Vehicle Dispatching and Scheduling Algorithms for Battery Electric Heavy-Duty Truck Fleets Considering En-route Opportunity Charging

Zhouqiao Zhao
College of Engineering – Center for Environmental Research Technology
University of California, Riverside
Riverside, CA, USA
zzhao084@ucr.edu

Guoyuan Wu
College of Engineering – Center for Environmental Research Technology
University of California, Riverside
Riverside, CA, USA
gywu@cert.ucr.edu

Kanok Boriboonsomsin
College of Engineering – Center for Environmental Research Technology
University of California, Riverside
Riverside, CA, USA
kanok@cert.ucr.edu

Aravind Kailas
Volvo Group North America
Costa Mesa, CA, USA
aravind.kailas@volvo.com

***Abstract*—There has been growing interest in the electrification of medium- and heavy-duty vehicles (M-HDVs) in real-world, regional distribution applications. Fleet dispatch optimization of battery-electric trucks (BETs) is critical given the limited onboard energy, charging characteristics, and operational considerations. Our paper proposes a bi-level hierarchical method to optimize BET dispatch during pickup and delivery runs. With any route/scheduling change, the average speed, travel time, and energy consumption from one location to another will change accordingly because of the weight of the goods and the real-time traffic condition. So, the "electric vehicle routing problem" was extended to include pickup and delivery, time windows, and partial recharge. The proposed algorithm significantly reduces the operation cost of the BET fleet considering labor, energy consumption, and time window penalties without compromising computational efficiency.**

Keywords—Vehicle Routing Problem (VRP), Ant Colony Optimization (ACO), Fleet Management, Battery Electric Truck (BET)

I. INTRODUCTION

A. Motivation and Objectives

Over the last decade, significant progress has been made on vehicle fleet electrification, especially for light-duty vehicles. Recently, there have been increasing interest in electrifying medium-duty and heavy-duty vehicles. For instance, heavy-duty battery-electric trucks (BETs) have been successfully demonstrated in drayage application. Now, there are efforts to demonstrate the use of these trucks in regional distribution and other applications.

BETs operating in the regional distribution application will return to home base daily and can be charged overnight. However, due to their limited range and long charging time,

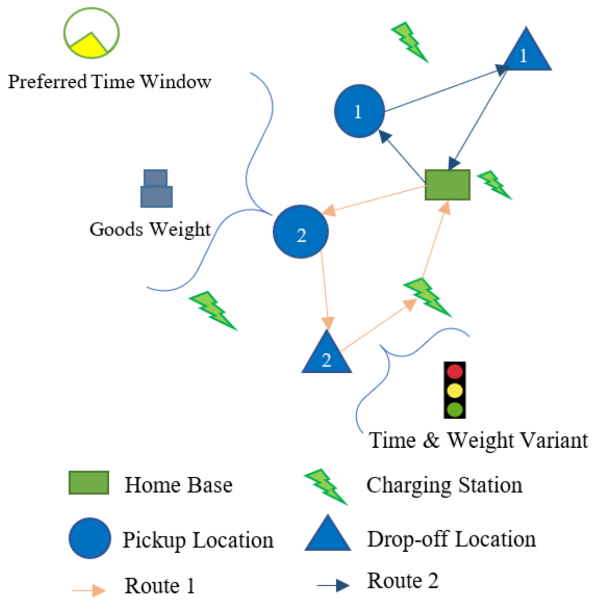

Figure 1. Illustration of the problem.

care must be taken when planning the routes and schedules of these trucks. The fleet must ensure that the route distance does not exceed the expected range of the BET. If it is necessary to assign a BET to serve a route with distance longer than the expected range, an en-route charging station needs to be identified and the required charging time needs to be included in the schedule. Similarly, if it is necessary to assign a BET to serve multiple routes in the day, required charging times between consecutive routes need to be included in the schedule to ensure that the BET can return to the depot before its battery is fully depleted.

B. Literature Review

Originated from the traveling salesman problem (TSP), the problem of dispatching vehicles for delivery of items has been investigated for decades in the field of operation research. TSP, an NP-hard problem in combinatorial optimization, aims to find the shortest distance loop among multiple locations. Despite the complexity, many existing methods are able to solve the problem efficiently, such as [1, 2]. Later, as an extension of TSP, the vehicle routing problem (VRP) is also well studied. Instead of a single salesman searching an optimal route, VRP considers the routing of multiple vehicles cooperatively, which is very close to our routing and scheduling problem. Therefore, in this section, we briefly review the studies by firstly combing the variations of VRP, and then introducing the major methods of solving VRP.

