Minimizing Energy Use of Mixed-Fleet Public Transit for Fixed-Route Service

Amutheezan Sivagnanam[1], Afiya Ayman[1], Michael Wilbur[2], Philip Pugliese[3], Abhishek Dubey[2], Aron Laszka[1]
[1]University of Houston, [2]Vanderbilt University, [3]Chattanooga Area Regional Transportation Authority

Abstract

Affordable public transit services are crucial for communities since they enable residents to access employment, education, and other services. Unfortunately, transit services that provide wide coverage tend to suffer from relatively low utilization, which results in high fuel usage per passenger per mile, leading to high operating costs and environmental impact. Electric vehicles (EVs) can reduce energy costs and environmental impact, but most public transit agencies have to employ them in combination with conventional, internal-combustion engine vehicles due to the high upfront costs of EVs. To make the best use of such a mixed fleet of vehicles, transit agencies need to optimize route assignments and charging schedules, which presents a challenging problem for large transit networks. We introduce a novel problem formulation to minimize fuel and electricity use by assigning vehicles to transit trips and scheduling them for charging, while serving an existing fixed-route transit schedule. We present an integer program for optimal assignment and scheduling, and we propose polynomial-time heuristic and meta-heuristic algorithms for larger networks. We evaluate our algorithms on the public transit service of Chattanooga, TN using operational data collected from transit vehicles. Our results show that the proposed algorithms are scalable and can reduce energy use and, hence, environmental impact and operational costs. For Chattanooga, the proposed algorithms can save \$145,635 in energy costs and 576.7 metric tons of CO_2 emission annually.

1 Introduction

Affordable public transit services are the backbones of many communities, providing diverse groups of people with access to employment, education, and other services. Affordable transit services are especially important in low-income neighborhoods where residents might not be able to afford personal vehicles. However, transit services that provide wide and equitable coverage tend to suffer from low utilization—compared to concentrating service into a few high-density areas—which leads to higher fuel usage per passenger per mile. This in turn results in higher operating costs, which threatens affordability—a problem that has recently been exacerbated by the COVID-19 pandemic.

Further, low utilization also leads to high environmental impact per passenger per mile. In the U.S., 28% of total energy use is for transportation (EIA 2019). While public transit services can be very energy-efficient compared to personal vehicles, their environmental impact is significant nonetheless. For example, bus transit services in the U.S. may be responsible for up to 21.1 million metric tons of CO_2 (EPA 2020b) emission every year.

Electric vehicles (EVs) can have much lower operating costs and lower environmental impact during operation than comparable internal combustion engine vehicles (ICEVs), especially in urban areas. Unfortunately, EVs are also much more expensive than ICEVs: typically, diesel transit buses cost less than \$500K, while electric ones cost more than \$700K, or close to around \$1M with charging infrastructure. As a result, many public transit agencies can afford only mixed fleets of transit vehicles, which may consist of EVs, hybrid vehicles (HEVs), and ICEVs.

Public transit agencies that operate such mixed fleets of vehicles face a challenging optimization problem. First, they need to decide which vehicles are assigned to serving which transit trips. Since the advantage of EVs over ICEVs varies depending on the route and time of day (e.g., advantage of EVs is higher in slower traffic with frequent stops, and lower on highways), the assignment can have a significant impact on energy use and, hence, on costs and environmental impact. Second, transit agencies need to schedule when to charge electric vehicles because EVs have limited battery capacity and driving range, and may need to be recharged during the day between serving transit trips. Since agencies often have limited charging capabilities (e.g., limited number of charging poles, or limited maximum power to avoid high peak loads on the electric grid), charging constraints can significantly increase the complexity of the assignment and scheduling problem.

Contributions: While an increasing number of transit agencies face these problems, there exist no practical solutions to the best of our knowledge. In this paper, we present a novel problem formulation and algorithms for assigning a mixed fleet of transit vehicles to trips and for scheduling the charging of electric vehicles. We developed this problem formulation in collaboration with the Chattanooga Area Regional Transportation Authority (CARTA), the public transit agency of Chattanooga, TN, which operates a fleet of EVs, HEVs, and ICEVs. To solve the problem, we introduce an integer program as well as greedy and simulated annealing algorithms. We evaluate these algorithms using operational data collected from CARTA (e.g., vehicle energy consumption data, transit routes and schedules) and from other sources (e.g., elevation and street maps). Based on our numerical results, the proposed algorithms can reduce energy costs by up to $145,635 and CO_2 emissions by up to 576.7 metric tons annually for CARTA. We will make all data and the implementation of our algorithms publicly available (also attached as appendix to this submission).

