



CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Formulations and Exact Algorithms for Drone Routing Problem

Chun Cheng
Yossiri Adulyasak
Louis-Martin Rousseau

July 2018

CIRRELT-2018-31

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Formulations and Exact Algorithms for Drone Routing Problem

Chun Cheng^{1,*}, Yossiri Adulyasak², Louis-Martin Rousseau¹

¹ Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

² HEC Montréal, Department of Logistics and Operations Management, 3000, chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. Drone transportation is known as a potential major contributor in improving efficiency and alleviating first and 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 the routes for drones is challenging due to multiple operational characteristics including multi-trip operations, recharge planning, as well as calculating energy consumption. Although various challenges arise, studies on drone operations are still scarce. To fill some important gaps, this paper solves a multi-trip drone routing problem, where drones' energy consumption is influenced by payload and travel distance whereas such relationships are nonlinear. To tackle the nonlinear (convex) energy function, which can be incorporated in the objective function within this particular problem, we propose two types of cuts, logical cuts and subgradient cuts. This allows us to make an exact calculation of energy consumption, instead of using the linear approximation method as in the literature, which can fail to detect infeasible routes due to excess energy consumption. We introduce two formulations to solve the problem, one with a drone index and the other without, which are further enhanced by valid inequalities. Branch-and-cut algorithms are developed for the formulations and benchmark instances (with up to 50 customers) are first generated for this problem. Extensive numerical experiments indicate that the formulation without a drone index is superior in solving more instances to optimality and providing high quality solutions for all the generated instances. The results also indicate that even though the original model is nonlinear, the proposed approach is highly efficient and the performance does not deteriorate so much compared to the linear approximation model in which the structure is much simpler. In addition, in multiple instances, the linear approximation model yields infeasible routes based on energy constraints. There is also a significant gap in energy calculation between the linear approximation model and our exact model.

Keywords. Drone routing, multi-trip, nonlinear energy function, logical cut, subgradient cut, branch-and-cut.

Acknowledgements. The first author wishes to acknowledge the supporters of this study: the Institute for Data Valorization (IVADO) and the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT). We thank Calcul Québec for providing computing facilities for the experiments.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

* Corresponding author: chun.cheng@polymtl.ca

1 Introduction

In recent years unmanned aerial vehicles (UAVs), also known as 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).

Compared to traditional trucks, drones have some specific advantages: (i) they can save labor, because no drivers (or pilots) are needed; (ii) they travel much faster than trucks; and (iii) they are not restricted to road networks. 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, drones could provide main relief for inner cities to reduce negative implications on congestion, safety, and environment, which come from rapid urbanization and population growth (Heutger and Kückelhaus 2014). This new transportation mode also has the potential to provide an innovative solution for the increasingly complex city logistics (Savelsbergh and Van Woensel 2016).

On the other hand, some unique characteristics of drones have presented new operational challenges. Limited battery capacity influences 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? How do we determine whether we need to locate some recharging stations to extend drones' flight range? Furthermore, the payload of drones 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? Although various operational challenges arise, current research about drones still focuses only on engineering issues and studies addressing these other challenges are scarce (Murray and Chu 2015).

Existing research on drone operations normally assumes that drone flight duration is a constant number. However, flight duration is actually influenced by several factors including battery energy capacity, payload, and travel distance. In addition, no benchmark instances and efficient exact algorithms are available for the drone routing problem (DRP), which poses a limitation on algorithm evaluation. To fill some gaps in this area, in this paper we solve a multi-trip drone routing problem (MTDRP) which explicitly considers the influence of payload and distance on flight duration. We propose two modeling schemes and develop exact algorithms for our formulations. We also generate several benchmark instance sets, which are open 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 and constructs the mathematical models. Section 4 introduces valid inequalities to strengthen the models. Section 5 presents the techniques for exact calculation of energy consumption and provides the details of our exact algorithms. Numerical tests and analyses are presented in Section 6. This is followed by the conclusions in Section 7.

2 Literature Review

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

2.1 Drone Routing Problem

Sundar and Rathinam (2014) study a single UAV routing problem where multiple depots are available for refueling the drone. The objective is to find a minimum path distance where each target is visited. Choi and Schonfeld (2017) study an automated drone delivery system, in which 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 swarm of UAVs to deliver items to target locations. They consider several possible constraints, including drone's payload, flight range, and the volume of each item (whether this item can be put in a box and carried by a drone). 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 energy function, which is nonlinear, they use a linear approximation function to calculate the energy consumption which varies linearly with payload and battery weight. In their study, battery weight is a decision variable. In order to save cost, each drone can perform multiple trips and visit multiple customers per trip. The authors use a simulated annealing (SA) heuristic to solve the models.

Murray and Chu (2015) consider two types of truck-drone delivery problems. The first is the flying sidekick traveling salesman problem (FSTSP), which is applicable to the case where the distribution center (DC) is relatively far from the customer locations and a single UAV is available. In the FSTSP, the truck carries one drone to deliver parcels. 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 latter one is the parallel drone scheduling traveling salesman problem (PDSTSP), which is designed for the situation, where a significant proportion of customers are located within a UAV's flight range from the DC. In the PDSTSP, drones depart from the depot and deliver parcels to customers, and then return to the depot. The truck services customers without carrying any drone. Simple heuristics are used to solve both problems. Ponza (2016) uses a SA heuristic to solve the FSTSP. Carlsson and Song (2017) use theoretical analysis to demonstrate that for FSTSP the improvement in efficiency is proportional to the square root of the speed ratio of the truck and the drone. Agatz et al. (2018) extend the FSTSP by allowing the truck to wait at the start node for the drone to return. A route first–cluster second heuristic method is used. Bouman et al. (2017) use a dynamic programming (DP) approach for the same problem. Ha et al. (2015) extend the FSTSP by considering time span—the maximum allowable time that either the truck or the drone can wait for each other at the assembly point. Both route first–cluster second and cluster first–route second methods are used. 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.

Wang et al. (2017) and Poikonen et al. (2017) consider a fleet of homogeneous trucks with multiple drones equipped with each truck. They focus on the worst-case analysis, which reveals the amount of time (best case scenario) that can be saved by using truck-drone rather than trucks alone. Pugliese and Guerriero (2017) extend the problem by considering time window constraints. They conclude that the use of drones is not economically convenient but can reduce negative environmental impacts (from a green perspective) and improve service quality.

Mathew et al. (2015) consider a team of one truck and one drone, in which the truck travels along the street network and carries the drone. The drone is responsible for delivering parcels to customers, and the truck is only used to carry the drone. The truck can wait there for the drone to come back or go to the next vertex to rendezvous with the drone. Luo et al. (2017) study a similar problem, but in their problem the drone can visit several targets per trip, and the truck and the drone must meet at the next vertex. Campbell et al. (2017) use a continuous approximation model to derive general insights. Ferrandez et al. (2016) assume that the truck follows the route generated by solving a traveling salesman problem (TSP), and that from each truck stop one or more drones can be launched. Each truck stop is a hub for drone deliveries. They use a K-means clustering algorithm to find truck stops and a genetic algorithm to determine the truck TSP tour. In their work, drones are not constrained by flight range. Moshref-Javadi and Lee (2017) use a truck-drone tandem system to minimize latency in a customer-oriented distribution system. In their problem, at each stop site, the truck will wait until all drones come back. The truck then carries drones to the next site. They compare the benefits of using drones for a single trip versus multiple trips. Ulmer and Thomas (2017) study a same-day delivery problem with trucks and drones, in which customer orders come dynamically during a shift. The dispatcher first determines whether to accept a customer for same-day delivery, and then decides to use a truck or a drone for the delivery. The authors present a Markov decision model and an approximate DP algorithm to solve the problem. Dayarian et al. (2017) present a same-day delivery problem with drone resupply, where trucks make deliveries to customers and drones are used to resupply the trucks. A heuristic is developed for the system with one truck and one drone. Ham (2018) extends the PDSTSP by assuming that drones can perform two types of tasks: drop-off and pickup. After a drone finishes the delivery task, it can fly back to the depot to begin a new trip or fly to the next customer to pick up a returned parcel. A constraint programming method is proposed.

Table 1: Summary of papers on drone routing problem

Authors	Problem					Solution method
	# truck	# drone	# customers per drone trip	multi-trip	energy function	
Sundar and Rathinam (2014)	N/A	1	multiple	✓		Route construction and improvement heuristics
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
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
Carlsson and Song (2017)	1	1	1	✓		Mathematical analysis
Agatz et al. (2018)	1	1	1	✓		Route first-cluster second
Bouman et al. (2017)	1	1	1	✓		Dynamic programming
Ha et al. (2015)	1	1	1	✓		Route first-cluster second, cluster first-route second
Marinelli et al. (2017)	1	1	1	✓		Greedy randomized adaptive search procedure
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
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
Campbell et al. (2017)	1	1/multiple	1	✓		Continuous approximation model
Ferrandez et al. (2016)	1	multiple	1	✓		K-means clustering, genetic algorithm
Moshref-Javadi and Lee (2017)	1	multiple	1	✓		Mixed-integer linear programming
Ulmer and Thomas (2017)	multiple	multiple	1	✓		Approximate dynamic programming
Dayarian et al. (2017)	1	1	multiple	✓		Route generation, order release heuristic
Ham (2018)	multiple	multiple	multiple	✓		Constraint programming, variable ordering heuristics
This paper	N/A	multiple	multiple	✓	✓	Branch-and-cut

N/A: trucks are not used in the system.

Based on the literature, we make the following observations: (1) the truck-drone tandem system is the most intensively studied area; (2) almost all mentioned studies assume that drone flight range is a constant number, except Dorling et al. (2017), which suggests that drone duration is related to load and travel distance; (3) most papers addressing the truck-drone system assume that during each trip a drone can only visit one customer. However, the papers on the drone-only system always assume that there are multiple drones and that each drone can cover several customers per trip; (4) for the truck-drone tandem system, the common assumption

is that there is only one truck and one drone. This is due to the fact that one needs to consider time synchronization between the truck and the drone in the tandem system, which makes the problem difficult; (5) no benchmark instance is available for algorithm evaluation; (6) no efficient exact algorithm is developed for the DRP.