The majority of the real-world logistics problems are often more complex than the classical VRP [3]. Therefore, studies of VRP usually try to extend the classical VRP by adding further constraints. For instance, the capacitated vehicle routing problem (CVRP) is defined by adding the carrying capacity limitation of the vehicles [4], and the vehicle routing problem with time windows (VRPTW) is defined by adding the scheduled time window of each customer [5]. Also, VRP with pickup and delivery (VRPPD) studies the scenario that customers may request services with pickup and delivery [6]. With the electrification of the vehicle fleet, more and more research turns to the topic of the "electric vehicle routing problem" (E-VRP) and its extensions. For E-VRPs, the possibility of recharging at available charging stations is considered such as [7]-[9]. However, the charging strategy enormously increases the complexity of the problem, and a certain level of simplification is conducted in the existing works.

Considering the complexity of the VRP, an essential goal of the algorithm design is to balance the computation time and the optimality of the solution. The metaheuristic algorithms and heuristic algorithms are most commonly used to save the computational load. For example, Bard et al. proposed a branch and cut algorithm [10] to solve the VRP formulated with a mixed-integer linear programming problem. An adaptive large neighborhood search (ALNS) algorithm was proposed by Keskin et al. for the E-VRP with partial recharging strategies [11]. Simulated annealing (SA) and genetic algorithm (GA) approaches were applied by Omidwar et al. to minimize travel distances, time, and emissions [12]. Rizzoli et al. utilized ant colony optimization (ACO) and validated the performance with real-world data [13]. Combining GA, large neighborhood search (LNS), local search and dynamic programming (DP), Hiermann et al. proposed an algorithm for the problem with the mix of internal-combustion engine vehicles (ICEVs), electric vehicles (EVs), and plug-in hybrid electric vehicles (PHEVs) [14].

C. Organization of the Paper

The remainder of this paper is organized as follows. Section II formulates the mathematical problem, Section III describes the bi-level hierarchical method, Section IV presents the simulation-based analysis, and Section V wraps up the paper with the concluding remarks.

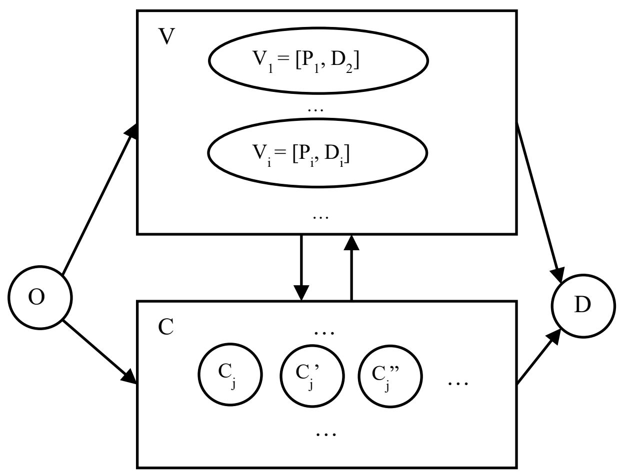

Figure 2. Graph model (the arcs within set V and set C are omitted, the arcs between set V and set C are simplified).

II. PROBLEM FORMULATION

The problem concerned in this study is illustrated in Figure 1. On each operation day, a set of customers would schedule services for pickup and delivery as shown in blue circles and triangles. The same indices of the locations denote the request from an identical customer. The green box denotes the home base of a fleet of BETs, where a charging station is deployed at the home base to ensure the overnight charging. The BETs should start from and return to the home base within the working hours. Also, there could be a few charging stations located around the operation region. To extend the range of BETs, opportunistic charging is allowed. Also, the opportunistic charging is set to be flexible to support the partial charging strategy. It should be noted that with the change of route and schedule, the average speed, travel time, and energy consumption from one location to another will change accordingly because of the weight of the goods and the real-time traffic condition. The proposed problem setup is defined as the "electric vehicle routing problem with pickup and delivery, time windows, and partial recharge" (EVRP-PD-TW-PR). If we consider all the key locations as nodes and the routes between nodes as bidirectional links, we can construct a graph model as shown in Figure 2.

We define node *O* to be the home base where the BETs departure from, while node *D* denotes the same home base where the BETS return. i is an index of a customer, and Vi is a set that contains the pickup and delivery nodes of customer *i*. The union of all customers set is named set *V*. *j* is an index of a charging station, and *Cj* denotes the charging station *j*. Also, the charging stations have multiple dummy notes marked by prime as each charging station can be visited multiple times. The union of all charging states with their dummy notes is named set *C*. The goal of the study is to generate itineraries for the BETs including the specific routes and schedules that could make the whole fleet of vehicles operating in an optimal or near-optimal manner.

