	5447.2	5447.2	0.0	0.4	5438.8	1	54	20	2	6	375.1	375.1	0.0	0.4	372.0	1	162	31	2	6	5822.3	5822.3	0.0	0.2	5810.7	2	96	18	2	6
		3	4946.7	4946.7	0.0	3.9	4582.0	1	208	23	2	5	318.7	318.7	0.0	2.1	290.8	1	151	23	2	5	5265.4	5265.4	0.0	2.9	4866.4	1	195	14	2	5
		4	4978.5	4978.5	0.0	2.4	4710.4	0	97	21	2	4	341.8	341.8	0.0	1.7	292.5	0	142	19	2	4	5321.6	5321.6	0.0	1.4	5137.7	0	206	24	2	4
		5	4845.3	4845.3	0.0	0.6	4654.7	1	67	28	2	5	326.5	326.5	0.0	0.4	313.4	1	136	28	2	5	5171.7	5171.7	0.0	0.4	5024.2	1	140	25	2	5
	15	1	6023.2	6023.2	0.0	2.1	5828.4	0	88	23	3	8	414.0	414.0	0.0	2.4	395.7	0	253	27	3	8	6437.2	6437.2	0.0	2.7	6327.2	0	253	48	3	8
		2	7368.8	7368.8	0.0	22.3	6910.5	1	287	31	2	7	499.9	499.9	0.0	8.0	457.3	2	302	26	2	8	7870.8	7870.5	0.0	21.4	7477.0	1	422	45	3	7
		3	6042.0	6042.0	0.0	1.3	5967.4	0	80	62	3	9	412.6	412.6	0.0	1.1	412.5	0	198	33	3	9	6454.6	6454.6	0.0	1.1	6363.8	0	266	44	3	9
		4	7111.2	7111.2	0.0	3.9	6863.5	0	32	25	2	7	479.6	479.6	0.0	4.4	450.6	0	311	37	2	7	7590.8	7590.8	0.0	7.3	7261.0	0	154	28	2	7
		5	8114.1	8113.6	0.0	14.7	7525.6	2	213	40	3	9	543.3	543.2	0.0	6.3	508.9	2	371	36	3	10	8673.0	8673.0	0.0	15.7	7996.7	2	585	25	3	9
	20	1	10078.9	10078.9	0.0	54.7	9573.2	1	224	30	4	12	703.5	703.5	0.0	8.2	684.0	0	296	68	4	13	10786.6	10785.5	0.0	31.7	10386.7	2	552	34	4	12
		2	8693.6	8693.6	0.0	104.0	8233.3	1	160	24	3	10	601.2	601.2	0.0	46.3	518.3	0	421	27	3	10	9294.8	9294.1	0.0	112.1	8771.2	4	514	23	3	10
		3	9334.0	9333.2	0.0	141.9	9018.6	7	362	33	3	10	644.5	644.5	0.0	113.0	619.6	5	1146	55	3	10	9978.5	9977.7	0.0	68.7	9633.2	5	559	34	3	10
		4	8858.5	8857.8	0.0	160.4	8500.1	1	471	36	4	10	610.9	610.9	0.0	15.8	588.1	0	436	37	4	11	9485.4	9484.7	0.0	120.5	9058.6	2	907	28	4	10
		5	7249.3	7249.3	0.0	26.0	7106.9	1	123	35	3	9	515.3	515.3	0.0	52.7	464.9	1	917	39	3	10	7764.8	7764.8	0.0	34.2	7560.0	0	325	44	3	9
	25	1	10219.2	10218.2	0.0	84.7	9801.4	0	257	31	4	12	717.8	717.8	0.0	59.0	628.8	5	907	39	4	12	10937.0	10937.0	0.0	49.0	10592.1	1	877	47	4	12
		2	11416.2	11415.1	0.0	4947.9	10767.5	4	461	40	4	13	785.8	785.8	0.0	375.0	697.5	5	734	35	4	13	12202.1	12200.9	0.0	4017.2	11504.2	3	819	46	4	13
		3	9677.8	9677.8	0.0	62.0	9364.0	2	544	35	4	12	675.2	675.2	0.0	26.2	643.4	2	781	50	3	12	10353.5	10352.9	0.0	81.2	10021.5	3	827	33	3	12
		4	11468.4	11467.2	0.0	766.6	10935.8	2	444	43	4	13	804.4	804.4	0.0	107.8	762.5	2	1020	39	4	14	12279.3	12278.1	0.0	810.7	11717.7	2	774	34	4	13
