

# Minimizing Carbon Footprint for Timely E-Truck Transportation: Hardness and Approximation Algorithm

Junyan Su, Qiulin Lin, Minghua Chen, Haibo Zeng

**Abstract**— Carbon footprint optimization (CFO) is important for sustainable heavy-duty e-truck transportation. We consider the CFO problem for timely transportation of e-trucks, where the truck travels from an origin to a destination across a national highway network subject to a deadline. The goal is to minimize the carbon footprint by orchestrating path planning, speed planning, and intermediary charging planning. We first show that it is NP-hard even just to find a feasible CFO solution. We then develop a  $(1 + \epsilon_F, 1 + \epsilon_\beta)$  bi-criteria approximation algorithm that achieves a carbon footprint within a ratio of  $(1 + \epsilon_F)$  to the minimum with no deadline violation and at most a ratio of  $(1 + \epsilon_\beta)$  battery capacity violation (for any positive  $\epsilon_F$  and  $\epsilon_\beta$ ). Its time complexity is polynomial in the size of the highway network,  $1/\epsilon_F$ , and  $1/\epsilon_\beta$ . Such algorithmic results are among the best possible unless P=NP. Simulation results based on real-world traces show that our scheme reduces up to 11% carbon footprint as compared to baseline alternatives considering only energy consumption but not carbon footprint.

## I. INTRODUCTION

Electrifying transportation is a promising and continuing trend toward achieving carbon neutrality for fighting climate change. Within the transportation sector, the trucking industry has garnered increasing attention due to its significant environmental impact. For instance, heavy-duty trucks in the US produce 25% of the CO<sub>2</sub> emissions in the transportation sector [1], despite only accounting for 4% of the total vehicle population. Furthermore, heavy-duty trucks are responsible for 8.8% of the total carbon emissions in the US.

Adopting electric trucks (e-trucks) presents significant potential for greening the heavy-duty truck industry as they produce zero CO<sub>2</sub> emissions during operation. However, charging e-trucks with energy generated from carbon-intensive resources (e.g., coal or petroleum) instead of carbon-free ones (e.g., renewable energy) incurs a non-trivial *carbon footprint*. The average *carbon intensity* (unit: kg CO<sub>2</sub> emission/kWh) of coal and petroleum generated electricity is 1.02 kg/kWh and 0.91 kg/kWh, respectively [2]. In contrast, renewable energy has a carbon intensity of zero. Therefore, to fully unlock the environmental benefits of e-trucks, it is crucial to charge them using electricity with low carbon intensity. Meanwhile, the carbon intensity of the electricity exhibits significant temporal and spatial variations due to fluctuating renewable penetration [3]. It is thus critical to decide when and where to charge during an e-truck transportation task. This observation, combined with the time-sensitive nature

of truck operations [4], emphasizes the need for joint optimization of path planning, speed planning, and charging planning for e-trucks to minimize their carbon footprints for sustainability.

In this paper, we study the carbon footprint optimization (CFO) of a heavy-duty e-truck, traveling from an origin to a destination across a national highway network subject to a hard deadline, by exploring the complete design space of path planning, speed planning, and en-route charging planning.

Most related existing studies focus on minimizing the energy consumption of conventional internal combustion engine (ICE) trucks [5], [6], [7], electric trucks [8], [9] and electric vehicles [10], [11], [12], [13], [14]. However, energy-efficient operations may not necessarily lead to a low carbon footprint. Our simulation based on real-world traces also shows that carbon-aware solutions achieve up to 11% less carbon footprint than the energy-efficient ones. Moreover, the temporal-spatial variation of the carbon intensity results in a non-convex and non-monotone objective that introduces additional challenges. Indeed, the existing discretization technique in [14], which relies on the monotonicity of the discretized function, is not directly applicable to the CFO problem. Our approximation algorithm differentiates from the existing ones, e.g., [14], in that we consider a realistic non-convex objective with both necessary battery and deadline constraints. A recent work [15] also considers carbon-aware truck operations. However, our work differentiates from it in that we present new theoretical hardness results (cf. Theorem. 1) and an approximation algorithm for the CFO problem. We summarize our **contributions** in the following.

▷ In Sec. II, we present the modeling and problem settings of CFO. We then show that it is NP-hard even just to find a feasible solution to the CFO problem, which directly implies that the NP-hardness of the problem.

▷ In Sec. III, we develop a bi-criteria  $(1+\epsilon_F, 1+\epsilon_\beta)$ -approximation algorithm that achieves at most  $(1 + \epsilon_F)$  carbon footprint of the optimal solution with at most  $(1 + \epsilon_\beta)$  battery capacity, for arbitrary  $\epsilon_F$  and  $\epsilon_\beta$ . The computational complexity of our algorithm is polynomial in the network size,  $1/\epsilon_F$ , and  $1/\epsilon_\beta$ . Such algorithmic results are among the best possible unless P=NP.

▷ In Sec. IV, we conduct simulations on real-world traces and the results show that our carbon-footprint optimized scheme saves up to 11% carbon footprint as compared to baseline alternatives considering only energy consumption but not carbon footprint.

Due to the space limitation, all proofs are included in the

Junyan Su, Qiulin Lin and Minghua Chen are with City University of Hong Kong, Hong Kong, China ({junyan.su@my., qiulin.lin@, minghua.chen@}cityu.edu.hk). Haibo Zeng is with Virginia Tech (hzeng@vt.edu).

Corresponding author: Minghua Chen.

technical report [16].