2.2 Multi-trip Vehicle Routing Problem

Fleischmann (1990) is the first to study MTRP. 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 MTRP. The first one is based on the generation of all feasible trips, while the second one requires the generation of all feasible schedules. A schedule of a vehicle is a subset of trips which satisfy the side constraints. They derive lower bounds for the models and embed the bounds into an exact solution method. Hernandez et al. (2016) develop two similar formulations for the MTRP with time windows (MTRPTW) and use branch-and-price (B&P) algorithms. They compare the two models on instances with the first 25 customers of Solomon’s “C2”, “R2”, and “RC2” instances (Solomon 1987).

Azi (2011) develops a B&P algorithm for the MTRPTW with a trip duration constraint. In their setting, it is not mandatory to visit every customer, so a hierarchical objective is used: first the number of served customers is maximized; then for a fixed number of customers, the travel distance is minimized. Their numerical tests focus on instances formed by the first 25 and 40 customers of Solomon’s type 2 instance sets. Macedo et al. (2011) propose a network flow model based on generated trips for the same problem. They can solve more instances to optimality. 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. Their algorithm is comparable with that of Azi (2011) and outperforms that of Macedo et al. (2011) on the “R2” instances. Later, 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. For more details on the MTRP, see the review paper by Cattaruzza et al. (2016).

The contributions of our study are fourfold. First, we explicitly represent drones’ energy consumption as a function of payload and travel distance, instead of simply assuming that their flight range is a constant number. To tackle the nonlinear energy function, instead of relying on the linear approximation method as proposed in Dorling et al. (2017), we propose two types of cuts for exact energy calculation, because the solution generated by the linear approximation method may either be infeasible (since it violates the energy constraint) or suboptimal (since it overestimates the energy consumption). Our numerical tests in Section 6.4 confirm this observation. Second, two formulation schemes are presented, one with a drone index and the other without, which are solved by branch-and-cut (B&C) algorithms. To the best of our knowledge, this paper is the first to formulate a MTRP and use exact algorithms 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 formulations and algorithms.

3 Formulations

In this section, we first introduce the problem and notation, then present the mathematical models.

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 $N^- = \{1, \dots, n + 1\}$. $A = \{(i, j) : i \in \{0\}, j \in N' \text{ and } i \in N', j \in N^-, i \neq j\}$ is the set of arcs. Sets $\delta^-(i)$ and $\delta^+(i)$ represent node i 's preceded and succeeded 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 homogeneous drones $K = \{1, \dots, |K|\}$ 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 is allowed to perform several trips as long as it returns to the ending depot before time b_{n+1} . We assume drone speed 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 .

The problem consists in designing a set of drone routes, such that the objective function is minimized and the following operational constraints are satisfied:

- Each route starts at depot 0 and ends at depot $n + 1$;
- Every customer is visited exactly once;
- The sum of duration of trips assigned to the same drone does not exceed b_{n+1} ;
- Drone weight capacity constraint, battery energy constraint, and customers' time windows must be respected.

3.2 Mathematical Models

According to Cattaruzza et al. (2014), 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. The second and the third one are the 3-index formulations with either a trip index, or with a vehicle index, respectively. And the last one is the 2-index formulation, which does not include any of these two indexes. For the formulations involving 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. As our problem has no limit on the number of trips that each drone can perform, we adopt the latter two modeling schemes, i.e., the 3-index formulation with a vehicle index and the 2-index formulation, and formulate the MTDRP.

3.2.1 Formulation With a Drone Index (3-index Formulation)

We use the following **decision variables**:

$y_i^k = 1$ if and only if customer i is visited by drone k , 0 otherwise.

$y_0^k \in \{0, 1, 2, \dots\}$ the number of trips performed by drone k .

$x_{ij}^k = 1$ if drone k travels through arc (i, j) , 0 otherwise.

$z_{ij}^k = 1$ if a trip finishing with customer i is followed by another trip visiting j as the first customer (performed by the same drone k), 0 otherwise.

$q_{ij}^k =$ the product weight carried by drone k through arc (i, j) .

$\tau_i^k =$ the start of service time at node $i \in N^-$ when serviced by drone k .

Constraints. We organize the constraints into seven groups, including

(i) **Route feasibility:**

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

$$\sum_{j \in \delta^-(i)} x_{ji}^k + \sum_{j \in \delta^+(i)} x_{ij}^k = 2y_i^k \quad \forall i \in N', \forall k \in K, \quad (2)$$

$$\sum_{j \in \delta^+(0)} x_{0j}^k = \sum_{j \in \delta^-(n+1)} x_{j,n+1}^k \quad \forall k \in K, \quad (3)$$

$$\sum_{j \in \delta^+(0)} x_{0j}^k + \sum_{j \in \delta^-(n+1)} x_{j,n+1}^k = 2y_0^k \quad \forall k \in K. \quad (4)$$

Constraints (1) ensure that each customer is visited exactly once. Equations (2) are the degree constraints at customers. Constraints (3) indicate that for each drone the number of trips leaving the starting depot is equal to the number arriving at the ending depot. Constraints (4) are degree constraints at the starting depot.

(ii) **Customer demand constraint:**

$$\sum_{i \in \delta^-(j)} q_{ij}^k - \sum_{i \in \delta^+(j)} q_{ji}^k = d_j y_j^k \quad \forall j \in N', \forall k \in K. \quad (5)$$

Equations (5) impose that each customer's demand must be satisfied, and also eliminate sub-tours.

(iii) **Drone weight capacity constraint:**

$$q_{ij}^k \leq Q x_{ij}^k \quad \forall (i, j) \in A, \forall k \in K, \quad (6)$$

$$q_{i,n+1}^k = 0 \quad \forall i \in N', \forall k \in K. \quad (7)$$

Constraints (6) guarantee that drone weight capacity is respected. Equations (7) indicate that drones cannot carry any product from a customer to the ending depot.

(iv) **Drone energy constraint:**

Leishman (2006) describes the energy consumption, $P(q)$, of a single rotor helicopter 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

$$P(q_{ij}^k) = (W + m + q_{ij}^k)^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varsigma h}}, \quad (8)$$

where W is the frame weight (kg), m is the battery weight (kg), q_{ij}^k is the payload (kg), g is the gravity (N), ρ is the fluid density of air (kg/m^3), ς is the area of spinning blade disc (m^2), and the unit of P is *Watt*. In order to avoid the undesirable nonlinearity of Equation (8), the authors propose to approximate power consumption as

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

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 falls into the range between points A and B, the linear approximation function overestimates the energy consumption but will not affect the feasibility of generated drone routes. However, when the sum is larger than B, the actual energy consumption will be underestimated if we use the linear approximation method, and the generated routes may violate the battery's energy capacity. Therefore, we use Equation (8) to compute power consumption exactly. To do this, we introduce a variable f_i^k

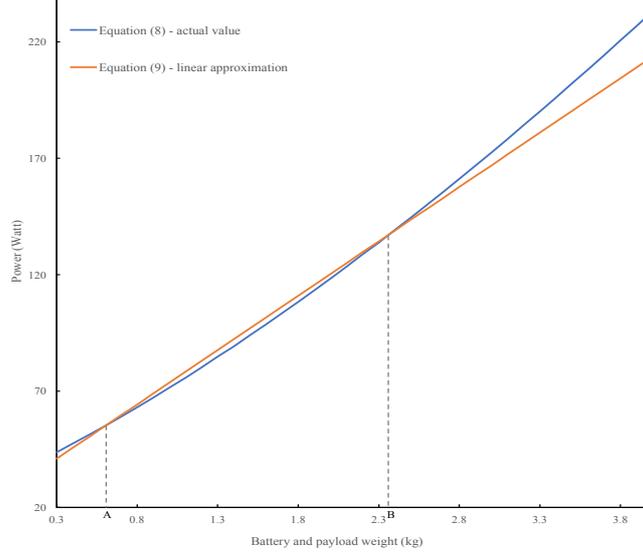


Figure 1: Energy calculation by exact and linear approximation methods (Dorling et al. 2017)

to track the accumulated energy consumption of drone k upon arrival at node i (kWh). Thus, drones' energy consumption constraints are written as

$$f_0^k = 0 \quad \forall k \in K, \quad (10)$$

$$f_i^k + [(W + m + q_{ij}^k)^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varsigma h}} / 1000] t_{ij} / 3600 \leq M_{ij}(1 - x_{ij}^k) + f_j^k \quad \forall (i, j) \in A, \forall k \in K, \quad (11)$$

$$f_{n+1}^k \leq \sigma \quad \forall k \in K. \quad (12)$$

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 give it a new battery. Equations (11) establish the energy relationship between node i and its immediate successor j , and M_{ij} is an arbitrary large constant. We can observe that, when $x_{ij}^k = 0$, according to Equations (6), q_{ij}^k also gets value 0, then we can set $M_{ij} = [(W + m)^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varsigma h}} / 1000] \times (t_{ij} / 3600) + \sigma$ (σ is the energy capacity of the battery (kWh)). 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 5.1, we introduce different types of cuts to tackle this group of constraints implicitly.

We also give the linear approximation version of constraints (11) which is adapted from Dorling et al. (2017):

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

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

(v) **Time window constraint:**

$$\tau_i^k + t_{ij} - M''_{ij}(1 - x_{ij}^k) \leq \tau_j^k \quad \forall i \in N', \forall j \in N^-, \forall k \in K, \quad (14)$$

$$a_i \leq \tau_i^k \leq b_i \quad \forall i \in N^-, \forall k \in K. \quad (15)$$

Constraints (14) establish the time relationship between customer i and its immediate successor j . According to Desaulniers et al. (2014), we can set the large constants $M''_{ij} = \max\{b_i +$

$t_{ij} - a_j, 0\}$. Constraints (15) denote that the time window constraint must be respected. We remark that, if node i is not visited by drone k , constraints (14) will not be active and the variable τ_i^k simply takes the value arbitrarily in the range $[a_i, b_i]$. Here we impose the time window constraint instead of the deadline constraint, because the latter is a special case of the former with $a_i = 0, \forall i \in N$. Furthermore, many revealed drone samples, like Amazon Prime Air and DHL Parcelcopter, show that when drones deliver parcels they land at customer sites; therefore, without loss of generality, we assume that drones can wait at customer locations without energy consumption until the opening of the time window. We can also incorporate the energy consumption during the waiting time, which can be obtained by multiplying the waiting time by the energy consumption per unit time, to the left-hand side of Equations (11).