		5	10545.5	10544.7	0.0	54.4	10322.4	1	388	44	4	13	750.4	750.4	0.0	28.2	717.2	0	819	41	4	13	11296.8	11296.3	0.0	74.9	11007.6	1	863	36	4	13
	30	1	13944.2	13418.9	3.8	14400.0	12783.9	2	1237	36	5	16	971.6	971.5	0.0	5019.4	851.3	2	1746	33	5	17	14930.1	14535.8	2.6	14400.0	13634.9	3	1732	35	5	16
		2	11459.8	11458.7	0.0	1507.8	10988.3	1	920	44	4	13	812.6	812.5	0.0	1709.8	751.6	1	1714	40	3	13	12272.4	12271.2	0.0	3026.4	11717.8	0	1236	49	4	13
		3	11088.6	10992.3	0.9	14400.0	10368.7	5	1172	36	5	14	781.7	767.6	1.8	14400.0	661.8	6	2279	33	5	14	11871.0	11869.9	0.0	11014.3	11066.1	4	2012	39	5	14
		4	10717.3	10716.3	0.0	13811.0	10147.1	4	1364	30	4	12	747.8	741.8	0.8	14400.0	617.5	11	2168	29	4	12	11468.2	11467.1	0.0	6351.6	10632.9	2	1485	28	4	12
		5	11527.6	11526.7	0.0	127.8	11299.8	4	594	32	4	12	826.1	826.0	0.0	4316.1	777.8	0	1696	44	4	12	12356.8	12355.7	0.0	151.6	11933.6	2	1089	29	4	12

Table C.15: Results for extended Solomon's instances of type 2

		R_e										E_e										$(R+E)_e$									
Cust	Inst	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
25	c201	603.1	603.1	0.0	1.4	599.1	0	309	26	4	10	44.4	44.4	0.0	11.5	39.9	0	467	41	4	10	647.5	647.5	0.0	2.5	641.2	0	467	27	4	10
	c202	603.1	603.1	0.0	13.4	595.9	0	232	42	4	10	43.8	43.8	0.0	48.5	38.1	0	1242	47	4	10	646.9	646.9	0.0	19.7	634.9	0	865	44	4	10
	c203	603.1	603.1	0.0	56.4	593.9	0	430	48	4	10	43.5	43.5	0.0	86.5	37.7	0	1294	73	4	10	646.6	646.6	0.0	74.8	633.0	0	1470	45	4	10
	c204	603.1	603.1	0.0	48.2	592.1	0	451	30	4	10	43.4	43.4	0.0	240.8	37.4	0	1099	73	4	10	646.5	646.5	0.0	78.6	631.0	0	880	52	4	10
	c205	603.1	603.1	0.0	13.0	592.2	0	678	35	4	10	43.9	43.9	0.0	29.5	39.4	0	716	46	4	10	647.3	647.3	0.0	9.0	637.0	0	446	30	4	10
	c206	603.1	603.1	0.0	21.1	594.6	0	296	42	3	10	43.6	43.6	0.0	40.8	38.1	0	1383	32	4	10	646.7	646.7	0.0	21.7	634.5	0	504	35	4	10
	c207	603.1	603.1	0.0	32.0	594.1	0	593	38	4	10	43.3	43.3	0.0	55.3	37.5	0	1127	53	4	10	646.4	646.4	0.0	43.4	631.7	0	1091	41	4	10
	c208	603.1	603.1	0.0	23.2	593.5	0	523	37	4	10	43.4	43.4	0.0	34.7	37.8	0	1374	34	4	10	646.5	646.5	0.0	34.3	635.3	0	780	47	4	10
	r201	761.7	761.7	0.0	4.0	749.2	0	362	29	4	12	54.2	54.2	0.0	8.7	51.7	0	729	50	4	12	816.2	816.2	0.0	5.6	794.2	0	226	30	4	12
	r202	755.7	755.7	0.0	32.9	737.9	0	118	42	4	12	53.0	53.0	0.0	26.9	48.1	0	414	47	4	12	809.0	809.0	0.0	29.2	786.2	0	873	38	4	12
	r203	749.5	749.5	0.0	132.1	733.2	0	579	47	4	12	52.9	52.9	0.0	40.2	47.3	0	811	40	4	12	802.9	802.9	0.0	58.0	778.0	0	876	39	4	12