## II. MODELS AND PROBLEM SETTINGS

In this section, we discuss the system model and problem formulation of our proposed carbon footprint optimization (CFO) problem. We further show that the problem is NP-hard, even for finding a feasible solution.

**Transportation Graph and Energy Consumption.** We consider an e-truck traveling across a national highway system modeled by a directed graph  $G = (V, E)$ . Here  $V = V_r \cup V_c$  is the set of nodes, where  $V_r$  denotes the set of nodes connecting highway road segments and  $V_c$  denotes the set of nodes for the charging stations. We denote the set of edges by  $E$  where each edge represents a road segment. For each edge  $e \in E$ , we define its minimum and maximum traveling time by  $t_e^{lb}$  and  $t_e^{ub}$  respectively. Without loss of generality, we assume homogeneous road condition (e.g., road grade) on each road segment and define the function  $c_e(t_e) : [t_e^{lb}, t_e^{ub}] \rightarrow \mathbb{R}$  as the *energy consumption function* for an e-truck traveling across edge  $e$  with traveling time  $t_e$ . In practice,  $c_e(t_e)$  is non-increasing over  $[t_e^{lb}, t_e^{ub}]$  as there is no benefit to drive with both high energy consumption and large traveling time [5]. We denote the minimum and maximum energy consumption for the road segment  $e \in E$  as  $c_e^{lb}, c_e^{ub}$ , respectively. Note that for e-trucks,  $c_e(t_e)$  can take negative values over certain road segments (e.g., downhill paths) because of regenerative braking [17].

**E-Truck Charging.** We model that an e-truck needs to make two decisions at a charging station: wait for  $t_w \in [t_w^{lb}, t_w^{ub}]$  unit of time and charge for  $t_c \in [0, t_c^{ub}]$  unit of time. Here we allow an e-truck to wait at the charging station for cleaner electricity. The lower bound  $t_w^{lb}$  denotes the time overhead at the charging station (for e.g., parking and paying the bills). We model the charging process at charging station  $v$  as a concave function  $\Phi_v(t)$ , which represents the final State-of-Charge (SoC) from zero after charging for  $t$  unit of time. We denote by  $\beta_v$  as the SoC when the e-truck arrives at node  $v \in V_c$ . Then, with initial SoC  $\beta_v$ , the increment of SoC after a charging time  $t_c$  at the charging station  $v \in V_c$  can be determined as

$$\phi_v(t_c, \beta_v) = \Phi_v(\Phi_v^{-1}(\beta_v) + t_c) - \beta_v, \quad (1)$$

where  $\Phi_v^{-1}$  is the inverse function of  $\Phi_v$ .

**SoC Evolution.** According to the energy consumption model, the SoC decreases by  $c_e(t_e)$  traveling over edge  $e = (u, v) \in E$  with  $v \notin V_c$  with time  $t_e$ , i.e.,

$$\beta_v = \min \{B, \beta_u - c_e(t_e)\}, \quad (2)$$

where  $B$  is the battery capacity. According to the energy consumption and the charging model, suppose an e-truck travels across edge  $e = (u, v) \in E$  with traveling time  $t_e$  and charges at  $v \in V_c$  with charging time  $t_c$ , the SoC update rule is given by

$$\beta_v = \min \{B, \beta_u - c_e(t_e) + \phi_v(t_c, \beta_u)\}. \quad (3)$$

**Carbon Intensity and Carbon Footprint.** For each charging station  $v \in V_c$ , we model the carbon intensity as a

piecewise monotone function  $\pi_v(\tau)$  (unit: kg/kWh) with  $\tau$  being the moment of charging process, i.e.,

$$\pi_v(\tau) = \begin{cases} \pi_{v,1}(\tau), & \text{if } \tau \in [0, \tilde{\tau}_1]; \\ \dots, & \dots; \\ \pi_{v,n_\pi}(\tau), & \text{if } \tau \in [\tilde{\tau}_{n_\pi-1}, \tilde{\tau}_{n_\pi}]; \end{cases} \quad (4)$$

In (4),  $\pi_v(\tau)$  consists of  $n_\pi$  pieces with breaking points  $\tilde{\tau}_1, \tilde{\tau}_2, \dots, \tilde{\tau}_{n_\pi}$ , and each piece  $\pi_{v,i}, \forall i = 1, \dots, n_\pi$ , is a monotone function with a well-defined inverse function  $\pi_{v,i}^{-1}(\cdot)$ . Note that in practice the carbon intensity is measured and forecast at the hourly scale [3] and often interpolated as a piece-wise linear function, which fits into our model.

Given the carbon intensity function  $\pi_v(\cdot)$  at a charging station  $v \in V_c$ , the initial SoC  $\beta_v$ , the arrival moment  $\tau_v$ , and the charging time  $t_c$ , the induced carbon footprint is

$$F_v(\beta_v, t_c, \tau_v) = \frac{1}{\eta} \pi_v(\tau_v) \phi(t_c, \beta_v), \quad (5)$$

where we denote by  $0 < \eta \leq 1$  the charging efficiency of the battery. Note that here we adopt the first-order approximation of the carbon footprint, i.e., we assume the carbon intensity is fixed during the charging process. This is reasonable as the typical charging time is less than 30 minutes while the carbon intensity is usually measured hourly [3].