(vi) **Time relationship between trips:**

$$\tau_i^k + (t_{i,n+1} + t_{0j}) \leq \tau_j^k + (1 - z_{ij}^k)M_{ij}'''' \quad \forall i, j \in N', i \neq j, \forall k \in K, \quad (16)$$

$$\sum_{\substack{i \in N' \\ i \neq j}} z_{ij}^k \leq x_{0j}^k \quad \forall j \in N', \forall k \in K, \quad (17)$$

$$\sum_{\substack{j \in N' \\ j \neq i}} z_{ij}^k \leq x_{i,n+1}^k \quad \forall i \in N', \forall k \in K, \quad (18)$$

$$\sum_{j \in N'} x_{0j}^k = \sum_{i \in N'} \sum_{\substack{j \in N' \\ j \neq i}} z_{ij}^k + 1 \quad \forall k \in K. \quad (19)$$

Equations (16) establish the time relationship between consecutive trips performed by the same drone k , 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)–(19) connect variables \mathbf{x} and \mathbf{z} (Karaođlan 2015).

(vii) **Variable domain:**

$$x_{ij}^k \in \{0, 1\}, \quad q_{ij}^k, e_{ij}^k \geq 0 \quad \forall (i, j) \in A, \forall k \in K, \quad (20)$$

$$f_i^k \geq 0 \quad \forall i \in N, \forall k \in K, \quad (21)$$

$$y_i^k \in \{0, 1\}, \quad y_0^k \in \{0, 1, 2, \dots\} \quad \forall i \in N', \forall k \in K, \quad (22)$$

$$\tau_i^k \geq 0 \quad \forall i \in N^-, \forall k \in K, \quad (23)$$

$$z_{ij}^k \geq 0 \quad \forall i, j \in N', \forall k \in K. \quad (24)$$

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

$$\min \sum_{k \in K} \sum_{(i,j) \in A} (c_{ij} x_{ij}^k + \delta e_{ij}^k), \quad (25)$$

where e_{ij}^k is the energy consumption on arc (i, j) of drone k , and δ is the unit energy cost which includes the cost of energy and battery amortized based on drone operations. e_{ij}^k is computed as

$$e_{ij}^k = (W + m + q_{ij}^k)^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varsigma h}} / 1000 \times (t_{ij}/3600) x_{ij}^k. \quad (26)$$

Correspondingly, if the linear approximation method is used, e_{ij}^k will take the value $[(\alpha m + \beta) x_{ij}^k + \alpha q_{ij}^k] t_{ij}^k / 3600$.

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 which 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 6.3.

3.2.2 Formulation Without a Drone Index (2-index Formulation)

The drawback of the previous formulation is that the number of decision variables increases in proportion to the number of drones. To overcome this, we introduce a formulation that does not require a drone index. The formulation uses variables x, q, e, f, τ, z with the same notation as in the previous section, but drops the index k . It is as follows:

$$\min \sum_{(i,j) \in A} (c_{ij}x_{ij} + \delta e_{ij}) \quad (27)$$

$$\text{s.t.} \quad \sum_{j \in \delta^+(i)} x_{ij} = 1 \quad \forall i \in N', \quad (28)$$

$$\sum_{j \in \delta^-(i)} x_{ji} = 1 \quad \forall i \in N', \quad (29)$$

$$\sum_{j \in \delta^+(0)} x_{0j} = \sum_{j \in \delta^-(n+1)} x_{j,n+1}, \quad (30)$$

$$\sum_{i \in \delta^-(j)} q_{ij} - \sum_{i \in \delta^+(j)} q_{ji} = d_j \quad \forall j \in N', \quad (31)$$

$$q_{ij} \leq Qx_{ij} \quad \forall (i, j) \in A, \quad (32)$$

$$q_{i,n+1} = 0 \quad \forall i \in N', \quad (33)$$

$$f_0 = 0, \quad (34)$$

$$f_i + [(W + m + q_{ij})^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varsigma h}} / 1000] t_{ij} / 3600 \leq M_{ij}(1 - x_{ij}) + f_j \quad \forall (i, j) \in A, \quad (35)$$

$$f_{n+1} \leq \sigma, \quad (36)$$

$$\tau_i + t_{ij} - M''_{ij}(1 - x_{ij}) \leq \tau_j \quad \forall i \in N', \forall j \in N^-, \quad (37)$$

$$a_i \leq \tau_i \leq b_i \quad \forall i \in N^-, \quad (38)$$

$$\tau_i + (t_{i,n+1} + t_{0j}) \leq \tau_j + (1 - z_{ij})M'''_{ij} \quad \forall i, j \in N', i \neq j, \quad (39)$$

$$\sum_{\substack{i \in N' \\ i \neq j}} z_{ij} \leq x_{0j} \quad \forall j \in N', \quad (40)$$

$$\sum_{\substack{j \in N' \\ j \neq i}} z_{ij} \leq x_{i,n+1} \quad \forall i \in N', \quad (41)$$

$$\sum_{j \in N'} x_{0j} - \sum_{i \in N'} \sum_{\substack{j \in N' \\ j \neq i}} z_{ij} \leq |K|, \quad (42)$$

$$x_{ij} \in \{0, 1\}, \quad q_{ij}, e_{ij} \geq 0 \quad \forall (i, j) \in A, \quad (43)$$

$$f_i \geq 0 \quad \forall i \in N, \quad (44)$$

$$\tau_i \geq 0 \quad \forall i \in N^-, \quad (45)$$

$$z_{ij} \geq 0 \quad \forall i, j \in N'. \quad (46)$$

Constraints (28)–(46) are equivalent to (1)–(3), (5)–(7), (10)–(12), (14)–(18), (20)–(21), and (23)–(24), respectively. Constraints (42) limit the number of drones that can be used in the system. The big- M in this formulation takes the same values as those in the previous section. Similarly, constraints (34)–(36) are also tackled implicitly using the method introduced in Section 5.1.

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 functions (25) and (27)), respectively. If the exact method is applied for energy calculation, we use a subscript e . Similarly, a subscript a is used for the linear approximation method. Symbols $|k$ and $|nk$ indicate whether drone index k is used or not when building the model. For example, $R_e|nk$ represents the formulation without a drone index that minimizes the travel cost and uses the exact method for energy calculation.

4 Valid Inequalities

In this section, we introduce some valid inequalities to further strengthen our models. We first present those adopted from the literature, then introduce another group of valid inequalities based on the energy function.

Valid inequalities from the literature. We adopt the following equations for the 3-index formulation:

$$x_{ij}^k \leq y_i^k \quad \forall i \in N^+, j \in \delta^+(i), \forall k \in K, \quad (47)$$

$$\sum_{k \in K} \sum_{j \in N'} x_{0j}^k \geq \left\lceil \frac{\sum_{i \in N'} d_i}{Q} \right\rceil, \quad (48)$$

$$y_i^k \leq y_0^k \quad \forall i \in N', \forall k \in K, \quad (49)$$

$$\sum_{i=1}^j 2^{j-i} y_i^k \geq \sum_{i=1}^j 2^{j-i} y_i^{k+1} \quad \forall j \in N', \forall k \in K \setminus \{|K|\}. \quad (50)$$

Constraints (47) are logical inequalities (Coelho and Laporte 2013). Constraints (48) indicate the least number of trips needed to visit all the customers. Constraints (49) mean that if drone k does not leave the starting depot, it cannot visit any customers (Coelho and Laporte 2013). Equations (50) break the symmetry, which are the lexicographic ordering constraints (Sherali and Smith 2001, Adulyasak et al. 2013).

Valid inequalities (48) are also modified to enhance the 2-index formulation:

$$\sum_{j \in N'} x_{0j} \geq \left\lceil \frac{\sum_{i \in N'} d_i}{Q} \right\rceil. \quad (51)$$

Valid inequalities based on the energy function. Constraints (52) are derived from Equations (26) using the constant d_j to replace the variable q_{ij}^k , which yields linear equations and a lower bound of e_{ij}^k since $q_{ij}^k \geq d_j$ when $x_{ij}^k = 1$. Constraints (52) mean that if arc (i, j) is traversed by drone k , the energy consumption is at least equal to the value of the right hand side.

$$e_{ij}^k \geq (W + m + d_j)^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\sigma h}} / 1000 \times (t_{ij}/3600)x_{ij}^k \quad \forall (i, j) \in A, \forall k \in K. \quad (52)$$

Correspondingly, for the 2-index formulation, we have

$$e_{ij} \geq (W + m + d_j)^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\varsigma h}} / 1000 \times (t_{ij}/3600)x_{ij} \quad \forall (i, j) \in A. \quad (53)$$

5 Solution Method

In this section, we introduce the techniques for exact calculation of energy consumption, and develop B&C algorithms for our models.

5.1 Cuts for Nonlinear Energy Function

We describe the techniques for exact energy calculation for the 3-index formulation here. The 2-index formulation can apply the same techniques with minor modifications.

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 $\{0, i_1, i_2, \dots, i_{l-1}, i_l, n + 1\}$ associated with drone \bar{k} , we add the logical cut

$$x_{i_1 i_2}^{\bar{k}} + x_{i_2 i_3}^{\bar{k}} + \dots + x_{i_{l-1} i_l}^{\bar{k}} \leq l - 2. \quad (54)$$

Equation (54) means that the customer sequence is not allowed to be performed by drone \bar{k} . We can also extend it to any drone

$$x_{i_1 i_2}^k + x_{i_2 i_3}^k + \dots + x_{i_{l-1} i_l}^k \leq l - 2 \quad \forall k \in K. \quad (55)$$

However, our preliminary tests demonstrate that Equation (54) is more efficient.