Next, to formulate the optimization problem mathematically, the variables are defined in TABLE I. The variables can be categorized into three types including system input, intermediate variables, and decision variables. The system inputs are the variables that are given before solving the problem. The intermediate variables are those who would be updated while searching for the solution. The decision variables determine the final itineraries for the fleet of the BETs.

TABLE I. VARIABLE DEFINITIONS

Type	Variable	Name	Description
System Input	M	Number of BETs	Number of BETs
	k_i	Node type	1 if charging station, 0 if customer
	q_i	Cargo weight	Positive if pickup, negative if delivery
	L_i	Loading/ unloading time	Time spend at node i
	$TW_i = [e_i, l_i]$	Time window	Scheduled time by customer *i*
	W_m	BET capacity	The maximum carrying capacity for BET *m*
	B_m	Maximum SOC	The maximum battery capacity for BET *m*
	T_O	Earliest departure time	The earliest time the BETs can leave from the home base
	T_D	Latest return time	The latest time the BETs should return to the home base
	R_c	Charging cost rate	Cost for recharging in $/kWh
	R_h	Labor cost rate	Salary for the drivers, $/hr.
	R_{ei}	Early penalty rate	Cost rate in $/hr., if vehicle arrive earlier than the scheduled time by customer *i*
	R_{li}	Late penalty rate	Cost rate in $/hr., if vehicle arriver later than the scheduled time by customer *i*
Intermediate Variable	τ_{im}	Arrival time	Arrival time for BET *m* at node *i*
	w_{im}	Current load	Current load for BET *m* at node *i*
	y_{im}	Current SOC	Current SOC for BET m at node i
	t_{ijm}	Travel time	Estimated travel time for BET *m* from node *i* to node *j*
	E_{ijm}	Energy consumption	Estimated electricity consumption for BET *m* from node *i* to node *j*
Decision Variable	τ_{0m}	Actual departure time	The actual time the BET *m* leaves from the home base
	x_{ijm}	Node level route	0 if the route from *i* to *j* is not visited by BET, 1 otherwise
	p_{im}	Charging Rate	Three values for slow, regular, and super charging
	Y_{im}	Finish charging SOC	The SOC when BET *m* leaving charging station *i*

The optimization problem is defined by the equations (1)-(19), aiming to minimize the overall operation cost of one day. The objective function has three parts, including energy consumption cost, labor cost, and time window penalties. Though, the time window constraints are formulated as soft constraints. Constraint (2) enforces the connectivity of the costumers and constraint (3) assures the connectivity of the charging stations. Constraints (4) and (5) make sure all the BETs should depart from and return to the home base. Constraint (6) defines the conservation law that guarantees the number of incoming arcs to each node equal to the number of outgoing arcs. Constraint (7) makes sure that one customer's pickup and delivery requests are served by an identical BET, and the pickup should finish before the delivery for the same customer. (8)-(10) regulate time-related constraints. Among them, constraint (8) defines how the intermediate variable τ_{im} updated while searching. (9) and (10) narrow down the working hours. Constraint (11) defines the update logic for the BETs' loading status, and constraint (12) and (13) specify the initial cargo weight and the cargo weight limit while running. Similarly, (14)-(18) are SOC-related constraints. It should note that p_{im} is a three-value discrete variable which indicates the charging rate according to three different charging types. Finally, we define x_{ijm} as a binary variable to indicate the selection of the routes.

$$\min \sum_{i\in(O\cap V\cap C), j\in(D\cap V\cap C), m\in M} E_{ijm}x_{ijm}R_C + \sum_{i\in(O\cap V\cap C), j\in(D\cap V\cap C), m\in M} t_{ijm}x_{ijm}R_h + \sum_{i\in C, j\in(V\cap C), m\in M} (Y_{im}-y_{im})/p_{im}\cdot x_{jim}R_h + \sum_{i\in V, m\in M}[R_e\cdot max(0, e_i-\tau_{im}) + R_l\cdot max(0,\tau_{im}-l_i)], \quad (1)$$

subject to:

a) Graph Constraints

$$\sum_{j\in(D\cap V\cap C), m\in M} x_{ijm} = 1, \forall i\in(O\cap V) \quad (2)$$
$$\sum_{j\in(D\cap V\cap C)} x_{ijm} \le 1, \forall i\in C, m\in M \quad (3)$$
$$\sum_{j\in(V\cap C)} x_{ijm} = 1, \forall i\in O, m\in M \quad (4)$$
$$\sum_{i\in(V\cap C)} x_{ijm} = 1, \forall j\in D, m\in M \quad (5)$$
$$\sum_{i\in(O\cap V\cap C)} x_{ijm} = \sum_{q\in(D\cap V\cap C)} x_{jqm}, \forall j\in(V\cap C), m\in M \quad (6)$$
$$\sum_{i\in(O\cap V\cap C)} x_{iP_sm} = \sum_{j\in(V\cap C)} x_{jD_sm}, \forall V_s\in V, s\in V, m\in M \quad (7)$$