**Carbon Footprint Optimization Problem.** We consider the *Carbon Footprint Optimization* (CFO) problem of an e-truck traveling across a national highway network from an origin  $s \in V$  to a destination  $d \in V$  subject to the deadline  $T$ . For ease of presentation, we consider that the e-truck starts at moment zero, i.e.,  $\tau_s = 0$ . The e-truck equips with an initially fully charged battery with capacity  $B$ . We denote a path by  $\vec{p} = [s = p_1, p_2, \dots, p_{n_p} = d] \in \mathcal{P}$ , where  $n_p$  is the number of nodes in path  $\vec{p}$  and  $\mathcal{P}$  is the set of all paths from  $s$  to  $d$ . We denote the traveling time from  $p_i$  to  $p_{i+1}$  as  $t_i^r$  for all  $i \in \{1, 2, \dots, n_p - 1\}$ . Further, we denote by  $\mathcal{I}_p \subset \{1, \dots, n_p\}$  the set of index for charging stops, i.e., for any  $j \in \mathcal{I}_p$ ,  $p_j \in V_c$  is a charging stop. We define the *Carbon Footprint Optimization* (CFO) problem as follows:

$$\min \sum_{j \in \mathcal{I}_p} F_{p_j}(\beta_j, t_j^c, \tau_j) \quad (6a)$$

$$\text{s.t.} \quad \sum_{i=1}^{n_p} t_i^r + \sum_{j \in \mathcal{I}_p} (t_j^w + t_j^c) \leq T, \quad (6b)$$

$$\tau_j = \sum_{i=1}^{j-1} t_i^r + \sum_{k \in \mathcal{I}_p; k < j} (t_k^w + t_k^c) + t_j^w, \quad (6c)$$

$$\beta_j = \min \left\{ B, \beta_{j-1} - c_{(p_{j-1}, p_j)}(t_{j-1}^r) + \phi_{p_j}(t_j^c, \beta_{j-1}) \right\}, \forall j \in \mathcal{I}_p, \quad (6d)$$

$$\beta_i = \min \{B, \beta_{i-1} - c_{(p_{i-1}, p_i)}(t_{i-1}^r)\}, \quad \forall i \in \{1, 2, \dots, n_p\} \setminus \mathcal{I}_p \quad (6e)$$

$$\beta_i \geq 0, \forall i \in \{1, 2, \dots, n_p\}, \quad (6f)$$

$$\beta_0 = B, \quad (6g)$$

$$\text{var. } \vec{p} \in \mathcal{P}, \quad (6h)$$

$$t_i^r \in [t_{(p_{i-1}, p_i)}^{lb}, t_{(p_{i-1}, p_i)}^{ub}], \beta_i, \forall i \in \{1, 2, \dots, n_p\}, \quad (6i)$$

$$t_j^w \in [t_w^{lb}, t_w^{ub}], t_j^c \geq 0, \tau_j, \forall j \in \mathcal{I}_p. \quad (6j)$$

Our objective in (6a) is the carbon footprint incurred by charging at  $m_p = |\mathcal{I}_p|$  charging stops. The deadline constraint is in (6b). In (6c), we obtain the arrival time at  $j$ -th stop, which is required for determining the carbon footprint in (6a). The evolution of SoC is determined by (6d) and (6e), according to our discussions in (2) and (3). We require the SoC to be always non-negative in (6f) and set the initial SoC as the battery capacity in (6g).

**Theorem 1.** *The CFO problem is NP-hard. Further, it is NP-hard even just to find a feasible solution for the problem.*

Theorem 1 implies that unless P=NP, one could not expect to solve the problem or find a feasible solution within a time complexity polynomial to the network size and the input size of  $B$  and  $T$ . In such case, the best possible algorithmic results one could expect is then a polynomial time bi-criteria approximation scheme [18]. We will design one such algorithm and characterize its performance in the next section.

### III. A BI-CRITERIA APPROXIMATION ALGORITHM

In this section, we develop a  $(1 + \epsilon_F, 1 + \epsilon_\beta)$  bi-criteria approximation algorithm that achieves a carbon footprint within a ratio of  $(1 + \epsilon_F)$  to the minimum with no deadline violation and with at most a ratio of  $(1 + \epsilon_\beta)$  battery capacity violation (for any positive  $\epsilon_F$  and  $\epsilon_\beta$ ) and a time complexity polynomial in the size of the highway network,  $1/\epsilon_F$ , and  $1/\epsilon_\beta$ . The essence of our approximation algorithm is a test procedure as outlined in Algorithm 1. Given a guess  $\Omega > 0$  of the optimal objective OPT to the CFO problem, Algorithm 1 approximately compare them and report if  $\text{OPT} > \Omega$  or  $\text{OPT} \leq (1 + \epsilon_F)\Omega$ . Then we use a bisection technique (cf. Algorithm 2) over the possible range of OPT with this test procedure to obtain an  $\Omega$  close enough to OPT and recover its corresponding solution to the CFO problem.

In Algorithm 1, we seek to find a feasible path that is within the given objective upper bound  $\Omega$  or report infeasibility otherwise. We discretize the SoC and the objective value and then use dynamic programming (DP) to obtain the optimal path under discretized values. We carefully determine the discretization precision to achieve the  $(1 + \epsilon_F, 1 + \epsilon_\beta)$  approximation while ensuring polynomial time complexity. Meanwhile, our discretization procedure guarantees that DP is able to find a feasible solution if  $\text{OPT} \leq (1 + \epsilon)\Omega$ .

We divide the algorithm into two phases. In the first phase, we compute the energy-constrained fastest path from a charging station to other nodes without intermediary charging stops. In the second phase, we decide the charging locations and scheduling by utilizing the information from phase I. Then we recover the solution based on the results of those two phases or report infeasible otherwise.