	r204	749.5	749.5	0.0	134.2	732.5	0	671	49	4	12	52.9	52.9	0.0	125.4	47.3	0	782	38	4	12	802.7	802.7	0.0	120.3	779.6	0	875	44	4	12
	r205	755.7	755.7	0.0	38.5	738.4	0	211	36	4	12	53.3	53.3	0.0	21.8	50.6	0	889	45	4	12	809.0	809.0	0.0	27.1	788.6	0	678	44	4	12
	r206	755.7	755.7	0.0	54.8	736.7	0	775	44	4	12	52.9	52.9	0.0	29.8	49.4	0	839	72	4	12	809.0	809.0	0.0	84.6	786.0	0	891	52	4	12
	r207	749.5	749.5	0.0	83.5	731.3	0	662	48	4	12	52.9	52.9	0.0	37.4	49.6	0	800	46	4	12	802.9	802.9	0.0	87.8	783.8	0	1298	56	4	12
	r208	749.5	749.5	0.0	75.7	731.0	0	358	42	4	12	52.9	52.9	0.0	47.9	49.4	0	689	37	4	12	802.7	802.7	0.0	106.7	780.7	0	1459	39	4	12
	r209	749.5	749.5	0.0	42.8	731.6	0	481	41	4	12	53.2	53.2	0.0	41.5	48.4	0	761	42	4	12	803.2	803.2	0.0	50.0	779.4	0	584	43	4	12
	r210	755.7	755.7	0.0	46.0	737.1	0	337	38	4	12	52.9	52.9	0.0	24.8	49.2	0	567	49	4	12	809.0	809.0	0.0	60.2	784.8	0	415	45	4	12
	r211	749.5	749.5	0.0	136.1	729.0	0	316	47	4	12	52.9	52.9	0.0	53.4	48.8	0	1154	59	4	12	802.7	802.7	0.0	102.5	777.8	0	874	37	4	12
	rc201	956.3	956.3	0.0	28.0	930.6	0	744	56	4	12	68.7	68.7	0.0	31.7	58.2	0	666	34	4	12	1025.4	1025.4	0.0	60.0	954.3	0	1074	30	4	12
	rc202	940.6	940.5	0.0	366.9	912.4	1	712	50	4	11	67.9	67.9	0.0	315.0	57.1	0	1517	51	4	12	1010.2	1010.2	0.0	125.1	969.3	1	1467	25	4	11
	rc203	934.4	934.4	0.0	15.7	931.5	0	266	26	4	11	67.4	67.4	0.0	29.9	60.3	0	780	40	4	12	1003.2	1003.1	0.0	56.8	974.0	1	697	27	4	11
	rc204	934.4	934.4	0.0	5.4	931.8	0	338	47	4	11	67.2	67.2	0.0	64.3	63.4	0	668	37	4	12	1003.2	1003.2	0.0	952.8	981.6	0	1144	59	4	11
	rc205	946.7	946.7	0.0	63.4	933.6	0	881	24	4	11	68.3	68.3	0.0	269.0	57.8	0	1446	49	4	12	1016.6	1016.6	0.0	100.6	993.3	1	363	45	4	11
	rc206	948.5	948.5	0.0	65.4	935.6	0	408	27	4	12	68.2	68.2	0.0	59.5	62.0	1	1708	53	4	12	1017.0	1016.9	0.0	755.0	977.8	0	1478	45	4	12
	rc207	940.7	940.6	0.0	1253.2	924.3	0	1053	36	4	12	67.6	67.6	0.0	58.3	59.6	1	816	41	4	12	1008.4	1008.3	0.0	7306.9	967.8	1	2029	29	4	12
	rc208	934.4	934.4	0.0	207.4	922.8	0	659	55	4	11	67.2	67.2	0.0	23.0	64.4	0	746	40	4	12	1002.9	1002.9	0.0	157.5	981.4	0	1101	33	4	11
40	c201	1032.8	1032.8	0.0	48.3	1017.2	0	1368	25	5	15	75.5	75.5	0.0	229.9	62.4	0	2236	29	5	15	1108.3	1108.3	0.0	43.8	1090.1	0	1181	43	5	15
	c202	1031.8	1031.8	0.0	372.9	1013.2	0	1274	47	5	15	74.9	74.9	0.0	14073.8	61.8	0	2747	51	5	15	1106.8	1106.7	0.0	955.4	1076.8	0	1840	42	5	15