b) Time constraints

$$\tau_{jm} = (\tau_{im} + t_{ijm} + (1-k_i)s_i + k_i\cdot\frac{Y_{im}-y_{im}}{p_{im}})x_{ijm}, \forall i\in(O\cap V\cap C), m\in M \quad (8)$$
$$\tau_{Om} \ge T_O \quad (9)$$
$$\tau_{Dm} \le T_D \quad (10)$$

c) Load constraints

$$u_{jm} = (w_{im}+q_i)x_{ijm}, \forall i\in(O\cap V\cap C), m\in M \quad (11)$$
$$u_{Om} = 0 \quad (12)$$
$$0 < u_{im} \le W_m \quad (13)$$

d) SOC constraints

$$y_{jm} = ((1-k_i)\cdot y_{im} + k_i\cdot Y_{im} - E_{ijm})\, x_{ijm} \quad (14)$$
$$0 < y_{im} \le B_m \quad (15)$$
$$y_{im} < Y_{im} \le B_m \quad (16)$$
$$p_i = p_1 \; or \; p_2 \; or \; p_3 \quad (17)$$
$$y_{Om} = B_m \quad (18)$$

e) Variable constraints

$$x_{ijm} = 0 \; or \; 1 \quad (19)$$

III. Methodology

VRP is NP-hard and computationally challenging to find optimal or near-optimal solutions [5]. This is because during real-world dispatch operations, there are many continuous decision variables such as partial charging and flexible departure times. To solve this in a computationally efficient manner, a bi-level hierarchy method has been proposed in the paper. In the first level, continuous variables are frozen (i.e., discretized) to narrow down the searching space of the problem. Thereafter, a metaheuristic method, the Ant Colony Optimization (ACO) algorithm (first proposed by [1]), is applied to calculate a near-optimal solution. In the second level, the variables that were frozen in the previous level are fine-tuned around the near-optimal solution from ACO.

A. Coarse Scheduling

Coarse scheduling is done using the ACO algorithm. Inspired by the ant foraging behavior that the optimal path (usually shortest) can be gradually constructed by the convergence of the ant pheromone trail, the algorithm is used to find the optimal path along a graph. When a group of ants forage, they explore randomly at the beginning. The ants lay pheromone along the path they walk once they find paths to food, and the pheromone on the path can arouse the interest of other ants and increase the probability of exploring that path. This positive feedback loop will eventually result in a path with the highest quantities of pheromone, and such a path is near-optimal.

The ACO algorithm has been proved to be a very efficient algorithm to solve VRP [13]. However, departure time and recharging strategy are the two continuous decision variables that extraordinarily increases the searching efforts. So, the departure times were discretized into multiple instances within a fixed interval, e.g. 8 AM, 10 AM, and so on. Additionally, the charging strategy is set to charge back to 100% battery capacity and does not change. Then, the ACO algorithm is iteratively applied a relatively small data set as described above.

The ACO algorithm has two major stages at each iteration - searching and updating. Initially, a number of ants are released to "explore the graph" randomly. During searching a feasible searching space is determined based on the current locations and the values of the intermediate variables. Then, the possibilities of visiting the feasible nodes to be explored can be calculated based on the existing pheromone level defined by (20).

$$p(i, j) = \frac{\tau(i, j)^\alpha \eta(i, j)^\beta}{\sum \tau(i, j)^\alpha \eta(i, j)^\beta} \quad (20)$$

where (i, j) depicts the arc from node i to node j; $\tau(i, j)$ is the current pheromone value of this arc; and $\eta(i, j)$ is a heuristic term defined by the *a priori*. For example, it can be defined as $1/distance(i,\ j)$ to attract ants searching the closest node. α and β are two variables to balance the importance between the pheromone and the heuristic term. The denominator is applied to normalize the overall value from 0 to 1. Based on these visitation possibilities, each ant decides its next visiting node using the roulette strategy.

Once the solution for the current iteration obtained, the algorithm goes to the updating stage. Based on the decisions, we update the values of the intermediate variables and the pheromone level. The pheromone updating strategy shows as follows:

$$\Delta\tau = \sum_{1}^{q} Q/cost \tag{21}$$

$$\tau(t+1) = (1-\rho)\tau(t) + \Delta\tau \tag{22}$$

where the cost in equation (21) is the objective of our optimization problem defined as (1); Q is the predefined pheromone increasing rate; q is the number of ants that find the feasible route in current iteration; ρ is the evaporation rate, which is applied to avoid the searching process being trapped into local optima. To increase the convergence speed of getting the near-optimal solution, we always keep one *elite* ant (i.e., the one with the minimum cost in the previous iteration) to the next iteration of the searching process.