**Phase I: Energy-Constrained Fastest Path from Single Source.** We use the rounding and scaling technique [5],

---

#### Algorithm 1 APXTEST( $s, d, T, B, \Omega, \epsilon_\beta, \epsilon_F$ )

---

- 1:  $\delta_\beta \leftarrow \frac{\epsilon_\beta B}{|V| + |V_c| + 1}, M_\beta \leftarrow \lceil \frac{|V| + |V_c| + 1}{\epsilon_\beta} \rceil + |V| + |V_c| + 1$
  - 2:  $\delta_F \leftarrow \frac{\epsilon_F \Omega}{|V| + |V_c| + 1}, M_F \leftarrow \lceil \frac{|V| + |V_c| + 1}{\epsilon_F} \rceil + |V| + |V_c| + 1$
  - 3:  $\hat{\beta}_0 \leftarrow \lfloor (1 + \epsilon_\beta)B / \delta_\beta \rfloor \cdot \delta_\beta$
  - 4: **Phase I: energy-constrained path without charging stops.**
  - 5: **for**  $\forall i_u \leftarrow 0, 1, \dots, M_\beta, \forall u \in V_c \cup \{s\}$  **do**
  - 6:      $\beta_u \leftarrow i_u \delta_\beta$
  - 7:      $\psi_{u, \beta_u}(u, \beta_u) \leftarrow 0$  and  $\psi_{u, \beta_u}(\cdot, \cdot) \leftarrow \infty$  otherwise.
  - 8:     **for**  $\forall i_v \leftarrow 0, \dots, M_\beta, \forall v \in V$  **do**
  - 9:         **for**  $\forall (w, v) \in E, \forall i_c \leftarrow \lfloor \frac{c_w^{lb}}{\delta_\beta} \rfloor \dots \lfloor \frac{c_w^{ub}}{\delta_\beta} \rfloor$  **do**
  - 10:              $(\beta_v, \hat{c}) \leftarrow (i_v \delta_\beta, i_c \delta_\beta)$
  - 11:              $\hat{\beta}_w \leftarrow \beta_w(\beta_v, \hat{c})$  according to (9).
  - 12:             **if**  $\beta_w \in [0, (1 + \epsilon_\beta)B]$  **then**
  - 13:                 Update  $\psi_{u, \beta_u}(v, \beta_v)$  according to (11).
  - 14: **Phase II: charging planning.**
  - 15:  $\sigma_{s, \hat{\beta}_0}(s, 0, \hat{\beta}_0) \leftarrow 0$  and  $\sigma_{s, \hat{\beta}_0}(\cdot, \cdot, \cdot) \leftarrow \infty$  otherwise.
  - 16: **for**  $\forall i_\rho \leftarrow 0, \dots, M_F, \forall i_F \leftarrow 0, \dots, i_\rho$  **do**
  - 17:     **for**  $\forall i_c \leftarrow 0, \dots, M_\beta, \forall i_v \leftarrow 0, \dots, M_\beta, \forall i_u \leftarrow 0, \dots, M_\beta,$  **do**
  - 18:         **for**  $\forall u \in V_c \cup \{s\}, \forall v \in V_c \cup \{d\}$  **do**
  - 19:              $(\rho, \hat{F}) \leftarrow (i_\rho \delta_F, i_F \delta_F)$
  - 20:              $(\beta_u, \beta_v, \beta_c) \leftarrow (i_u \delta_\beta, i_v \delta_\beta, i_c \delta_\beta)$
  - 21:              $\beta'_u = \min\{B, \beta_u + \beta_c\}$
  - 22:              $\sigma_u \leftarrow \sigma_{s, \hat{\beta}_0}(u, \rho - \hat{F}, \beta_u)$
  - 23:              $t_w \leftarrow g_v(\beta_u, t_c, \sigma_u, \hat{F})$  (cf. problem (14))
  - 24:              $t_c \leftarrow \Phi_u^{-1}(\beta'_u) - \Phi_u^{-1}(\beta_u)$
  - 25:              $t_r \leftarrow \psi_{u, \beta'_u}(v, \beta_v)$
  - 26:              $\tilde{\sigma} \leftarrow \sigma_u + t_r + t_w + t_c$
  - 27:              $\sigma_v \leftarrow \sigma_{s, \hat{\beta}_0}(v, \rho, \beta_v)$
  - 28:             **if**  $\tilde{\sigma} \leq \sigma_v$  **then**
  - 29:                  $\sigma_{s, \hat{\beta}_0}(v, \rho, \beta_v) \leftarrow \tilde{\sigma}$
  - 30: Let  $i_\rho^* \in \{0, \dots, M_F\}$  be the minimal  $i_\rho$  such that  $\sigma_{s, \hat{\beta}_0}(d, i_\rho \delta_F, 0) \leq T$
  - 31: **if**  $\rho^*$  exists **then**
  - 32:     sol  $\leftarrow$  the solution corresponding to  $\rho^*$
  - 33: **else**
  - 34:     sol  $\leftarrow$  NULL
  - 35: **return** sol
- 

[14] to ensure the performance and guarantee polynomial time complexity. In particular, we discretize the SoC into  $M_\beta$  states with precision  $\delta_\beta$  (cf. line 1 of Algorithm 1). For each discretized energy consumption  $\hat{c}$ , we compute its corresponding travel time by

$$\hat{t}_e(\hat{c}, \delta_\beta) = \begin{cases} t_e^{lb}, & \text{if } \lceil \frac{\hat{c}}{\delta_\beta} \rceil = \lceil \frac{c_e^{ub}}{\delta_\beta} \rceil; \\ c_e^{-1}(\hat{c}), & \text{if } \hat{c} \in [c_e^{lb}, c_e^{ub}); \\ \infty, & \text{otherwise.} \end{cases} \quad (7)$$

Recall that  $c_e^{lb}, c_e^{ub}$  are the minimum and maximum energy consumption for the road segment  $e \in E$ , respectively.