Subgradient cut. In Equation (26), when x_{ij}^k is fixed ($x_{ij}^k \neq 0$), e_{ij}^k is a convex function in q_{ij}^k . Thus, when $\bar{x}_{ij}^k = 1$ the tangent line at point $(\bar{q}_{ij}^k, \bar{e}_{ij}^k)$ (we use a bar ‘-’ to represent known values) is

$$e_{ij}^k = \bar{e}_{ij}^k + \bar{\beta}_{ij}^k (q_{ij}^k - \bar{q}_{ij}^k), \quad (56)$$

where $\bar{\beta}_{ij}^k = \frac{3}{2}(W + m + \bar{q}_{ij}^k)^{\frac{1}{2}} \sqrt{\frac{g^3}{2\rho\varsigma h}} / 1000 \times (t_{ij}/3600)$, 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:

$$e_{ij}^k \geq \bar{e}_{ij}^k x_{ij}^k + \bar{\beta}_{ij}^k (q_{ij}^k - \bar{q}_{ij}^k) \quad \forall (i, j) \in A, \forall k \in K. \quad (57)$$

When $x_{ij}^k = 0$, the right-hand side of Equation (57) is a negative number ($q_{ij}^k = 0$ because of constraints (6)) and the cut is inactive. When $x_{ij}^k = 1$, the cut is added to guarantee that the energy consumption and cost are accurately computed.

Remarks: (i) Being different from the logical cuts, constraints (10) and (12) are necessary when applying the subgradient cuts. (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 correct calculation 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 them 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 (52) and (53).

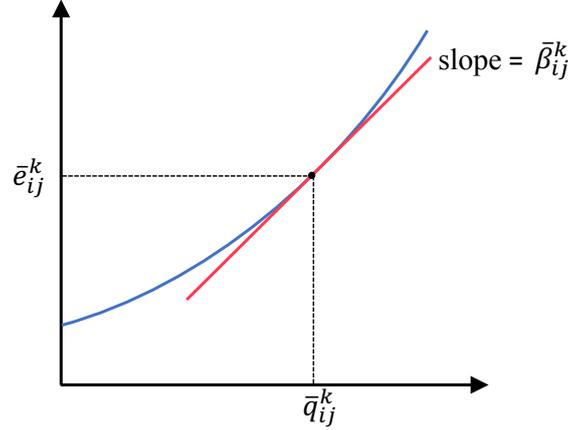


Figure 2: The tangent line of the energy function

5.2 Branch-and-Cut Algorithms

The B&C algorithm has been extensively used to solve MILP problems, which is a combination of a cutting plane method with a branch-and-bound (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.

5.2.1 Separation of Subtour Inequalities

Although constraints (5) and (31) can eliminate subtours, we introduce another group of subtour elimination constraints (SECs) which can help improve computational efficiency for the B&C scheme.

Formulation with a drone index. The SECs for the formulation with a drone index can be stated as follows (Gendreau et al. 1998, Fischetti et al. 1998):

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq \sum_{i \in S} y_i^k - y_l^k \quad \forall S \subseteq N', |S| \geq 2, \forall l \in S, \forall k \in K. \quad (58)$$

To identify the violated constraints, we use an exact separation algorithm which solves a minimum $s - t$ cut problem for each drone. We construct a graph for drone k with nodes where $\bar{y}_i^k > 0, \forall i \in N^+$ and set the capacity of edges as \bar{x}_{ij}^k . Then a set of minimum $s - t$ cut problems are solved by setting the starting depot as the source and each customer $i \in N' | \bar{y}_i^k > 0$ as the sink. A violated SEC is obtained if the value of the minimum cut is smaller than $2\bar{y}_l^k$. If a subtour is found for drone k , we add constraints (58) with $l = \arg \min_{i \in S} \{\bar{y}_i^k\}$ to the model. To solve the minimum $s - t$ cut problem, we use the Python networkx 2.0 package (<https://networkx.github.io/documentation/networkx-1.10/index.html>).

Formulation without a drone index. The SECs for the formulation without a drone index are as follows (Laporte 1986):

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - q(S) \quad \forall S \subseteq N', |S| \geq 2, \quad (59)$$

where $q(S) = \lceil \frac{\sum_{i \in S} d_i}{Q} \rceil$ 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).

5.2.2 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 because they are redundant for our models due to the fact that the subtours are eliminated by the customer demand constraints and it is time consuming to solve the separation problems at all nodes of the B&B tree.

6 Numerical Experiment

In this section, we present the instances and discuss our numerical tests. 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 single core of one node unless specified. The computing time limit is set to four hours.

6.1 Instance Sets

Here, 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). And the second set, named *Set B*, is an extension of Solomon’s instances, taking into account drones’ specific characteristics. We use Set A instances for preliminary tests and performance comparisons. After knowing the performance of the two formulations, we conduct experiments on Set B instances in Section 6.6. Given that Set B instances are an extension of the well-known Solomon’s instances, one can also evaluate the robustness and gain the insights regarding the performance of our approach. The details of the instances and solutions are available at the following URL: <https://sites.google.com/view/chengchun/instances>.

6.1.1 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 A₁*, the depots are located at the lower left corner of the region. For type 2 instances, named *Set A₂*, 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 Z th instance of Y customers in Set A_x .

According to Dorling et al. (2017), we set $\alpha = 0.217 \text{ kW/kg}$, $\beta = 0.185 \text{ kW}$, $m = 1.5 \text{ kg}$, $Q = 1.5 \text{ kg}$, $\sigma = 0.27 \text{ kWh}$, $g = 9.81 \text{ N/kg}$, $\rho = 1.204 \text{ kg/m}^3$, $\varsigma = 0.0064 \text{ m}^2$, $h = 6$, $\delta = 360 \text{ \$/kWh}$.

For both Set A_1 and Set A_2 instances, 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 $[0.1, 1.5]$. Here we do not generate all the demands from $[0.1, 1.5]$, because our preliminary tests show that if this generation method is used, for most trips only one customer is included because of the limited drone weight capacity. 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 h th numbers in

order, where $h = \lceil \frac{|N'|}{K} \rceil$ 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 A₂, 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 A₁.

6.1.2 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) for the 3-index formulation (constraints (37) and (39) for the 2-index formulation) when conducting our numerical tests. We also make some modifications to customers' demands to fit 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. Here, we set the battery energy capacity as $\sigma = 0.027 \text{ kWh}$, and keep other drone-related parameters the same, as given in Section 6.1.1.

6.2 Enhancement Strategy Evaluation and Model Comparison

This section analyzes the effect of valid inequalities and SECs, and also compares the 3-index and 2-index formulations. We conduct all the tests on instances with 10–30 customers in Set A. First, we only apply subgradient cuts to the models to evaluate the valid inequalities and SECs. After evaluating the performance of the 3-index and 2-index formulations, we further compare different implementations of cuts on the best formulation scheme. Tables 2–3 present the results. The third 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 models. *Opt* is the number of instances solved to optimality. *Unsol* denotes the number of instances, for which no feasible solution is generated under the time limit. Note that if there exist unsolved instances, the reported results are the average of the solved instances (i.e., an optimal or feasible solution is generated for the instance). *UP*, *LB*, *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.

From Table 2, for the 3-index formulation, more instances are solved to optimality when all the valid inequalities are applied. When the SECs are added, the algorithm can optimally solve two more instances for the $(R + E)_e|k$ model, and reduce the number of unsolved instances for the $E_e|k$ model. When valid inequalities (52) are added for the $E_e|k$ and $(R + E)_e|k$ models, the lower bound at the root node can be significantly improved, especially for the $E_e|k$ model.

From Table 3, for the 2-index formulation, different implementations of cuts yield different performances. In general, the simultaneous application of logical cuts and subgradient cuts,

Table 2: Average results with different valid inequalities and SECs for the 3-index formulation

		None	(47)	(48)	(49)	(50)	(52)	All	All+SECs
$R_e k$	Opt	22/50*	20/50	22/50	19/50	25/50	19/50	26/50	25/50
	Unsol	7/50*	5/50	5/50	7/50	3/50	7/50	6/50	4/50
	UB	7870.38	8053.94	8083.73	7882.41	8263.65	7954.25	7824.83	8035.41
	LB	7462.27	7548.35	7644.49	7438.44	7836.00	7451.99	7573.10	7722.08
	Gap	3.89	4.74	4.08	4.26	3.84	4.73	2.48	2.97
	CPU	8336.30	8685.40	8535.10	9014.17	7875.92	8742.56	6951.51	7460.86
	RLB	6971.74	7144.68	7200.88	6976.40	7330.72	7001.01	7153.02	7299.76
$E_e k$	Opt	11/50	13/50	10/50	11/50	13/50	18/50	21/50	21/50
	Unsol	14/50	6/50	14/50	10/50	11/50	9/50	4/50	3/50
	UB	519.70	578.53	525.79	555.76	542.41	541.81	571.60	586.01
	LB	217.19	228.29	220.28	208.74	221.34	221.34	512.26	517.75
	Gap	46.87	48.16	46.58	50.55	46.31	9.22	7.91	8.81
	CPU	10265.33	10674.74	10423.45	10658.98	9766.58	9000.03	8614.09	8911.76
	RLB	0.00	0.00	2.04	0.00	2.74	433.27	463.12	470.11
$(R + E)_e k$	Opt	18/50	18/50	18/50	17/50	19/50	18/50	23/50	25/50
	Unsol	6/50	7/50	7/50	9/50	7/50	9/50	4/50	5/50
	UB	8563.65	8476.04	8419.78	8163.00	8527.50	8220.80	8726.56	8555.89
	LB	7777.36	7713.21	7727.38	7523.64	7797.20	7761.73	8257.30	8161.45
	Gap	7.20	6.87	6.32	6.16	6.26	4.38	3.96	3.46
	CPU	9406.63	8763.15	9132.24	8988.13	8653.47	8739.88	7842.17	7230.65
	RLB	7064.39	7027.90	7020.14	6802.50	6996.78	7268.07	7809.29	7691.62

* indicates the number of instances (out of 50) that are solved to optimality or unsolved.

as well as all the enhancement strategies, gives the best performance for the three models. Specifically, more instances can be solved to optimality for the $R_e|nk$ and $E_e|nk$ models. For the $(R + E)_e|nk$ 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 and CPU time is relatively close. We can also observe that, the average lower bounds are significantly improved for the models with the energy cost (E_e) when applying the valid inequalities based on the energy function, i.e., constraints (52) and (53), for the 2-index and the 3-index formulation, respectively.