B. Fine Scheduling

From the Cooperative Dispatching, Routing, and Coarse Scheduling level, we get rough itineraries for each BET. However, because we simplify the problem in the previous level, the flexibilities of the departure time and the recharging strategy are lost. The fine scheduling process is used to recover the flexibilities by iteratively searching for a better solution to approximate the optima. Figure 3 illustrates such an idea in the travel distance-time plot.

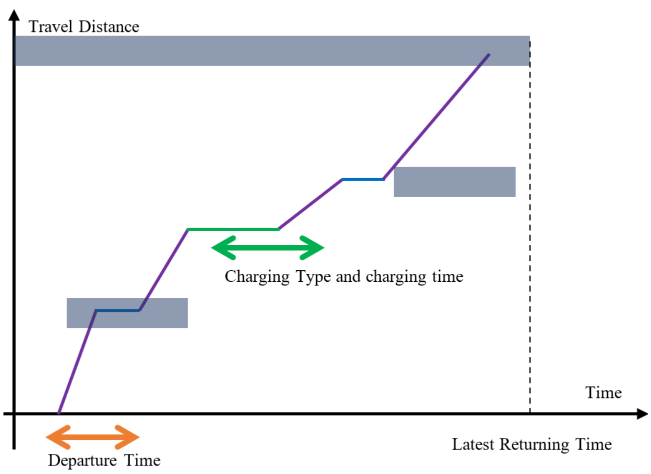

Figure 3. Illustration of fine scheduling process.

In the figure, an example trajectory of a BET calculated from the ACO algorithm is depicted, where only one customer is served. The gray bars in the figure depict the time window constraints including the scheduled time for the customer's pickup and delivery as well as the return time. The blue segments denote that the BET stops at the pickup or delivery location. The green segment denotes the process of recharging. As can be observed, by moving the departure time or changing the charging strategy, the trajectory is able to better align the time windows to achieve a lower cost. However, it should be noted that because of the time-variant traffic condition, the shape of the trajectory, specifically, the slope of the moved segments, will also change accordingly. This requires further tuning of the continuous variables. When there are multiple customers and recharging, the tuning process can be rather complicated. Therefore, we proposed an iterative workflow to solve this tuning process for each BET given in Algorithm 1.

Algorithm 1: Fine tuning process

```
Start from the route calculated by the ACO algorithm
for i from N to 0 (N is the recharging times)
  repeat
    record cost
    if i is not 0
      change charging strategy of charging station i to minimize the cost from current node to the end
      change charging strategy of charging station i to minimize the cost from current node to the end
      change charging strategy of charging station i to minimize the cost from current node to the end
      change charging strategy of charging station i to minimize the cost from current node to the end
    else
      change departure time to minimize the overall cost
      update traffic condition
    end if
  until (recorded cost - current cost) < threshold
end
```

C. Heuristic Algorithm

For validating the proposed bi-level hierarchical method, a heuristic algorithm is deployed as the baseline, which uses the greedy strategy to search for the feasible solution of the proposed optimization problem. The pseudocode of the heuristic algorithm is given in Algorithm 2.

Algorithm 2: Heuristic algorithm

```
Sort BETs in the order of SOC level from high to low
for each BET
    manage target nodes based on pickup/delivery history of all BETs
    if the elements in the target nodes less than 3
        add home base into the set of target nodes
    end if
    sort target nodes in the order of the time window constraints from early to late
    for each target node
```

```
9:          if (distance from the current node to the
            target node + distance from the target node
            to the closest charging station) >= rest range
10:             select the unvisited node as next node
11:             break
12:         else
13:             continue
14:         end if
15:     end
16:     if no feasible next node found
17:         select the closest charging station as the
            next node, use standard charging to the
            100% battery capacity
18:     end if
19: end
```

IV. Case Study

To validate the proposed bi-level hierarchical method, we perform the numerical simulation in this section. The previously introduced heuristic algorithm is carried out as a baseline. The numerical case study setup (in TABLE II) is specified as follows.