We then use dynamic programming (DP) to compute the

energy-constrained fastest solutions without charging stops. We denote by  $\psi_{u,\beta_u}(v, \beta_v)$  the minimal traveling time from  $u$  to  $v$  with initial SoC  $\beta_u$  and the final SoC lower bounded by  $\beta_v$ . Then the Bellman equation is given by

$$\psi_{u,\beta_u}(v, \beta_v) = \min \left\{ \psi_{u,\beta_u}(v, \beta_v - \delta_\beta), \min_{(w,\hat{c}) \in \mathcal{N}(v,\beta_v)} \left\{ \psi_{u,\beta_u}(w, \beta_w(\beta_v, \hat{c})) + \hat{t}_{(w,v)}(\hat{c}, \delta_\beta) \right\} \right\}, \quad (8)$$

with the initial condition  $\psi_{u,\beta_u}(u, \beta_u) = 0$ . Here the previous SoC state  $\beta_w$  is given by

$$\beta_w(\beta_v, \hat{c}) = \begin{cases} (1 + \epsilon_\beta)B - \hat{c}, & \text{if } \beta_v = (1 + \epsilon_\beta)B \text{ \& } \hat{c} < 0; \\ \beta_v + \hat{c}, & \text{if } (\beta_v + \hat{c}) \in [0, (1 + \epsilon_\beta)B]; \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Note that here we drop the harvested energy when battery overflow happens. The state set  $\mathcal{N}(v, \beta_v)$  is given by

$$\mathcal{N}(v, \beta_v) = \left\{ (w, \hat{c}) : e = (w, v) \in E, \hat{c} \in \left\{ i_c \delta_\beta : i_c \in \lfloor c_e^{lb} / \delta_\beta \rfloor, \dots, \lfloor c_e^{ub} / \delta_\beta \rfloor \right\}, \beta_w(\beta_v, \hat{c}) \in [0, (1 + \epsilon_\beta)B] \right\}. \quad (10)$$

Therefore, given any  $u, v \in V$ , any discretized states  $\beta_u, \beta_v$ , any road segment  $e = (w, v) \in E$ , and discretized energy consumption  $\hat{c}$ , we update  $\psi$  in line 13 of Algorithm 1 as follows:

$$\psi_{u,\beta_u}(v, \beta_v) = \min \left\{ \psi_{u,\beta_u}(v, \beta_v), \psi_{u,\beta_u}(w, \beta_w) + \hat{t}_e(\hat{c}, \delta_\beta) \right\}. \quad (11)$$

The overall phase I procedure is summarized in line 4-13 of Algorithm 1. For any source node  $u$  with the initial SoC  $\beta_u$ , we update  $\psi$  in a bottom-up manner: we first update for all nodes with final SoC  $\beta_v = \delta_\beta$ , and then  $\beta_v = 2\delta_\beta$ , until  $\beta_v = M_\beta \delta_\beta$  and get the complete value function  $\psi$ .

**Phase II: Routing and Planning for Charging.** In phase II, we discretize the objective value with precision  $\delta_F$  and the total number of discretized values  $M_F$ . We then use DP to compute the approximate solution. We denote by  $\sigma_{s,\hat{\beta}_0}(v, \rho, \beta_v)$  the minimal travel time from the source  $s$  to  $v \in V_c \cup \{d\}$  with initial SoC  $\hat{\beta}_0$ , the final SoC lower bound  $\beta_v$  and the objective upper bound  $\rho$ . Then the Bellman equation is given by

$$\sigma_{s,\hat{\beta}_0}(v, \rho, \beta_v) = \min \left\{ \sigma_{s,\hat{\beta}_0}(v, \rho - \delta_F, \beta_v), \min_{(u,t_w,t_c,\beta_u,\hat{F}) \in \mathcal{M}} \left[ \sigma_{s,\hat{\beta}_0}(u, \rho - \hat{F}, \beta_u) + t_w + t_c + \psi_{u,(\phi_u(t_c,\beta_u)+\beta_u)}(v, \beta_v) \right] \right\}, \quad (12)$$

with the initial condition is  $\sigma_{s,\hat{\beta}_0}(s, 0, \hat{\beta}_0) = 0$ . The state set  $\mathcal{M}$  is given by

$$\mathcal{M} = \left\{ (u, t_w, t_c, \beta_u, \hat{F}) : \forall u \in V_c \cup \{s\}, F_u(\beta_u, t_c, \tau_u + t_w) \leq \hat{F}, \tau_u = \sigma_{s,\hat{\beta}_0}(u, \rho - \hat{F}, \beta_u), t_w \in [t_w^{lb}, t_w^{ub}], \forall \hat{F} \in \{0, \delta_F, \dots, \delta_F M_F\}, \forall \beta_u \in \{0, \delta_\beta, \dots, \delta_\beta M_\beta\}, \forall \phi_u(t_c, \beta_u) \in \{0, \delta_\beta, \dots, \delta_\beta M_\beta\} \right\}. \quad (13)$$