Tables 2–3 also show that the 2-index formulation outperforms the 3-index formulation in two aspects: (1) the 2-index formulation can solve more instances to optimality. The 3-index formulation can solve 26, 21, and 25 instances to optimality for the R_e , E_e , and $(R + E)_e$ models, respectively; however, the 2-index formulation can increase the corresponding number to 47, 46, and 46; (2) the 2-index formulation can generate good feasible solutions for all the instances in a shorter time. For the R_e and $(R + E)_e$ models, the average gap can be controlled under 0.4% by the 2-index formulation, regardless of whether enhancement strategies are used or not. For the E_e model, the comparison is even obvious. When Equations (52) or (53) are not used, the average gap is over 46% for the 3-index formulation while it is under 9% for the 2-index formulation. When these two valid inequalities are used, the average gap of the 3-index formulation (over 7%) is also dramatically larger than that of the 2-index formulation (under 0.6%). Therefore, in the following sections, we use the 2-index formulation, together with all the enhancement strategies and both logical and subgradient cuts, to perform our tests for each model.

6.3 Details of Solutions for Set A Instances With Size 10–30

Tables 4–5 give further details of results provided by the 2-index formulation. *Cust* is the number of customers. *Log*, *Sub*, and *SECs* are the number of generated logical cuts, subgradient cuts, and SECs, respectively. *UAVs* is the number of drones used. *Swap* represents the times of battery swap. *T/d* indicates the average number of trips performed by each drone. The last column in Table 5 is the proportion of energy cost to total cost. More detailed results for each instance are presented in Appendix (Table 12).

Table 3: Average results with different valid inequalities and SECs for the 2-index formulation

		Only subgradient					Only logical	Subgradient +logical
		None	(51)	(53)	(51)+(53)	(51)+(53) +SECs	(51)+SECs	(51)+(53) +SECs
$R_e nk$	Opt	46/50	45/50	43/50	45/50	46/50	46/50	47/50
	UB	8275.03	8275.68	8275.92	8277.65	8275.03	8275.03	8276.33
	LB	8258.96	8257.78	8252.39	8256.71	8260.45	8262.29	8265.28
	Gap	0.14	0.16	0.22	0.19	0.13	0.11	0.10
	CPU	1891.19	1879.87	2676.27	2163.63	1809.08	1598.88	1852.69
	RLB	7801.30	7837.34	7809.34	7831.27	7845.03	7855.32	7847.05
$E_e nk$	Opt	31/50	31/50	41/50	41/50	43/50		46/50
	UB	585.05	585.50	583.68	583.76	583.66		583.74
	LB	515.36	521.38	579.51	579.90	581.84	Not applicable	581.92
	Gap	8.83	8.06	0.54	0.50	0.22		0.22
	CPU	6277.92	6070.81	2979.98	3096.69	2862.01		2388.78
	RLB	168.74	155.80	510.49	506.82	512.79		512.32
$(R + E)_e nk$	Opt	44/50	43/50	45/50	45/50	46/50		46/50
	UB	8867.14	8871.10	8863.44	8864.42	8863.48		8866.82
	LB	8830.79	8825.42	8839.74	8843.10	8843.84	Not applicable	8845.15
	Gap	0.30	0.38	0.19	0.18	0.16		0.18
	CPU	3385.75	2675.13	2256.70	2301.36	2196.45		2236.96
	RLB	7966.44	7993.49	8317.81	8357.03	8374.24		8374.46

Table 4: Average results on cuts generated by the 2-index formulation for Set A instances with size 10–30

	Cust	$R_e nk$						$E_e nk$						$(R + E)_e nk$					
		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
Average		47/50	0.1	1852.7	2.2	441.6	32.4	46/50	0.2	2388.8	2.5	810.0	34.6	46/50	0.2	2237.0	2.0	770.0	33.3

Table 4 shows that the number of subgradient cuts is significantly 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. On average, the $R_e|nk$ model consumes the least computing time and the $E_e|nk$ model consumes the most computing time. Further, instances in Set A₁ require more time than those in Set A₂, which suggests that depot location influences the computational efficiency.

Table 5 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 $(R + E)_e|nk$ 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 objective functions, different schedules are indeed generated for some instances. An example is given in Table 6. It shows that two more trips are performed for the $E_e|nk$ model, leading to a greater travel distance and a lower energy consumption. In addition, we find that for this instance the schedules generated by the $R_e|nk$ and $(R + E)_e|nk$ 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 $(R + E)_e|nk$ model includes the energy cost in the objective, it can always guarantee that drones travel in directions with minimal energy consumption. Therefore, in realistic applications, for undirected networks, even though decision makers favor a VRP objective which minimizes the travel cost, they can still add the energy cost in the objective and set a small value for the energy price to

Table 5: Average results on drones generated by the 2-index formulation for Set A instances with size 10–30

	Cust	$R_e nk$			$E_e nk$			$(R + E)_e nk$			Energy cost (%)
		UAVs	Swap	T/d	UAVs	Swap	T/d	UAVs	Swap	T/d	
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	6.7
	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

save battery energy consumption and further reduce the recharging time.

Table 6: Schedules generated by different objectives for instance *Set_A2_Cust_15_2*

Trips	$R_e nk$		$E_e nk$		$(R + E)_e nk$	
	Trips	Energy (kWh)	Trips	Energy	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		1.5785		1.4834		1.5061
Total travel distance		7995.39		8153.26		7995.39

6.4 Performance Comparison Between Subgradient-based B&C and Linear Approximation Model

This section discusses and provides the insights regarding the differences in solutions generated by the exact models and the linear approximation models. Table 7 presents the statistical results, and detailed results for the linear approximation models are presented in Appendix (Table 13). For solutions generated by the linear approximation models, after obtaining the trips, we use equation (26) to calculate the actual value of the energy consumption for each trip and report the average results in the last two rows of Table 7. *Infeasible* is the number of instances for which the linear approximation models actually yield trips with violated energy constraints. *Energy gap* is the percentage difference in energy calculation, which is computed as $(actual\ value - linear\ value) / linear\ value \times 100$.

From Table 7, we get the following observations: (1) **Computational efficiency.** For both models, the computational efficiency of the approximation method is better than that of the exact method. By using the approximation method, more instances can be solved to optimality in a shorter time frame. However, the gap difference of the two methods is not significant. Therefore, we can conclude that, even though our original models are nonlinear, the use of logical and subgradient cuts is highly efficient and the performance does not deteriorate so much compared to the approximation models for which the structure is much simpler. (2)

Table 7: Statistics of solutions generated by the exact and linear approximation models

	model R		model $R + E$	
	Exact method	Linear approximation	Exact method	Linear approximation
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

Feasibility. In multiple instances, the approximation models yield infeasible trips which violate the energy capacity constraints. For the R and $R + E$ models, the approximation method produces infeasible trips for 20 and 18 instances, respectively. In addition, there is a significant gap in energy calculation, over 9%, between the two methods.

To further display the value of exact energy calculation, we give an example in Table 8 to show the different schedules generated by the exact and approximation models. It demonstrates that the first trip given by the two approximation models consumes 0.2925 kWh energy, which violates the battery’s energy capacity (0.27 kWh). However, if the linear approximation method is used, it will consider these trips as feasible ones. Therefore, it is necessary to use our proposed cuts for exact energy calculation.

Table 8: Details of solutions generated by the exact and approximation models for instance *Set_A1_Cust_25_2*

	Exact method		Linear approximation method			
	Trips	Energy consumption (kWh)	Trips	Linear approximation	Actual value	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

6.5 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 (37)–(39) and solve a multi-trip drone routing problem. Average results are reported in Table 9 and detailed results are presented in Appendix (Tables 14–15).

Table 9: Average results for exact models with tighter time windows and without time windows

Cust	model R						model E						model $R + E$						
	Tighter time windows			Without time windows			Tighter time windows			Without time windows			Tighter time windows			Without time windows			
	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
Average	48/50	0.1	985.6	43/50	0.2	2934.3	46/50	0.3	1774.7	44/50	0.2	2624.0	48/50	0.1	1037.9	44/50	0.2	2602.7	

From Table 4 and Table 9, when the time windows are tighter, one more instance can be solved to optimality for the R model, and two more instances for the $R + E$ model. Moreover, the average computation time has seen a significant decrease 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 branch-and-price algorithm is used for the MTRVPTW.

6.6 Algorithm Performance on Extended Solomon's Instances

In this section we test our algorithm on Set B instances which are extended from Solomon (1987). All the experiments are performed on 4 core processors with a 12-hour (43200 seconds) time limit. Summarized results are shown in Table 10 and detailed results on each instance are provided in Appendix (Table 16). In Table 10, column *Inst* is the instance label.

Table 10: Algorithm performance on Solomon's instances of type 2

Inst	25 customers						40 customers					
	model R		model E		model $R + E$		model R		model E		model $R + E$	
	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 , and this number decreases to 14 and 18 for model E and model $R + E$, respectively. The CPU time also varies widely, ranging from a few minutes to many hours.

In terms of computational performance as opposed to the MTRVRPTW 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 MTRVRPTW despite the fact that the original models are nonlinear and more complex. Here, we give some details about results presented in the literature where a MTRVRPTW is solved to provide some insights for our algorithms. Azi (2011) conduct their experiments with 72 hours allotted on an AMD Opteron 3.1 GHz with 16 GB of RAM. Their results suggest that when the trip duration limit is tight, for instances of 25 (40) customers, 15 (7) out of 27 are solved to optimality. When the trip duration constraint becomes less restrictive, only 14 and 4 instances are optimally solved, correspondingly. Hernandez et al. (2014) focus on generating feasible solutions in a shorter time and conduct their experiments on an Intel Core 2 duo 2.10 GHz with 2GB of RAM. Within a 2-hour time limit, for the case with a tight trip duration, 2 (0) out of 27 are solved to optimality for instances with size 25 (40). For the case with a longer trip duration, the corresponding number is 4 and 0. As a whole, the optimality gap ranges from 0.00% to 5.86% for their algorithm.

6.7 Results for Large Instances of Set A

Here, we report the results of Set A instances with 35–50 customers in Table 11. More detailed results are given in Table 17. 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. Our results show that all the previous instances, except *Set A1_Cust.30_5* for model E , are solved to optimality under the new experiment setting. The optimality gap of this instance for model E is 1.77%. We also solve instances with over 50 customers in Set A_1 and with over 55 customers in Set A_2 . The results show that for some instances no feasible solution is generated within the allotted time. Therefore, we do not report the results of these instances here. For some instances, when we directly solve the E model or the $R + E$ model, we find that the optimality gap is over 5% within the time limit, mainly resulting from the poor lower bound. Considering the R model is relatively easier than the other two models, for these specific instances, we first solve the R 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.