TABLE II. Case Study Setup

Type	Variable	Value
Scenario Assumptions	Map scale	70 × 70 miles
	Number of costumers	8
	Number of charging stations	5
	Loading/unloading time	0.3 hour
	Cargo weight	(0, 5] tons
	Working hour	[8 am, 6 pm]
	Buffer time	4 hours
	Traffic condition fidelity	0.5 hour
	Vehicle speed	40, 50, 60 mph
	Charging rate	1, 2, 3 hours to full capacity
	BET range	150 miles
	BET carrying capacity	20 tons
	Labor cost rate	25 dollars/hour
	Early/late penalty rate	90 dollars/hour
	Charging rate	0.3, 0.5, 0.7 dollars/mile
ACO Parameters	Number of ants	150
	Number of iterations	600
	Pheromone weight	0.8
	Heuristic term weight	0.1
	Evaporation rate	0.4
	Pheromone increasing rate	5

To mimic the regional dispatching scenario, we generate a map on the scale of 70 miles square (i.e., 70 × 70 miles). Within this region, we randomly define 5 charging stations and 8 customers with 16 corresponding pickup and delivery locations. For each location, the loading/unloading time is set to be identical to 0.3 hours, and the cargo weight is randomly set within 5 tons. Also, the time windows are stochastically created no shorter than one hour. We set the working hour of the fleet to be from 8 am to 6 pm, and a 4-hour buffer time after 6 pm is set such that the late returning is allowed with the penalty. The fidelity of the time-variant traffic condition is 0.5 hour. Within the 0.5-hour intervals, we define three

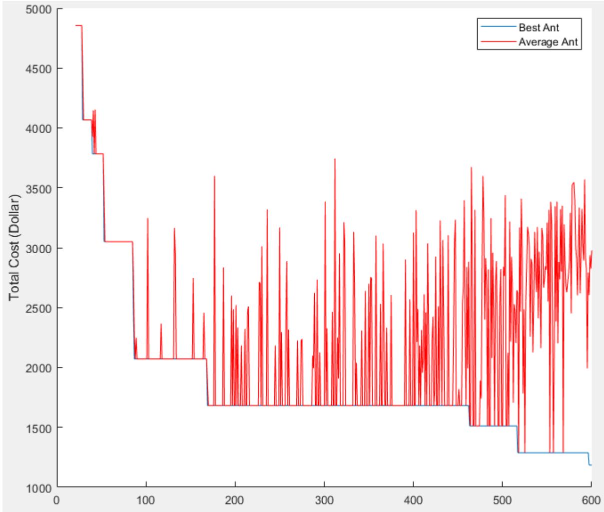

Figure 4. Convergence curve of ACO algorithm.

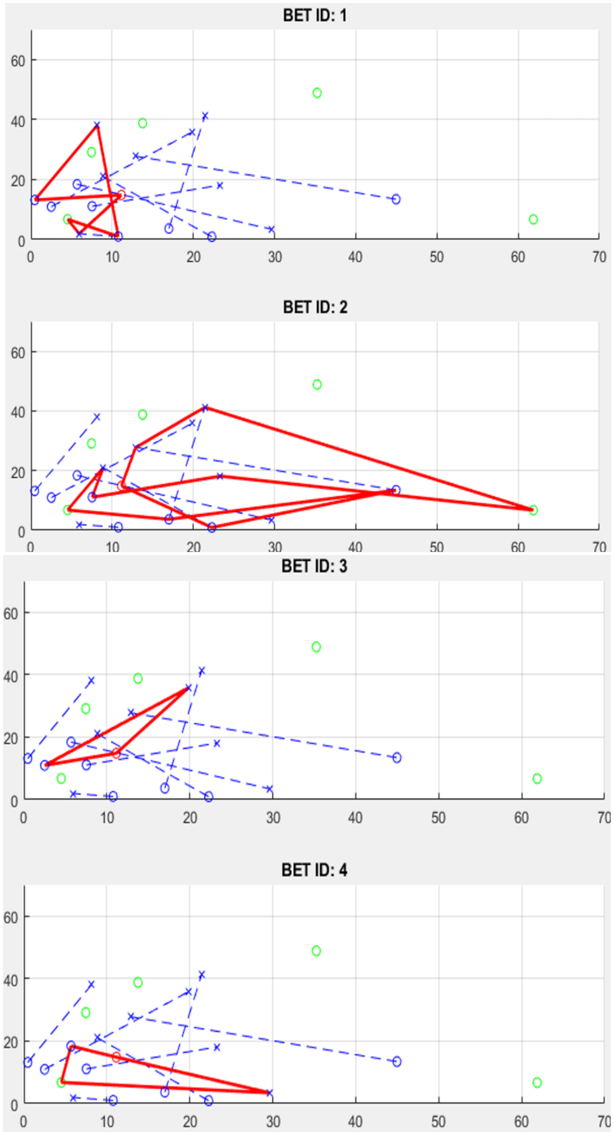

Figure 5. Shortest route giving by ACO algorithm.