	c203	1031.1	1031.0	0.0	3881.6	1011.9	0	1968	60	5	15	74.7	74.0	1.0	43200.0	63.4	0	4554	55	5	15	1106.2	1106.1	0.0	10685.1	1072.4	0	3031	49	5	15
	c204	1031.1	1029.1	0.2	43200.0	1009.5	0	2076	55	5	15	74.5	73.3	1.5	43200.0	60.8	0	3838	61	5	15	1105.9	1105.8	0.0	16940.6	1075.0	0	2173	62	5	15
	c205	1031.1	1031.0	0.0	871.2	1011.7	0	1830	40	5	15	75.1	75.1	0.0	23630.3	60.9	0	2617	47	5	15	1106.4	1106.4	0.0	881.7	1073.7	0	1633	38	5	15
	c206	1031.1	1031.0	0.0	2011.9	1009.1	0	1726	33	5	15	74.8	73.4	1.8	43200.0	60.9	0	3321	49	5	15	1106.0	1105.9	0.0	8368.8	1074.1	0	2944	41	5	15
	c207	1031.1	1031.0	0.0	5757.3	1010.0	0	2628	51	5	15	74.4	74.4	0.0	5567.9	60.6	0	3439	52	5	15	1105.9	1105.8	0.0	8358.9	1074.9	0	2271	64	5	15
	c208	1031.1	1031.0	0.0	1936.7	1012.2	0	1990	49	5	15	74.5	74.0	0.6	43200.0	60.7	3	2665	53	5	15	1105.9	1105.8	0.0	3547.3	1071.5	0	1443	46	5	15
	r201	1204.8	1204.8	0.0	437.7	1160.3	1	1111	37	6	19	85.8	85.8	0.0	2730.5	74.4	0	3568	32	6	20	1291.6	1291.6	0.0	348.1	1233.7	0	1687	37	6	19
	r202	1169.9	1169.9	0.0	11666.6	1121.2	4	1566	53	6	19	83.2	82.2	1.2	43200.0	71.1	9	5734	40	6	19	1253.2	1253.2	0.0	8183.1	1188.0	0	2305	38	6	19
	r203	1155.4	1155.4	0.0	33302.4	1110.4	0	774	42	6	19	81.5	81.5	0.0	17013.0	71.9	0	1928	69	6	19	1237.0	1236.9	0.0	40920.9	1180.4	0	1563	53	6	19
	r204	1155.4	1148.7	0.6	43200.1	1108.9	1	291	45	6	19	81.1	81.1	0.0	37471.4	71.8	0	1914	57	6	19	1236.6	1221.5	1.2	43200.0	1177.2	0	1275	51	6	19
	r205	1163.5	1163.5	0.0	5921.7	1122.1	2	1024	45	6	19	82.4	82.3	0.0	7086.4	72.1	0	3252	44	6	19	1245.9	1245.9	0.0	8098.5	1196.3	0	3244	44	6	19
	r206	1161.0	1149.9	1.0	43200.0	1111.3	0	583	47	6	19	81.7	81.7	0.0	5167.2	74.9	0	2301	72	6	19	1242.8	1237.6	0.4	43200.0	1181.1	2	3483	55	6	19
	r207	1155.4	1141.9	1.2	43200.0	1107.0	0	797	50	6	19	81.5	81.5	0.0	22529.7	70.0	0	2488	42	6	19	1237.0	1236.9	0.0	39388.0	1178.9	0	771	54	6	19
	r208	1155.4	1144.3	1.0	43200.0	1103.9	0	376	41	6	19	81.1	80.9	0.4	43200.0	71.4	0	3191	50	6	19	1236.6	1228.4	0.7	43200.0	1182.9	0	1810	64	6	19
	r209	1155.4	1155.4	0.0	42044.0	1112.2	0	1497	37	6	19	82.1	77.3	5.9	43200.0	71.0	8	3092	46	6	19	1237.8	1220.5	1.4	43200.0	1183.4	2	1784	49	6	19
	r210	1161.0	1161.0	0.0	14821.3	1113.6	0	1302	48	6	19	81.7	81.7	0.0	2832.5	72.0	0	1623	58	6	19	1242.8	1234.4	0.7	43200.0	1179.7	0	2038	46	6	19
	r211	1155.4	1138.8	1.4	43200.0	1106.6	0	502	46	6	19	81.1	80.6	0.6	43200.0	71.4	0	3347	59	6	19	1236.6	1222.2	1.2	43200.0	1180.5	0	3028	54	6	19