Our problem formulation applies to a wide range of public transit agencies that operate fixed-route services. Our objective is to minimize energy consumption (i.e., fuel and electricity use), which leads to lower operating costs and environmental impact—as demonstrated by our numerical results. Our problem formulation considers assigning and scheduling for a single day (it may be applied to any number of consecutive days one-by-one), and allows any physically possible re-assignment during the day. Our formulation also allows capturing additional constraints on charging; for example, CARTA aims to charge only one vehicle at a time to avoid demand charges from the electric utility.

Organization: In Section 2, we describe our model and problem formulation. In Section 3, we introduce a mixed-integer program as well as greedy and simulated annealing algorithms. In Section 4, we provide numerical results based on real-world data from CARTA. In Section 5, we present a brief overview of related work. Finally, in Section 6, we summarize our findings and outline future work.

2 Transit Model and Problem Formulation

Vehicles We consider a transit agency that operates a set of *buses* $\mathcal{V}$. Note that we will use the terms *bus* and *vehicle* interchangeably. Each bus $v \in \mathcal{V}$ belongs to a *vehicle model* $M_v \in \mathcal{M}$, where $\mathcal{M}$ is the set of all vehicle models in operation. We divide the set of vehicle models $\mathcal{M}$ into two disjoint subsets: liquid-fuel models $\mathcal{M}^{\text{gas}}$ (e.g., diesel, hybrid), and electric models $\mathcal{M}^{\text{elec}}$. Based on discussions with CARTA, we assume that vehicles belonging to a liquid-fuel model can operate all day without refueling. On the other hand, vehicles belonging to an electric model have limited battery capacity, which might not be enough for a whole day. For each electric vehicle model $m \in \mathcal{M}^{\text{elec}}$, we let C_m denote the *battery capacity* of vehicles of model m.

Locations Locations $\mathcal{L}$ include bus stops, garages, and charging stations in the transit network.

Trips During the day, the agency has to serve a given set of *transit trips* $\mathcal{T}$ using its buses. Based on discussions with CARTA, we assume that locations and time schedules are fixed for every trip. A bus serving trip $t \in \mathcal{T}$ leaves from trip origin $t^{\text{origin}} \in \mathcal{L}$ at time t^{start} and arrives at destination $t^{\text{destination}} \in \mathcal{L}$ at time t^{end}. Between t^{origin} and $t^{\text{destination}}$, the bus must pass through a series of stops at fixed times; however, since we cannot re-assign a bus during a transit trip, the locations and times of these stops are inconsequential to our model. Finally, we assume that any bus may serve any trip. Note that it would be straightforward to extend our model and algorithms to consider constraints on which buses may serve a trip (e.g., based on passenger capacity).

Charging To charge its electric buses, the agency operates a set of *charging poles* $\mathcal{CP}$, which are typically located at bus garages or charging stations in practice. We let $cp^{\text{location}} \in \mathcal{L}$ denote the location of charging pole $cp \in \mathcal{CP}$.

For the sake of computational tractability, we use a discrete-time model to schedule charging, which divides time into uniform-length *time slots* $\mathcal{S}$. A time slot $s \in \mathcal{S}$ begins at s^{start} and ends at s^{end}. A charging pole $cp \in \mathcal{CP}$ can charge $P(cp, M_v)$ energy to one electric bus v in one time slot. We will refer to the combination of a charging pole $cp \in \mathcal{CP}$ and a time slot $s \in \mathcal{S}$ as a *charging slot* (cp, s); and we let $\mathcal{C} = \mathcal{CP} \times \mathcal{S}$ denote the set of charging slots.

Non-Service Trips Besides serving transit trips, buses may also need to drive between trips or charging poles. For example, if a bus has to serve a trip that starts from a location that is different from the destination of the previous trip, the bus first needs to drive to the origin of the subsequent trip. An electric bus may also need to drive to a charging pole after serving a transit trip to recharge, and then drive from the pole to the origin of the next transit trip. We will refer to these deadhead trips, which are driven outside of revenue service, as *non-service trips*. We let $T(l_1, l_2)$ denote the non-service trip from location $l_1 \in \mathcal{L}$ to $l_2 \in \mathcal{L}$; and we let $D(l_1, l_2)$ denote the time duration of this non-service trip.

2.1 Solution Space

Our primary goal is to assign a bus to each transit trip. Additionally, electric buses may also need to be assigned to charging slots to prevent them from running out of power.