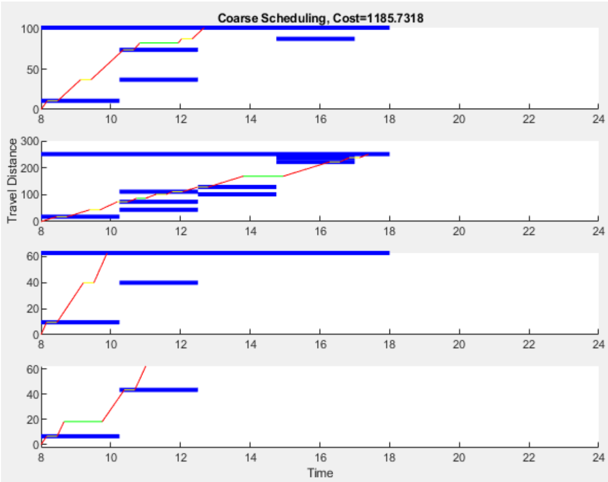

Figure 6 Travel distance – time plot giving by ACO algorithm.

levels of the vehicle speed, namely, 40 mph, 50 mph and 60 mph, to express different congestion levels. The distance from one location to another is calculated by their Euclidian distance. As to the charging stations, we use the full charging time, 1 hour, 2 hours and 3 hours to depict three types of charging, i.e. slow charging, regular charging, and fast charging. We further assume the charging process is at a linear rate. Therefore, if a BET aims to charge from 50% to 100% using regular charging, for instance, it will take 1 hour. From the BET perspective, we use the remaining range to reflect the battery status, and the maximum range of a BET is 150 miles in this case study. Still, we assume a linear relationship between the battery/mileage consumption and the driving distance. There are 4 BETs in the fleet, and the carrying capacity of each BET is identical to 20 tons. Finally, we define the labor cost rate as 25 dollars per hour, the early/late penalty rate as 90 dollars per hour, and the charging rate for three types as 0.3, 0.5 and 0.7 dollars per mile respectively.

The parameters of the ACO algorithm are specified as follows. There are 150 ants searching in parallel for 600 iterations. The importance of the pheromone, α, is set to be 0.8, while the importance of the heuristic term, β, is set to be 0.1; The evaporation rate of the pheromone is 0.4, and the pheromone increasing rate is 5. The convergence curve of the ACO algorithm is shown in Figure 4. In the figure, the blue curve shows the performance of the best ant which is kept to the following iteration. The red curve shows the average performance of the ants that found the feasible route in the current iteration. The fluctuation of the red curve indicates the randomicity of the exploration within the searching spacing, which prevents the search from falling into the local optimum.

The routes and the dispatching schemes are shown in Figure 5. The red circle depicts the home base, and the green circles denote the locations of the charging station. The blue circles and cross marks connected by the blue dash lines represent pairs of the pickup and delivery locations of corresponding customers. The travel distance-time plot illustrates the solution given by the ACO algorithm (see Figure 6). The trajectories are marked with three different colors. The red segments show that the BETs travel from one location to another; the yellow segments show the loading/unloading stages; the green segments show the charging stages. The blue bars represent the time window constraints. From the figure, we can observe that the trajectories align well with the time windows in the conditions of the fixed departure time and charging strategy. A lower-cost alternative can be expected after the Fine Scheduling process.

The results of the *Fine Scheduling* algorithm are shown in Figure 7, and the costs are given in TABLE III. In this process, the results from the ACO algorithm are applied. Therefore, similar trajectories are demonstrated in the figure. By moving the departure time along the time horizon, and adjusting the charging strategy, more time window constraints can be fulfilled. As a result, much lower cost is given. Although the heuristic algorithm is capable of finding a feasible solution, the final cost is far from being acceptable. On the other hand, the result purely from ACO algorithm can already save 30% of the cost. When applying *the Fine Scheduling* algorithm, the overall cost is reduced by 60%.

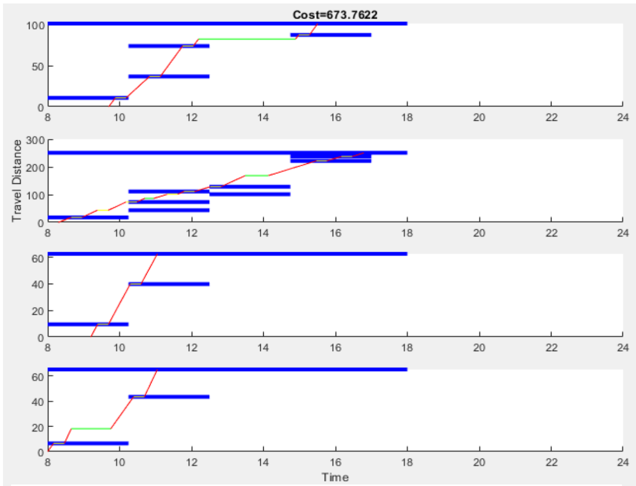

Figure 7 Travel distance – time plot giving by Fine Scheduling algorithm.