	rc201	1582.9	1582.9	0.0	959.5	1529.2	0	1092	48	6	18	114.1	105.6	7.4	43200.0	94.2	2	1223	63	6	18	1697.1	1697.0	0.0	2540.3	1604.4	3	2397	42	6	18
	rc202	1533.9	1521.8	0.8	43200.0	1503.6	3	984	49	6	17	111.3	108.2	2.9	43200.0	93.6	1	3139	74	6	18	1647.2	1637.0	0.6	43200.0	1593.9	0	2844	45	6	17
	rc203	1527.7	1527.6	0.0	1542.6	1497.9	0	954	33	6	17	110.7	110.7	0.0	509.2	90.1	0	1170	42	6	18	1640.1	1640.0	0.0	5064.3	1595.5	2	2609	43	6	17
	rc204	1527.7	1527.7	0.0	110.3	1517.8	0	381	38	6	17	110.6	110.6	0.0	2663.9	96.9	0	2678	39	6	18	1640.1	1640.0	0.0	9900.8	1600.5	0	1752	39	6	17
	rc205	1543.5	1543.4	0.0	27642.5	1490.9	0	1592	38	6	18	111.9	96.7	13.6	43200.0	91.4	4	2668	40	6	18	1655.4	1629.2	1.6	43200.0	1600.7	7	2087	61	6	18
	rc206	1551.2	1551.1	0.0	4719.6	1527.2	1	2226	53	6	18	112.4	100.7	10.4	43200.0	93.8	0	2676	44	6	18	1663.8	1663.7	0.0	39338.2	1602.9	2	2373	41	6	18
	rc207	1533.6	1527.7	0.4	43200.0	1506.4	0	1472	41	6	17	110.8	109.6	1.1	43200.0	90.1	0	3426	48	6	18	1643.9	1628.9	0.9	43200.0	1598.9	0	3363	49	6	17
	rc208	1527.7	1527.7	0.0	1684.8	1498.3	0	803	46	6	17	110.6	110.6	0.0	237.6	97.0	0	1044	42	6	18	1639.8	1639.7	0.0	7961.0	1598.1	0	2476	40	6	17

Table C.16: Results using multicore processors for Set A instances with 35–50 customers

	Cust	Inst	R_e UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A_1	35	1	11719.2	11237.2	4.1	43200.0	10962.9	12	5471	54	5	13
		2	12145.5	11927.6	1.8	43200.0	11495.5	9	3031	54	5	15
		3	12346.4	12345.2	0.0	20642.0	11955.2	11	2866	37	5	15
		4	12925.3	12924.1	0.0	30214.8	12528.6	5	3409	37	5	15
		5	12384.3	12383.3	0.0	29126.0	11900.5	0	1144	48	6	18
	40	1	14546.6	14055.3	3.4	43200.0	13792.5	4	1916	58	6	18
		2	15741.8	15740.9	0.0	13947.5	15181.1	1	1370	46	7	20
		3	13230.2	12735.2	3.7	43200.1	12415.0	7	3905	52	6	17
		4	15189.5	15122.6	0.4	43200.0	14560.9	16	3179	47	7	20
		5	14554.9	14448.5	0.7	43200.0	14075.2	0	2293	46	6	19
	45	1	13832.5	13245.7	4.2	43200.1	12999.7	3	3065	64	6	18
		2	18423.5	18026.8	2.2	43200.0	17692.1	4	1753	67	7	22
		3	17548.7	17283.6	1.5	43200.0	16809.4	1	1768	50	7	21
		4	15231.5	14767.5	3.0	43200.0	14429.8	10	3801	64	7	21
		5	17467.0	17126.7	1.9	43200.0	16781.1	12	2274	65	8	23
Set A_2	35	1	13338.1	13000.9	2.5	43200.0	12340.2	11	1607	42	5	14
		2	16285.8	16285.7	0.0	1755.4	15114.2	5	759	42	6	20
		3	13709.3	13707.9	0.0	8732.7	13091.3	4	1814	45	5	14
		4	16511.4	16510.1	0.0	9765.1	15590.8	9	1233	45	6	17