Table 11 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 model, 13 out of 35 instances are solved to optimality. For the E and $R + E$ 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 model as a start for the other two models. In particular, for the E model, 5 instances can be solved to optimality by using this method. We further use this idea to model E for Solomon’s r209, rc201, rc205, and rc206 instances with 40 customers (i.e., instances whose optimality gap is over 5% in Table 10). The results show that all these instances can be solved to optimality now.

7 Conclusions

This paper solves the multi-trip drone routing problem with time windows. Two formulations, one with a drone index and the other without, are introduced. B&C algorithms are developed to solve the formulations. We propose two types of cuts to tackle the nonlinear energy function. This strategy can always guarantee the feasibility of the given solutions and the exact calculation of energy cost. We further demonstrate the necessity and benefits of exact energy calculation as

Table 11: Results using multicore processors for Set A instances with 35–50 customers

		model R		model E		model $R + E$		
Cust	Inst	Gap	CPU	Gap	CPU	Gap	CPU	
Set A ₁	35	1	4.11	43200.0	4.55 [□]	43200.0	3.24 [□]	43200.0
		2	1.80	43200.0	3.58 [□]	43200.0	2.78	43200.0
		3	0.00	20642.0	0.00 [□]	21866.9	0.98	43200.0
		4	0.00	30214.8	0.00 [□]	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 [□]	43200.0
		4	0.44	43200.0	0.00 [□]	23208.1	0.39	43200.0
		5	0.73	43200.0	1.30 [□]	43200.0	2.32	43200.0
	45	1	4.24	43200.0	3.65 [□]	43200.0	3.96	43200.0
		2	2.15	43200.0	2.66 [□]	43200.0	2.06 [□]	43200.0
		3	1.51	43200.0	3.76	43200.0	2.46 [□]	43200.0
		4	3.05	43200.0	4.42 [□]	43200.0	3.15 [□]	43200.0
		5	1.95	43200.0	1.79 [□]	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 [□]	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 [□]	7308.4	2.16	43200.0
		5	5.05	43200.0	6.39 [□]	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 [□]	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 [□]	43200.0	2.36	43200.0	
	2	3.81	43200.0	2.24 [□]	43200.0	3.80	43200.0	
	3	2.92	43200.0	4.20 [□]	43200.0	2.72	43200.0	
	4	1.32	43200.0	3.29	43200.0	0.72 [□]	43200.0	
	5	1.80	43200.0	1.55 [□]	43200.0	2.01 [□]	43200.0	
Average		1.29	31276.1	1.50	34770.27	1.31	30896.4	

[□] We use the first feasible solution of the R model as an initial solution for this model.

opposed to the linear approximation used in the literature which results in either a suboptimal or infeasible solution. We generate the benchmark instances for this problem based on the instances used in the literature and conduct extensive numerical experiments to evaluate the effects of valid inequalities and user cuts. We also compare the formulations, which demonstrate that the 2-index formulation can solve more instances to optimality and provide good quality solutions. The effectiveness of the modeling scheme and the B&C algorithms are further confirmed by solving Solomon’s type 2 instances. Therefore, our work has filled important gaps in the literature concerning the formulations and the exact algorithms for the multi-trip drone routing problem with energy consumption.

Appendix. Detailed Results of Numerical Tests

Tables 12–17 present detailed results of our numerical tests, which are also available online.

Table 12: Results for Set A instances with 10–30 customers

Cust.	Inst	$R_c n_k$										$F_c n_k$										$(R + E) n_k$										
		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	
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	313.6	313.6	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	303.3	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	313.9	0	125	59	2	5	4557.8	4557.8	0.0	0.5	4555.4	0	77	43	2	5	
	4	4106.3	4106.3	0.0	1.2	3892.5	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	130	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	1.4	4349.3	0	187	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	7074.7	0.0	12.8	6742.0	0	316	38	3	7	
	2	4114.7	4114.7	0.0	6.2	3831.2	2	135	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	76.0	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	5105.3	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	7493.1	7492.8	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	
Set A1	20	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	8297.7	8297.7	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	595.5	595.4	0.0	84.9	481.9	4	805	48	3	9	7785.3	7784.5	0.0	983.3	7297.8	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	2	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	1193	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	560.6	11664.9	1	484	40	5	14	868.0	867.9	0.0	942.4	734.0	0	1113	39	5	14	12669.3	12668.1	0.0	1732.8	12361.7	1	1476	46	5	14	
	3	11530.4	11285.6	2.1	14400.0	10692.4	2	1243	34	5	14	831.8	807.8	2.9	14400.0	706.7	2	2226	35	5	14	12435.7	12077.9	2.9	14400.0	11081.0	2	1584	35	5	14	
	4	11670.9	11665.7	0.0	2699.8	11281.7	0	949	45	5	13	845.5	845.4	0.0	11981.8	755.7	1	1829	35	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	4	13	12690.4	12620.0	0.6	14400.0	11644.5	8	1560	60	5	13	
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	30	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	319.0	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	8	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	9	8674.8	8674.5	0.0	5.8	7978.3	1	365	29	3	9	
Set A2	20	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	11435.9	11435.0	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	549.4	0	428	42	3	10	9733.1	9733.1	0.0	27.1	9045.0	0	392	31	3	10	
	3	9434.0	9434.0	0.0	14.5	8046.6	2	240	43	3	11	632.9	632.9	0.0	21.1	577.3	2	342	41	3	11	10966.9	10966.9	0.0	13.4	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	4.0	395.0	1	438	44	4	11	9495.3	9495.3	0.0	22.0	9136.3	2	712	42	4	10	
	5	7590.7	7590.4	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	7865.1	3	764	31	3	9	
30	1	10667.2	10666.2	0.0	182.1	10044.2	1	394	32	4	13	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	13	
	2	11623.1	11622.4	0.0	78.7	11094.8	0	52	35	4	12	805.3	805.2	0.0	186.4	707.5	6	1110	27	4	12	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	10973.3	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									

Table 13: Results for linear approximation models

Cust	Inst	R_a [r/k]										$(R + E)_a$ [r/k]									
		UP	LB	Gap	CPU	RLB	SECs	UAV's	Trip	Viol*	Linear energy	Exact energy	UP	LB	Gap	CPU	RLB	SECs	UAV's	Trip	Viol
10	1	2930.4	2930.4	0.0	0.8	2791.5	12	2	5	F*	0.5204	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	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	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	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	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	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	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	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	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	7165.7	0.0	17.1	6635.4	29	2	7	T	1.2131	1.3246
Set A ₁	1	7720.1	7720.1	0.0	1.6	7675.0	42	4	11	F	1.4327	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	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	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	6652.9	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	7640.7	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	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	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	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	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	11097.5	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	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	12592.0	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	12289.7	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	12441.4	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	50	4	13	T	2.0783	11968.4	0.0	1573.8	11683.8	56	5	13	T	2.0724	2.2805
10	1	4668.6	4668.6	0.0	0.2	4475.2	4	2	5	F	0.8428	4972.0	0.0	0.2	4770.1	7	2	5	F	0.8428	0.9212
	2	7762.6	7762.6	0.0	0.1	5447.2	1	2	6	F	0.9694	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	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	6131.1	0.0	0.8	5584.6	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	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	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	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	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	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	8630.6	0.0	7.1	7957.6	26	3	9	F	1.4349	1.5575
Set A ₂	1	10684.6	10683.7	0.0	34.4	9689.7	41	4	13	F	1.8992	11365.6	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	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	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	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	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	11366.1	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	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	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	F	2.1005	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	48	4	14	T	2.0058	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	14909.4	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	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	11878.9	0.0	1065.3	11324.4	32	5	13	T	2.0414	2.2980
	4	10827.6	10826.6	0.0	291.7	10481.7	38	4	13	F	1.9436	11527.3	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	13050.0	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 14: Results for Set A instances with tighter time windows

Chest	Inst	$R_e nk$										$E_c nk$										$(R + E) nk$											
		UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Thp	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Thp	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Thp		
10	1	3135.3	3135.3	0.0	0.2	2952.4	0	26	17	2	2	5	7	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	7
	2	5499.8	5499.8	0.0	0.2	4679.7	0	72	12	2	2	5	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	5
	3	4175.2	4175.2	0.0	0.3	4118.3	0	68	19	2	5	5	5	304.2	304.2	0.0	0.2	291.5	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	5	5	296.6	296.6	0.0	0.6	291.2	1	120	55	2	5	4615.9	4615.9	0.0	0.3	4155.8	0	210	14	2	5
	5	4184.3	4184.3	0.0	0.1	4177.9	0	18	6	2	5	5	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	6836.2	6836.2	0.0	0.9	6335.3	3	393	30	3	7	7	469.8	469.8	0.0	5.2	426.1	2	540	26	3	7	7106.1	7106.1	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	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	5983.5	5983.0	0.0	1.0	4955.7	3	353	30	2	6	6	398.3	398.2	0.0	186.7	298.6	2	758	35	2	6	5981.9	5981.6	0.0	1.6	5292.6	2	441	32	2	6	
	4	5237.8	5237.6	0.0	1.3	4816.9	0	230	22	2	7	7	383.1	383.0	0.0	25.4	303.3	0	508	35	2	7	5920.9	5920.7	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	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	
Set A1	1	7295.5	7295.5	0.0	0.7	7717.7	0	42	24	4	11	11	559.0	559.0	0.0	3.5	536.5	0	219	36	4	11	8298.5	8298.5	0.0	0.8	8277.3	0	281	27	4	11	
	2	9258.7	9252.9	0.0	62.6	8741.4	0	565	37	4	10	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	8365.5	8363.5	0.0	16.7	8166.9	0	230	33	4	10	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	190.5	6049.3	1	728	20	3	9	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	6988.7	3	404	31	3	9	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	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	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	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	13	698.5	698.5	0.0	1636.5	591.9	4	1069	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	12	744.0	726.6	2.3	14400.0	614.2	4	1696	39	4	12	11255.9	11254.8	0.0	959.6	10455.0	2	1113	45	4	12	
30	1	9314.1	9313.2	0.0	1912.9	8936.0	1	1684	40	5	13	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	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	14	831.8	803.8	3.4	14400.0	691.7	1	1588	33	5	14	12452.0	12295.3	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	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	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	
10	1	5287.2	5287.2	0.0	0.1	5239.2	0	20	10	2	6	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	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	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	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	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	8	429.1	429.1	0.0	0.8	424.9	0	186	34	3	8	6927.3	6927.3	0.0	0.8	6420.4	0	115	21	3	8	
	2	7597.7	7597.7	0.0	1.0	7415.4	0	132	32	3	8	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	9	470.2	470.2	0.0	1.1	422.7	1	171	21	3	9	7296.4	7296.4	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	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	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	
Set A2	1	10542.8	10542.8	0.0	2.9	10055.5	3	378	22	4	12	12	732.4	732.4	0.0	2.8	672.8	2	461	34	4	13	11283.8	11283.8	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	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	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	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	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	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	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	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	13	812.7	812.6	0.0	26.5	759.8	0	703	34	4	13	12342.3										