Therefore, in phase II, given the discretized objective value  $\hat{F}$ , the discretized SoC  $\beta_u$  at  $u$ , the SoC  $\beta_v$  at  $v$ , and the charged energy  $\beta_c$ , we can determine the charging time and the travel time between  $(u, v)$  (cf. line 24 and line 25 in Algorithm 1). However, to solve the Bellman equation (12), a simple enumeration over the discretized objective values and battery states is not sufficient because the decision variable waiting time  $t_w$  remains undetermined. To obtain the waiting time that satisfies the Bellman equation (12), we need to find the minimal waiting time subject to the objective upper bound:

$$g_u(\beta_u, t_c, \tau_u, \hat{F}) = \min_{t_w \in [t_w^{lb}, t_w^{ub}]} t_w \quad (14a)$$

$$\text{s.t. } F_u(\beta_u, t_c, \tau_u + t_w) \leq \hat{F}. \quad (14b)$$

We set  $g_v = \infty$  if the problem (14) is infeasible. The problem (14) is non-convex due to the constraint (14b). However, we are able to solve it by exploring the piecewise monotone property of the carbon intensity function.

**Lemma 1.** *The problem (14) can be solved in polynomial time by finding the minimum waiting times satisfying (14b) among all pieces in  $\pi_u(\tau)$ , i.e.*

$$t_w^* = \min_{i=1, \dots, n_\pi} \pi_{u,i}^{-1}(\hat{F} / \phi_u(t_c, \beta_u)) \quad (15)$$

The overall phase II procedure is summarized in line 14-29 of Algorithm 1. After initializing the boundary case for  $\sigma$ , we use DP to update the information of  $\sigma$ . In line 16, we define the objective upper bound  $\rho$  from source  $s$  to the node  $v$  and the objective upper bound  $\hat{F}$  from the node  $u$  to  $v$ . In line 17, we denote by  $\beta_u$  the arrival SoC at the node  $u$ , by  $\beta_v$  the arrival SoC at the node  $v$ , and by  $\beta_c$  the amount of charged energy at node  $u$ . We then compute the waiting time  $t_w$  in line 23, the charging time  $t_c$  in line 24, and the travel time  $t_r$  between the node  $u$  and the node  $v$  in line 25. We then update the  $\sigma$  in line 29 according to the Bellman equation.

Overall, the test procedure Algorithm 1 approximately compare  $\Omega > 0$  with the optimal carbon footprint OPT, and report if  $\text{OPT} > \Omega$  or  $\text{OPT} \leq (1 + \epsilon_F)\Omega$  as summarized in the following lemma.

**Lemma 2.** *If the CFO problem is feasible and  $\Omega \geq \text{OPT}$ , then Algorithm 1 must return a solution profile  $(\vec{p}, \vec{t}^r, \vec{t}^w, \vec{t}^c)$*

---

**Algorithm 2** APXCFO( $s, d, T, \epsilon_\beta, \epsilon_F$ )

---

```
1: LB  $\leftarrow$  0
2: UB  $\leftarrow$   $|V_c|B\pi^{ub}$ 
3: while  $\frac{UB}{LB} > 2$  do
4:    $\Omega \leftarrow (LB + UB)/2$ 
5:   if APXTEST( $s, d, T, \Omega, \epsilon_\beta, \epsilon_F/2$ ) = NULL then
6:      $LB \leftarrow \Omega$ 
7:   else
8:      $UB \leftarrow \Omega$ 
9:   sol  $\leftarrow$  APXTEST( $s, d, T, UB, \epsilon_\beta, \epsilon_F/2$ )
10: return sol
```

---

with the SoC profile  $\vec{\beta}$  and the initial SoC  $\hat{\beta}_0 = (1 + \epsilon_\beta)B$  that satisfy the following conditions:

$$a. \quad \sum_{i=1}^{n_p} t_i^r + \sum_{j \in \mathcal{I}_p} (t_j^w + t_j^c) \leq T, \quad (16a)$$

$$b. \quad 0 \leq \beta_i \leq (1 + \epsilon_\beta)B, \quad (16b)$$

$$c. \quad F(\vec{p}, \vec{t}^r, \vec{t}^w, \vec{t}^c) \leq OPT + \epsilon_F \Omega. \quad (16c)$$

The proof idea is to round the optimal solution to the discretized solution such that the rounded solution satisfies the conditions (16) and we can ensure that the rounded solution has been examined in the loops of Algorithm 1.

Lemma 2 implies that if Algorithm 1 returns a feasible solution in line 32, then we have  $OPT \leq (1 + \epsilon_F)\Omega$  and if Algorithm 1 returns an empty solution in line 34, we have  $OPT \geq \Omega$ . This result naturally leads to our main algorithm which bisects over the possible range of OPT.

**Main Algorithm and Performance Analysis.** We now present the main algorithm outlined in Algorithm 2. We first set the lower and upper bound of the optimal objective value in lines 1-2. At each iteration we maintain a lower bound and upper bound for the optimal objective value OPT and update one of the bounds after probing the middle point  $\Omega$ . We terminate the iteration and recover the approximate solution when the ratio between the upper bound and the lower bound is less than two. We summarize the main results for Algorithm 2 in the following two theorems.