TABLE III. COST COMPARISON FOR DIFFERENT ALGORITHMS

	ACO Algorithm	ACO + Fine Scheduling Algorithm	Heuristic Algorithm
Cost	1185.7318	673.7622	1681.7737

V. CONCLUSIONS

In this paper, we proposed a methodology to route and dispatch energy-constrained BETs by a two-step optimization of the "electric vehicle routing problem" with pickup and delivery, time windows and partial recharge. We first apply the ACO algorithm coarsely scheduled the BETs. In order to speed up computation, the departure times for BETs were discretized, and the charging strategy was fixed. Then, with the optimization result from the ACO algorithm, we recover the continuity of the problem by the fine scheduling process. The case study results show a 30% saving of the operation cost comparing the ACO algorithm only with the heuristic algorithm, and a 60% saving comparing the bi-level method with the heuristic algorithm. This is because the ACO algorithm can find out the near optimal solution out of a large set of feasible solutions and allowing flexible departure time and charging strategies can further satisfy the time window requirement of the customers. Some possible directions for future research include adding detailed energy consumption and travel time estimation models to obtain more accurate itineraries and incorporating real-world scenarios to evaluate the system effectiveness.

ACKNOWLEDGMENT

This paper was prepared as a result of work sponsored by the Volvo LIGHTS (Low Impact Green Heavy Transport Solutions) project (www.lightsproject.com). Volvo LIGHTS is part of California Climate Investments, a statewide program that puts billions of Cap-and-Trade dollars to work reducing greenhouse gas emissions, strengthening the economy, and improving public health and the environment — particularly in disadvantaged communities. Funding is also provided by South Coast Air Quality Management District's Clean Fuels Program, which since 1988 has provided over $320 million, leveraging $1.2 billion, to fund projects to accelerate the demonstration and deployment of clean fuels and transportation technologies through public-private partnerships. The opinions, findings, conclusions, and recommendations are those of the authors and do not necessarily represent the views of project sponsors or other team members.

REFERENCES

[1] Helvig, Christopher S., Gabriel Robins, and Alex Zelikovsky. "The moving-target traveling salesman problem." Journal of Algorithms 49.1,(2003): 153-174.

[2] Gambella, Claudio, Joe Naoum-Sawaya, and Bissan Ghaddar. "The vehicle routing problem with floating targets: Formulation and solution approaches." INFORMS Journal on Computing 30.3 (2018): 554-569..

[3] Caric, Tonci, and Hrvoje Gold. "Vehicle routing problem". 2008.

[4] Ralphs, Ted K., et al. "On the capacitated vehicle routing problem." Mathematical programming 94.2-3 (2003): 343-359.

[5] Desrochers, Martin, Jacques Desrosiers, and Marius Solomon. "A new optimization algorithm for the vehicle routing problem with time windows." Operations research 40.2 (1992): 342-354.

[6] Yanik, Seda, Burcin Bozkaya, and Ronan deKervenoael. "A new VRPPD model and a hybrid heuristic solution approach for e-tailing." European Journal of Operational Research 236.3 (2014): 879-890.

[7] Bruglieri, Maurizio, Alberto Colorni, and Alessandro Lue. "The vehicle relocation problem for the one-way electric vehicle sharing: an application to the Milan case." Procedia-Social and Behavioral Sciences 111 (2014): 18-27.

[8] Erdelic, Tomislav, et al. "Electric vehicle routing problem with single or multiple recharges." Transp. Res. Procedia 40 (2019): 217-224.

[9] Afroditi, Anagnostopoulou, et al. "Electric vehicle routing problem with industry constraints: trends and insights for future research." Transportation Research Procedia 3 (2014): 452-459.

[10] Bard, Jonathan F., et al. "A branch and cut algorithm for the VRP with satellite facilities." IIE transactions 30.9 (1998): 821-834.

[11] Keskin, Merve, and Bülent Çatay. "Partial recharge strategies for the electric vehicle routing problem with time windows." Transportation Research Part C: Emerging Technologies 65 (2016): 111-127.

[12] Omidvar, A., and R. Tavakkoli-Moghaddam. "Sustainable vehicle routing: Strategies for congestion management and refueling scheduling." 2012 IEEE international energy conference and exhibition (ENERGYCON). IEEE, 2012.

[13] Rizzoli, Andrea Emilio, et al. "Ant colony optimization for real-world vehicle routing problems." Swarm Intelligence 1.2 (2007): 135-151.

[14] Hiermann, Gerhard, et al. "Routing a mix of conventional, plug-in hybrid, and electric vehicles." European Journal of Operational Research 272.1 (2019): 235-248.

[15] Dorigo, Marco. "Optimization, learning and natural algorithms." PhD Thesis, Politecnico di Milano (1992).