Table 15: Results for Set A instances without time windows

Cust	Inst	R_e [mk]										E_e [mk]										$(R+E)$ [mk]									
		UP	LB	Cap	CPU	RLB	Log	Sub	SFCs	UAVs	Trip	UP	LB	Cap	CPU	RLB	Log	Sub	SFCs	UAVs	Trip	UP	LB	Cap	CPU	RLB	Log	Sub	SFCs	UAVs	Trip
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	4428.5	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	4009.0	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	1.9	276.6	0	355	21	2	5	4288.2	4288.2	0.0	0.4	4288.2	0	135	34	2	5
	4	3547.3	3547.3	0.0	0.4	3491.5	0	46	14	2	4	288.0	288.0	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	288.4	288.4	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	3	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	4883.1	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	5108.7	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
Set A1	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	1898	35	4	12
	2	7810.7	7809.9	0.0	11970.3	7365.2	10	1361	35	3	10	588.3	588.2	0.0	459.7	468.1	15	2333	46	3	10	8372.8	8372.0	0.0	818.1	8039.9	11	1390	48	3	10
	3	9228.6	9227.7	0.0	423.5	9084.6	1	784	28	4	10	688.2	688.2	0.0	600.7	574.6	3	1687	27	4	10	9888.3	9887.3	0.0	837.8	9731.1	3	1117	52	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	1064	28	4	13	11023.6	11023.9	0.0	869.9	9576.4	3	1418	56	4	13
	5	10383.3	10016.5	3.5	14400.0	9704.5	1	1281	47	4	12	734.5	719.2	2.1	14400.0	646.7	3	1644	36	3	11	11128.4	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	14400.0	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	11909.9	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	14400.0	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
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	489.9	489.9	0.0	6.9	457.3	2	302	26	2	8	7870.8	7870.5	0.0	21.4	7477.0	1	322	46	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	35	3	9	6454.6	6454.6	0.0	1.1	6363.8	0	206	44	3	9
	4	7111.2	7111.2	0.0	5.9	6863.5	0	32	25	2	7	479.6	479.6	0.0	4.4	450.6	0	311	37	2	7	7390.8	7390.8	0.0	7.3	7201.0	0	154	28	2	7
	5	8114.1	8113.6	0.0	14.7	7325.6	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
Set A2	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	24	3	10
	3	9334.0	9333.2	0.0	141.9	9018.6	7	362	33	3	10	644.5	644.5	0.0	118.0	619.6	5	1146	55	3	10	9975.5	9977.7	0.0	68.7	9633.2	5	559	34	3	10
	4	8858.5	8857.8	0.0	169.4	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	643.2	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	44	4	13	804.4	804.4	0.0	107.8	762.5	2	1020	39	4	14	12279.3	12278.1	0.0	84.9	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										

Table 16: Results for extended Solomon's instances of type 2

Cust	Inst	R_e, link										E_e, link										$(R + E)_e, \text{link}$									
		UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trips	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trips	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trips
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	3.4	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	503.9	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	502.1	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	295	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	27.1	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	84.6	788.6	0	678	44	4	12
	r206	749.5	749.5	0.0	83.5	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	75.7	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	780.7	0	1298	56	4	12
	r208	749.5	749.5	0.0	42.8	731.6	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	46.0	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	136.1	729.0	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	46.0	737.1	0	316	47	4	12	52.9	52.9	0.0	53.4	48.8	0	666	34	4	12	1025.4	1025.4	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	1010.2	1010.2	0.0	60.0	954.3	0	1074	30	4	12
rc202	940.6	940.6	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	1003.2	1003.2	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	1003.2	0.0	95.2	974.0	1	697	27	4	11	
rc204	934.4	934.4	0.0	5.4	931.6	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	100.6	981.6	0	1144	59	4	11	
rc205	946.7	946.7	0.0	63.4	933.8	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	1017.0	0.0	755.0	977.8	0	1478	45	4	12	
rc207	940.7	940.7	0.0	1253.2	924.3	0	1063	36	4	12	67.6	67.6	0.0	58.3	59.6	1	816	41	4	12	1008.4	1008.3	0.0	7396.9	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	1407.38	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	1031.1	0.0	3881.6	1011.9	0	1968	60	5	15	74.7	74.0	1.0	4320.00	63.4	0	4554	55	5	15	1105.9	1105.8	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	75.1	73.3	1.5	4320.00	60.8	0	3888	61	5	15	1106.2	1106.1	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	35	5	15	74.8	73.4	1.8	4320.00	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	74.5	74.0	0.6	4320.00	60.7	3	2665	53	5	15	1105.9	1105.8	0.0	8398.9	1074.9	0	2271	64	5	15
	c208	1031.1	1031.0	0.0	1996.7	1012.2	0	1990	49	5	15	74.5	74.0	0.6	4320.00	60.7	3	2665	53	5	15	1105.9	1105.8	0.0	348.1	1233.7	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	5754	40	6	19	1291.6	1291.6	0.0	348.1	1233.7	0	1687	37	6	19
	r202	1169.9	1169.9	0.0	1166.6	1121.2	4	1566	53	6	19	83.2	82.2	1.2	4320.00	71.1	9	1928	69	6	19	1237.0	1236.9	0.0	8183.1	1188.0	0	2305	38	6	19
	r203	1155.4	1155.4	0.0	3392.4	1110.4	0	774	42	6	19	81.5	81.5	0.0	17013.0	71.9	9	5734	40	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	1245.9	1245.9	1.2	43200.0	1177.2	0	3244	44	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	1567.2	74.9	0	2301	72	6	19</										

Table 17: Results using multicore processors for Set A instances with 35–50 customers