**Theorem 2.** *If the CFO problem is feasible, then Algorithm 2 must return a solution profile  $(\vec{p}, \vec{t}^r, \vec{t}^w, \vec{t}^c)$  with the initial SoC  $\hat{\beta}_0 = (1 + \epsilon_\beta)B$  and the SoC profile  $\vec{\beta}$  that satisfy (16a), (16b) and*

$$F(\vec{p}, \vec{t}^r, \vec{t}^w, \vec{t}^c) \leq (1 + \epsilon_F)OPT. \quad (17)$$

**Theorem 3.** *The time complexity of Algorithm 2 is  $O\left((1 + 1/\epsilon)^5 n^5 |V_c|^2 n_\pi \log UB\right)$ , where  $n = |V| + |V_c|$  and  $\epsilon = \min\{\epsilon_\beta, \epsilon_F\}$ .*

By combining Theorem 2 and Theorem 3, we show that Algorithm 2 is a  $(1 + \epsilon_F, 1 + \epsilon_\beta)$  bi-criteria approximation scheme that achieves a carbon footprint within a ratio of  $(1 + \epsilon_F)$  to the optimal carbon footprint with no deadline violation and at most a ratio of  $(1 + \epsilon_\beta)$  battery violation for any positive  $\epsilon_F$  and  $\epsilon_\beta$ . The time complexity of Algorithm 2

is polynomial in the network size,  $1/\epsilon_F, 1/\epsilon_\beta$ . The algorithm is among the best possible given the hardness of the CFO problem (cf. Theorem 1).

**Discussion on constraint violation.** Our proposed approximation algorithm may result in battery capacity violations (cf. (16b)). To obtain a feasible solution without violating the battery capacity constraints of size  $B$ , we can use APXCFO with a capacity of  $B/(1 + \epsilon_\beta)$ . We refer to the CFO instance with a capacity of  $B$  as CFO( $B$ ). If CFO( $B/(1 + \epsilon_\beta)$ ) with a reduced battery capacity of  $B/(1 + \epsilon_\beta)$  is also feasible<sup>1</sup>, then APXCFO can generate a strictly feasible solution with a capacity of  $B$  and a deadline of  $T$ . In practice, it should be easy for CFO( $B/(1 + \epsilon_\beta)$ ) to be feasible by choosing a small enough  $\epsilon_\beta > 0$ .

#### IV. NUMERICAL EXPERIMENT

##### A. Experimental Setup

*Transportation network.* We collect the highway network data from the Map-based Educational Tools for Algorithm Learning (METAL) project [19], the road speed data from HERE Maps, and the elevation data from the Shuttle Radar Topography Mission (SRTM) project. We consider a typical origin-destination pair from Atlanta to Nashville, representing 7.5 billion dollars of transported freight [20]. We preprocess the network to the relevant region and merge the edges with the same grade. The reduced network has 48 nodes and 242 edges.

*Charging station data.* We collect the charging station locations from the OpenStreetMap (OSM) and obtain 11 charging stations in the considered region. We consider a piecewise linear charging function [17]. We then collect and compute the carbon intensity data from Energy Information Administration (EIA) [2].

*Energy consumption model.* We consider an e-truck with battery capacity  $B = 300$  kWh with total weight of 36 tons. We use the simulator FASTSim [21] to collect the energy consumption data under different speeds and grades and then fit the data with cubic polynomial functions.

*Algorithm Comparison.* We implement and compare the following approaches.

▷ **APX-C:** our approximation algorithm with carbon footprint objective. We set  $\epsilon_\beta = \epsilon_F = 0.1$ .

▷ **APX-E:** the energy-efficient baseline. We use our approximation algorithm with the energy minimization objective by setting the carbon intensity  $\pi(\cdot) \equiv 1$ . We also set  $\epsilon_\beta = \epsilon_F = 0.1$ .

▷ **LB:** the lower bound of the optimal objective provided by Theorem 2.

##### B. Benefit of Carbon Footprint Optimization

We vary the deadline from 7 hours to 12 hours and present the result in Fig. 1. We observe that when the deadline is tight (e.g., 7 hours), APX-E and APX-C incur similar carbon footprint. Meanwhile, as the deadline gets relaxed, the

<sup>1</sup>Note that the CFO problem with a smaller battery capacity may not be feasible, since the feasible region shrinks and may be empty.

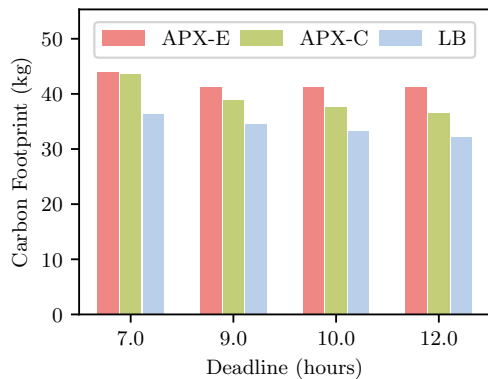


Fig. 1: Carbon footprint of different baselines with respect to the deadline.

solutions of both APX-E and APX-C incur less carbon footprint, but our scheme APX-C gives larger carbon footprint reduction and its performance is closer to the lower bound. In particular, when the deadline is set to 12 hours, our proposed APX-C algorithm can achieve an 11% reduction in carbon footprint compared to APX-E. These findings highlight that an energy-efficient solution, such as that produced by APX-E, may not necessarily optimize carbon footprint since it does not consider the carbon intensity during charging. It is therefore imperative to implement the intelligent carbon footprint aware design, which can result in substantial carbon reduction as shown in Fig. 1.