		5	15706.0	15704.8	0.0	9491.5	14796.8	8	1676	48	5	16
	40	1	15866.1	15865.0	0.0	32162.3	15050.8	2	813	47	6	19
		2	16751.9	16563.7	1.1	43200.0	15858.9	4	1167	36	6	18
		3	16867.9	16866.9	0.0	3298.6	16156.5	9	1854	50	7	20
		4	17327.7	16959.0	2.1	43200.0	16506.2	14	1771	52	6	19
		5	12989.9	12333.8	5.1	43200.0	11903.8	19	3308	41	5	15
	45	1	17395.5	17393.9	0.0	6142.2	16659.3	9	2285	50	8	23
		2	18299.8	18298.0	0.0	41018.0	17418.5	17	1752	66	7	23
		3	18854.9	18627.9	1.2	43200.0	18056.7	8	1878	44	8	23
		4	16156.7	15830.2	2.0	43200.0	15187.7	7	3575	38	6	19
		5	22695.8	22693.7	0.0	37956.5	21613.5	5	1321	57	8	26
	50	1	23045.5	22630.7	1.8	43200.0	22117.4	17	3932	43	8	26
		2	20734.0	19945.0	3.8	43200.0	19248.1	16	3064	39	8	24
		3	18482.6	17942.6	2.9	43200.0	17595.5	3	3278	47	7	20
		4	18537.6	18293.1	1.3	43200.0	17778.5	5	1269	44	7	23
		5	22160.8	21761.2	1.8	43200.0	20922.1	3	1607	43	8	25

	Cust	Inst	E_e UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A_1	35	1	846.2	807.7	4.6	43200.0	691.9	4	2734	39	5	13
		2	880.0	848.5	3.6	43200.0	735.6	4	3748	38	5	15
		3	889.5	889.4	0.0	21866.9	762.9	19	3273	37	5	15
		4	941.7	941.6	0.0	36895.7	808.6	20	4282	30	5	15
		5	894.5	894.4	0.0	15121.7	752.6	1	2687	36	6	18
	40	1	1050.1	1019.4	2.9	43200.0	899.9	6	5416	40	6	18
		2	1144.1	1133.8	0.9	43200.0	1036.8	3	3824	42	7	20
		3	960.9	915.4	4.7	43200.0	759.8	6	4500	63	6	18
		4	1099.0	1098.9	0.0	23208.1	984.3	14	5721	43	7	20
		5	1056.1	1042.4	1.3	43200.0	996.1	3	4015	59	6	19
	45	1	991.7	955.6	3.6	43200.0	840.5	14	8605	37	6	19
		2	1343.0	1307.2	2.7	43200.0	1203.8	8	7610	36	7	22
		3	1284.5	1236.2	3.8	43200.0	1053.9	1	8810	50	7	22
		4	1104.8	1055.9	4.4	43200.0	928.9	29	8422	52	7	21
		5	1254.4	1232.0	1.8	43200.0	1116.2	31	6849	42	8	23
Set A_2	35	1	941.9	941.8	0.0	31873.4	804.1	29	4738	30	5	15
		2	1158.5	1158.4	0.0	18306.0	986.1	5	2471	47	6	20
		3	982.4	954.6	2.8	43200.0	822.4	6	2328	33	5	14
		4	1171.3	1171.3	0.0	38648.4	997.3	13	2774	47	6	18
		5	1108.8	1108.7	0.0	42292.8	946.2	6	2742	34	5	16
	40	1	1131.8	1131.7	0.0	6219.5	982.1	7	1930	35	6	20
		2	1189.6	1172.6	1.4	43200.0	1010.3	3	3376	37	6	18
		3	1202.4	1202.3	0.0	30554.3	1023.7	2	1955	47	7	21
		4	1233.5	1233.4	0.0	7308.4	1111.1	10	3392	25	6	20
		5	920.0	861.2	6.4	43200.0	763.2	17	3425	42	5	17
	45	1	1242.9	1242.9	0.0	1802.5	1076.3	1	2044	50	8	24
		2	1288.6	1271.8	1.3	43200.0	1085.5	14	4237	36	7	23
		3	1355.5	1326.3	2.2	43200.0	1169.6	5	3513	38	8	23