Cust	Inst	$R_e pk$										$(R + E) pk$																			
		UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip	UP	LB	Gap	CPU	RLB	Log	Sub	SECs	UAVs	Trip										
35	1	11719.2	1237.2	4.1	43200.0	10962.9	12	5471	54	5	13	846.2	807.7	4.6	43200.0	691.9	4	2734	39	5	13	12510.2	12104.5	3.2	43200.0	11693.5	0	3485	47	5	13
	2	12145.5	11927.6	1.8	43200.0	11495.5	9	3031	54	5	15	880.0	848.5	3.6	43200.0	735.6	4	3748	38	5	15	13025.5	12663.3	2.8	43200.0	12218.1	5	1695	50	5	15
	3	12346.4	12345.2	0.0	20642.0	11955.2	11	2866	37	5	15	889.5	889.4	0.0	21866.9	762.9	19	3273	37	5	15	13266.1	13106.4	1.0	43200.0	12706.7	10	2637	44	5	15
	4	12925.3	12924.1	0.0	30214.8	12528.6	5	3409	37	5	15	941.7	941.6	0.0	36895.7	808.6	20	4282	30	5	15	13268.2	13388.5	0.2	43200.0	13305.6	10	4865	40	5	15
	5	12384.3	12383.3	0.0	29126.0	11900.5	0	1144	48	6	18	894.5	894.4	0.0	15121.7	752.6	1	2687	36	6	18	13286.7	13285.7	0.0	20446.6	12709.6	1	1495	43	6	18
Set A1	1	14546.6	14055.3	3.4	43200.0	13792.5	4	1916	58	6	18	1050.1	1019.4	2.9	43200.0	899.9	6	5416	40	6	18	15546.0	14931.9	4.0	43200.0	14608.2	12	5103	53	6	18
	2	15741.8	15740.9	0.0	13947.5	15181.1	1	1370	46	7	20	1144.1	1133.8	0.9	43200.0	1036.8	3	3824	42	7	20	16887.0	16787.9	0.6	43200.1	16156.1	0	3204	55	7	20
	3	13290.2	12735.2	3.7	43200.1	12415.0	7	3905	52	6	17	960.9	915.4	4.7	43200.0	759.8	6	4500	63	6	18	14183.7	13648.1	3.8	43200.0	13238.9	3	4583	59	6	17
	4	15189.5	15122.6	0.4	43200.0	14560.9	16	3179	47	7	20	1099.0	1098.9	0.0	23208.1	984.3	14	5721	43	7	20	16292.1	16229.3	0.4	43200.0	15563.0	2	4024	64	7	20
	5	14534.9	14448.5	0.7	43200.0	14075.2	0	2293	46	6	19	1056.1	1042.4	1.3	43200.0	996.1	3	4015	59	6	19	15625.8	15264.1	2.3	43200.0	14965.9	3	4185	59	6	19
45	1	13832.5	13245.7	4.2	43200.1	12999.7	3	3065	64	6	18	991.7	955.6	3.6	43200.0	840.5	14	8605	37	6	19	14760.2	14175.2	4.0	43200.0	13817.3	4	5690	54	6	18
	2	18423.5	18026.8	2.2	43200.0	17692.1	4	1753	67	7	22	1343.0	1307.2	2.7	43200.0	1203.8	8	7610	36	7	22	19747.2	19340.7	2.1	43200.0	18942.6	3	7054	67	7	22
	3	17548.7	17283.6	1.5	43200.0	16809.4	1	1768	50	7	21	1284.5	1236.2	3.8	43200.0	1053.9	1	8810	50	7	22	18899.1	18484.2	2.5	43200.0	17974.2	1	4837	54	7	22
	4	15231.5	14767.5	3.0	43200.0	14429.8	10	3801	64	7	21	1104.8	1055.9	4.4	43200.0	928.9	29	8422	52	7	21	16341.1	15826.7	3.1	43200.0	15398.8	8	5617	49	7	21
	5	17467.0	17126.7	1.9	43200.0	16781.1	12	2274	65	8	23	1254.4	1232.0	1.8	43200.0	1116.2	31	6849	42	8	23	18736.6	18307.2	2.3	43200.0	17850.3	18	4315	56	8	23
35	1	13338.1	13000.9	2.5	43200.0	12340.2	11	1607	42	5	14	941.9	941.8	0.0	31873.4	804.1	29	4738	30	5	15	14287.9	13909.9	2.6	43200.0	13108.8	18	2912	38	5	14
	2	16285.8	16285.7	0.0	1755.4	15114.2	5	759	42	6	20	1158.5	1158.4	0.0	18306.0	986.1	5	2471	47	6	20	17450.5	17449.3	0.0	3397.5	16138.0	15	3099	32	6	20
	3	13709.3	13707.9	0.0	8732.7	13091.3	4	1814	45	5	14	982.4	954.6	2.8	43200.0	822.4	6	2328	33	5	14	14696.5	14695.1	0.0	11645.0	13828.6	9	3204	36	5	14
	4	16511.4	16510.1	0.0	9765.1	15300.8	9	1233	45	6	17	1171.3	1171.3	0.0	38648.4	997.3	13	2774	47	6	18	17694.9	17693.2	0.0	25076.3	16599.0	6	1778	37	6	17
	5	15706.0	15704.8	0.0	9491.5	14796.8	8	1676	48	5	16	1108.8	1108.7	0.0	42292.8	946.2	6	2742	34	5	16	16817.4	16815.8	0.0	18041.8	15740.4	9	2594	43	5	16
40	1	15866.1	15865.0	0.0	32162.3	15050.8	2	813	47	6	19	1131.8	1131.7	0.0	6219.5	982.1	7	1830	35	6	20	17008.8	17007.3	0.0	21628.1	16050.1	10	3197	49	6	19
	2	16751.9	16563.7	1.1	43200.0	15858.9	4	1167	36	6	18	1189.6	1172.6	1.4	43200.0	1010.3	3	3376	37	6	18	17954.7	17952.9	0.0	41897.4	16900.3	12	2304	50	6	18
	3	16867.9	16866.9	0.0	3298.6	16156.5	9	1854	50	7	20	1202.4	1202.3	0.0	30554.3	1023.7	2	1955	47	7	21	18085.0	18083.4	0.0	495.7	17039.1	6	3058	50	7	20
	4	17327.7	16959.0	2.1	43200.0	16506.2	14	1771	52	6	19	1233.5	1233.4	0.0	7308.4	1111.1	10	3392	25	6	20	18565.5	18164.3	2.2	43200.0	17526.7	6	5158	45	6	20
	5	12989.9	12333.8	5.1	43200.0	11903.8	19	3308	41	5	15	920.0	861.2	6.4	43200.0	763.2	17	3425	42	5	17	13886.7	13218.2	4.8	43200.0	12651.4	24	4235	39	5	16
Set A2	1	17395.5	17393.9	0.0	6142.2	16659.3	9	2285	50	8	23	1242.9	1242.9	0.0	1802.5	1076.3	1	2044	50	8	24	18602.0	18600.4	0.0	8093.6	17684.0	6	2824	41	8	23
	2	18299.8	18298.0	0.0	41018.0	17418.5	17	1752	66	7	23	1288.6	1271.8	1.3	43200.0	1085.5	14	4237	36	7	23	19596.7	19337.3	1.3	43200.0	18204.9	14	2576	45	7	22
	3	18854.9	18627.9	1.2	43200.0	18056.7	8	1878	44	8	23	1355.5	1326.3	2.2	43200.0	1169.6	5	3513	38	8	23	20214.0	20212.5	0.0	7452.0	19205.3	2	2111	38	8	23
	4	16136.7	15830.2	2.0	43200.0	15187.7	7	3575	38	6	19	1142.6	1121.5	1.8	43200.0	979.6	6	4761	31	6	19	17322.9	16859.6	2.7	43200.0	16154.7	7	2342	38	6	19
	5	22695.8	22693.7	0.0	37956.5	21613.5	5	1321	57	8	26	1617.5	1583.3	2.1	43200.0	1403.6	3	4321	50	8	26	24318.9	24075.2	1.0	43200.0	22998.6	3	3652	46	8	26
50	1	23045.5	22630.7	1.8	43200.0	22117.4	17	3932	43	8	26	1649.2	1637.1	0.7	43200.0	1513.3	6	5118	43	8	26	24723.8	24141.0	2.4	43200.0	23646.8	5	3907	51	8	26
	2	20734.0	19945.0	3.8	43200.0	19248.1	16	3064	39	8	24	1461.9	1429.1	2.2	43200.0	1271.5	26	5673	30	8	25	21947.0	21112.5	3.8	43200.0	20476.4	14	4361	43	8	24
	3	18482.6	17942.6	2.9	43200.0	17595.5	3	3278	47	7	20	1304.5	1249.7	4.2	43200.0	1149.9	4	4313	36	7	21	19707.0	19170.5	2.7	43200.0	18708.9	3	4128	40	7	21
	4	18537.6	18293.1	1.3	43200.0	17778.5	5	1269	44	7	23	1309.1	1266.1	3.3	43200.1	1128.8	5	2372	52	7	24	19849.8	19707.0	0.7	43200.0	19078.5	2	4748	46	7	23
	5	22160.8	21761.2	1.8	43200.0	20922.1	3	1607	43	8	25	1578.2	1553.6	1.6	43200.0	1454.6	8	3312	35	8	26	23729.1	23252.5	2.0	43200.0	22417.4	0	1800	46	8	25

References

- Adulyasak Y, Cordeau JF, Jans R (2013) Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing* 26(1):103–120.
- Agatz N, Bouman P, Schmidt M (2018) Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, ePub ahead of print, April 6, <https://doi.org/10.1287/trsc.2017.0791>.
- 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 JY (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 (2017) Dynamic programming approaches for the traveling salesman problem with drone. *ERIM Report Series Research in Management (ERS-2017-011-LIS)*.
- Campbell JF, Sweeney D, II ZJ (2017) Strategic design for delivery with trucks and drones. Technical Report.
- Carlsson JG, Song S (2017) Coordinated logistics with a truck and a drone. *Management Science*, ePub ahead of print, October 23, <https://doi.org/10.1287/mnsc.2017.2824>.
- Cattaruzza D, Absi N, Feillet D (2016) Vehicle routing problems with multiple trips. *4OR* 14(3):223–259.
- Cattaruzza D, Absi N, Feillet D, Vidal T (2014) A memetic algorithm for the multi trip vehicle routing problem. *European Journal of Operational Research* 236(3):833–848.
- Choi Y, Schonfeld PM (2017) Optimization of multi-package drone deliveries considering battery capacity. *96th Annual Meeting of the Transportation Research Board, Washington, DC (Paper No. 17-05769)*.
- Chowdhury S, Emelogu A, Marufuzzaman M, Nurre SG, Bian L (2017) Drones for disaster response and relief operations: a continuous approximation model. *International Journal of Production Economics* 188:167–184.
- Coelho LC, Laporte G (2013) A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research* 51(23-24):7156–7169.
- Dayarian I, Savelsbergh M, Clarke JP (2017) Same-day delivery with drone resupply. Optimization Online, http://www.optimization-online.org/DB_FILE/2017/09/6206.pdf.
- Desaulniers G, Madsen OB, Ropke S (2014) Chapter 5: The vehicle routing problem with time windows. *Vehicle Routing: Problems, Methods, and Applications, 2nd edition*, 119–159 (SIAM, Philadelphia).
- Dorling K, Heinrichs J, Messier GG, Magierowski S (2017) Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47(1):70–85.
- Ferrandez SM, Harbison T, Weber T, Sturges R, Rich R (2016) Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management* 9(2):374–388.
- Fischetti M, Gonzalez JJS, Toth P (1998) Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing* 10(2):133–148.
- Fleischmann B (1990) The vehicle routing problem with multiple use of the vehicles. Working Paper, Fachbereich Wirtschaftswissenschaften, Universität Hamburg.
- Gendreau M, Laporte G, Semet F (1998) A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks* 32(4):263–273.
- Ha QM, Deville Y, Pham QD, Hà MH (2015) Heuristic methods for the traveling salesman problem with drone. ArXiv preprint arXiv: 1509.08764.
- Ham AM (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.
- Karaođlan 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.
- Laporte G (1986) Generalized subtour elimination constraints and connectivity constraints. *Journal of the Operational Research Society* 37(5):509–514.
- Leishman GJ (2006) *Principles of helicopter aerodynamics, 2nd edition* (Cambridge university press, New York).
- 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 AN, Eglese RW (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 JV, 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 SL, Waslander SL (2015) Planning paths for package delivery in heterogeneous multi-robot 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 JE (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. *IIE Annual Conference. Proceedings*, 235–240 (Institute of Industrial and Systems Engineers (IISE)).
- Murray CC, Chu AG (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, 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 LDP, Guerriero F (2017) Last-mile deliveries by using drones and classical vehicles. *International Conference on Optimization and Decision Science*, 557–565 (Springer).
- Reddy TB (2010) Chapter 14: Lithium primary batteries. *Linden’s Handbook of Batteries, 4th edition*, 14.1–14.90 (McGraw-Hill Professional, New York).
- Rose C (2013) Amazons jeff bezos looks to the future. Accessed June 18, 2018, <https://www.cbsnews.com/news/amazons-jeff-bezos-looks-to-the-future/>.
- San KT, Lee EY, Chang YS (2016) The delivery assignment solution for swarms of uavs dealing with multi-dimensional chromosome representation of genetic algorithm. *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual*, 1–7 (IEEE).
- Savelsbergh M, Van Woensel T (2016) 50th anniversary invited article city logistics: Challenges and opportunities. *Transportation Science* 50(2):579–590.
- Sherali HD, Smith JC (2001) Improving discrete model representations via symmetry considerations. *Management Science* 47(10):1396–1407.
- Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research* 35(2):254–265.
- Sundar K, Rathinam S (2014) Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots. *IEEE Transactions on Automation Science and Engineering* 11(1):287–294.

- Ulmer MW, Thomas BW (2017) Same-day delivery with a heterogeneous fleet of drones and vehicles. Technical report.
- 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.