## V. CONCLUDING REMARKS

We address the problem of optimizing carbon footprint (CFO) for sustainable and timely transportation of E-trucks. We prove that even finding a feasible solution for this problem is NP-hard. We then develop a  $(1 + \epsilon_F, 1 + \epsilon_\beta)$  bi-criteria approximation algorithm that achieves a carbon footprint within a ratio of  $(1 + \epsilon_F)$  to the minimum with no deadline violation and at most a ratio of  $(1 + \epsilon_\beta)$  battery capacity violation. Through simulations, we show that our algorithm can achieve up to an 11% reduction in carbon footprint compared to baseline alternatives. We remark that the method used to design our algorithm in this paper can also be applied to develop a distinct approximation algorithm that avoids violating battery capacity but may violate deadlines. Further, the formulation and approach can be extended to address other problems, such as minimizing fuel costs for ICE-truck transportation that requires gas refills under geographically and temporally varying gas prices.

## ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their insightful comments that help improve this work. The work presented in this paper was supported in part by a General Research Fund from Research Grants Council, Hong Kong (Project No. 14207520).

## REFERENCES

[1] S. Davis and R. G. Boundy, "Transportation Energy Data Book: Edition 39," Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States), Tech. Rep., 2021. [Online]. Available: <https://www.osti.gov/biblio/1767864>

[2] U.S. Energy Information Administration (EIA), "June 2021 monthly energy review," Tech. Rep., 2021.

[3] D. Maji, P. Shenoy, and R. K. Sitaraman, "Carboncast: Multi-day forecasting of grid carbon intensity," in *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (ACM BuildSys 2022)*, New York, NY, USA, 2022, p. 198–207.

[4] B. H. Ashby, *Protecting perishable foods during transport by truck*. US Department of Agriculture, Office of Transportation, 1987, no. 669.

[5] L. Deng, M. H. Hajiesmaili, M. Chen, and H. Zeng, "Energy-efficient timely transportation of long-haul heavy-duty trucks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, pp. 2099–2113, 2017.

[6] Q. Liu, H. Zeng, and M. Chen, "Energy-efficient timely truck transportation for geographically-dispersed tasks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5148–5159, 2020.

[7] W. Xu, Q. Liu, M. Chen, and H. Zeng, "Ride the tide of traffic conditions: Opportunistic driving improves energy efficiency of timely truck transportation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–17, 2023.

[8] J. Su, M. Chen, and H. Zeng, "Energy efficient timely transportation: A comparative study of internal combustion trucks and electric trucks," in *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (ACM BuildSys 2021)*, New York, NY, USA, 2021, p. 224–225.

[9] Y. Zhang, X. Qu, and L. Tong, "Optimal eco-driving control of autonomous and electric trucks in adaptation to highway topography: Energy minimization and battery life extension," *IEEE Transactions on Transportation Electrification*, vol. 8, no. 2, pp. 2149–2163, 2022.

[10] Y. Liu, H. S. Seah, and G. Shou, "Constrained energy-efficient routing in time-aware road networks," *GeoInformatica*, vol. 21, pp. 89–117, 2017.

[11] A. Cela, T. Jurik, R. Hamouche, R. Natowicz, A. Reama, S.-I. Niculescu, and J. Julien, "Energy Optimal Real-Time Navigation System," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, pp. 66–79, 2014.

[12] M. Baum, J. Dibbelt, D. Wagner, and T. Zündorf, "Modeling and Engineering Constrained Shortest Path Algorithms for Battery Electric Vehicles," *Transportation Science*, vol. 54, pp. 1571–1600, 2020.

[13] M. W. Fontana, "Optimal Routes for Electric Vehicles Facing Uncertainty, Congestion, and Energy Constraints," Ph.D. dissertation, Massachusetts Institute of Technology, 2013.

[14] M. Strehler, S. Merting, and C. Schwan, "Energy-efficient shortest routes for electric and hybrid vehicles," *Transportation Research Part B: Methodological*, vol. 103, pp. 111–135, 2017.

[15] J. Su, Q. Lin, and M. Chen, "Follow the sun and go with the wind: Carbon footprint optimized timely e-truck transportation," in *Proceedings of the 14th ACM International Conference on Future Energy Systems*, ser. e-Energy '23, 2023, p. 159–171.

[16] J. Su, Q. Lin, M. Chen, and H. Zeng, "Minimizing carbon footprint for timely e-truck transportation: Hardness and approximation algorithm," 2023, arXiv:2308.09866.

[17] M. Baum, J. Dibbelt, A. Gamsa, D. Wagner, and T. Zündorf, "Shortest Feasible Paths with Charging Stops for Battery Electric Vehicles," *Transportation Science*, vol. 53, pp. 1627–1655, 2019.

[18] M. Horváth and T. Kis, "Multi-criteria approximation schemes for the resource constrained shortest path problem," *Optimization Letters*, vol. 12, no. 3, pp. 475–483, 2018.

[19] J. Teresco. Map-based Educational Tools for Algorithm Learning (METAL) project. [Online]. Available: <https://courses.teresco.org/metal/>

[20] H.-L. Hwang, S. Hargrove, S.-M. Chin, D. W. Wilson, H. Lim, J. Chen, R. Taylor, B. Peterson, and D. Davidson, "The freight analysis framework version 4 (faf4)-building the faf4 regional database: Data sources and estimation methodologies," Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), Tech. Rep., 2016.

[21] J. D. Gonder, A. D. Brooker, E. W. Wood, and M. Moniot, "Future automotive systems technology simulator (fastsim) validation report," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2018.