		4	1142.6	1121.5	1.8	43200.0	979.6	6	4761	31	6	19
		5	1617.5	1583.3	2.1	43200.0	1403.6	3	4321	50	8	26
	50	1	1649.2	1637.1	0.7	43200.0	1513.3	6	5118	43	8	26
		2	1461.9	1429.1	2.2	43200.0	1271.5	26	5673	30	8	25
		3	1304.5	1249.7	4.2	43200.0	1149.9	6	4313	36	7	21
		4	1309.1	1266.1	3.3	43200.1	1128.8	5	2372	52	7	24
		5	1578.2	1553.6	1.6	43200.0	1454.6	8	3312	35	8	26

	Cust	Inst	$(R+E)_e$ UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip
Set A_1	35	1	12510.2	12104.5	3.2	43200.0	11693.5	0	3485	47	5	13
		2	13025.5	12663.3	2.8	43200.0	12218.1	5	1695	50	5	15
		3	13236.1	13106.4	1.0	43200.0	12706.7	10	2637	44	5	15
		4	13868.2	13838.5	0.2	43200.0	13305.6	10	4865	40	5	15
		5	13286.7	13285.7	0.0	20446.6	12709.6	1	1495	43	6	18
	40	1	15546.0	14931.9	4.0	43200.0	14608.2	12	5103	53	6	18
		2	16887.0	16787.9	0.6	43200.1	16156.1	0	3204	55	7	20
		3	14183.7	13648.1	3.8	43200.0	13238.9	3	4583	59	6	17
		4	16292.1	16229.3	0.4	43200.0	15563.0	2	4024	64	7	20
		5	15625.8	15264.1	2.3	43200.0	14965.9	3	4185	59	6	19
	45	1	14760.2	14175.2	4.0	43200.0	13817.3	4	5690	54	6	18
		2	19747.2	19340.7	2.1	43200.0	18942.6	3	7054	67	7	22
		3	18899.1	18434.2	2.5	43200.0	17974.2	1	4837	54	7	22
		4	16341.1	15826.7	3.1	43200.0	15398.8	8	5617	49	7	21
		5	18736.6	18307.2	2.3	43200.0	17850.3	18	4315	56	8	23
Set A_2	35	1	14287.9	13909.9	2.6	43200.0	13108.8	18	2912	38	5	14
		2	17450.5	17449.3	0.0	3397.5	16138.0	15	3099	32	6	20
		3	14696.5	14695.1	0.0	11645.0	13828.6	9	3204	36	5	14
		4	17694.9	17693.2	0.0	25076.3	16599.0	6	1778	37	6	17
		5	16817.4	16815.8	0.0	18041.8	15740.4	9	2594	43	5	16
	40	1	17008.8	17007.3	0.0	21628.1	16050.1	10	3197	49	6	19
		2	17954.7	17952.9	0.0	41897.4	16900.3	12	2304	50	6	18
		3	18085.0	18083.4	0.0	5495.7	17039.1	6	3058	50	7	20
		4	18565.5	18164.3	2.2	43200.0	17526.7	6	5158	45	6	20
		5	13886.7	13218.2	4.8	43200.0	12651.4	24	4235	39	5	16
	45	1	18662.0	18660.4	0.0	8093.6	17684.0	6	2824	41	8	23
		2	19596.7	19337.3	1.3	43200.0	18204.9	14	2576	45	7	22
		3	20214.0	20212.5	0.0	7452.0	19205.3	2	2111	38	8	23
		4	17322.9	16859.6	2.7	43200.0	16154.7	7	2342	38	6	19
		5	24318.9	24075.2	1.0	43200.0	22998.6	3	3652	46	8	26
	50	1	24723.8	24141.0	2.4	43200.0	23646.8	5	3907	51	8	26
		2	21947.0	21112.5	3.8	43200.0	20476.4	14	4361	43	8	24
		3	19707.0	19170.5	2.7	43200.0	18708.9	3	4128	40	7	21
		4	19849.8	19707.0	0.7	43200.0	19078.5	2	4748	46	7	23
		5	23729.1	23252.5	2.0	43200.0	22417.4	0	1800	46	8	25