Solution Representation We represent a solution as a *set of assignments* $\mathcal{A}$. For each trip $t \in \mathcal{T}$, a solution assigns exactly one bus $v \in \mathcal{V}$ to serve trip t; this assignment is represented by the relation $\langle v, t \rangle \in \mathcal{A}$. Secondly, each electric bus v must be charged before its battery state of charge drops below the safe level for operation. A solution assigns at most one electric bus v to each charging slot $(cp, s) \in \mathcal{C}$; this assignment is represented by the relation $\langle v, (cp, s) \rangle \in \mathcal{A}$. We assume that when a bus is assigned for charging, it remains at the charging pole for the entire duration of the corresponding time slot.

Constraints If a bus v is assigned to serve an earlier transit trip t_1 and a later trip t_2, then the duration of the non-service trip from $t_1^{\text{destination}}$ to t_2^{origin} must be less than or equal to the time between t_1^{end} and t_2^{start}. Otherwise, it would not be possible to serve t_2 on time. We formulate this constraint as:

$$\forall t_1, t_2 \in \mathcal{T};\ t_1^{\text{start}} \leq t_2^{\text{start}};\ \langle v, t_1 \rangle \in \mathcal{A};\ \langle v, t_2 \rangle \in \mathcal{A}: \quad t_1^{\text{end}} + D(t_1^{\text{destination}}, t_2^{\text{origin}}) \leq t_2^{\text{start}} \tag{1}$$

Note that if the constraint is satisfied by every pair of consecutive trips assigned to a bus, then it is also satisfied by every pair of non-consecutive trips assigned to the bus.

We need to formulate similar constraints for non-service trips to, from, and between charging slots:

$$\forall t \in \mathcal{T}; (cp, s) \in \mathcal{C}; t^{\text{start}} \leq s^{\text{start}}; \langle v, t\rangle, \langle v, (cp, s)\rangle \in \mathcal{A}: \\ t^{\text{end}} + D(t^{\text{destination}}, cp^{\text{location}}) \leq s^{\text{start}} \quad (2)$$

$$\forall t \in \mathcal{T}; (cp, s) \in \mathcal{C}; t^{\text{start}} \geq s^{\text{start}}; \langle v, t\rangle, \langle v, (cp, s)\rangle \in \mathcal{A}: \\ s^{\text{end}} + D(cp^{\text{location}}, t^{\text{origin}}) \leq t^{\text{start}} \quad (3)$$

$$\forall (cp_1, s_1), (cp_2, s_2) \in \mathcal{C}; s_1^{\text{start}} \leq s_2^{\text{start}}; \langle v, (cp_1, s_1)\rangle, \langle v, (cp_2, s_2)\rangle \in \mathcal{A}: \\ s_1^{\text{end}} + D(cp_1^{\text{location}}, cp_2^{\text{location}}) \leq s_2^{\text{start}} \quad (4)$$

We also need to ensure that electric buses never run out of power. First, we let $\mathcal{N}(\mathcal{A}, v, s)$ denote the set of all non-service trips that bus v needs to complete by the end of time slot s according to the set of assignments $\mathcal{A}$. In other words, $\mathcal{N}(\mathcal{A}, v, s)$ is the set of all necessary non-service trips to the origins of transit trips that start by s^{end} and to the locations of charging slots that start by s^{end}. Next, we let $E(v, t)$ denote the amount of energy used by bus v to drive a transit or non-service trip t. Then, we let $e(\mathcal{A}, v, s)$ be the amount of energy used by bus v for all trips completed by the end of time slot s:

$$e(\mathcal{A}, v, s) = \sum_{t \in \mathcal{N}(\mathcal{A}, v, s)} E(v, t) + \sum_{t \in \mathcal{T},\, \langle v, t\rangle \in \mathcal{A},\, t^{\text{end}} \leq s^{\text{end}}} E(v, t) \quad (5)$$

Similarly, we let $r(\mathcal{A}, v, s)$ be the amount of energy charged to bus v by the end of time slot s:

$$r(\mathcal{A}, v, s) = \sum_{(cp, \hat{s}) \in \mathcal{C},\, \langle v, (cp, \hat{s})\rangle \in \mathcal{A},\, \hat{s}^{\text{end}} \leq s^{\text{end}}} P(cp, M_v) \quad (6)$$

Since a bus can be assigned for charging only to complete time slots, the minima and maxima of its battery level will be reached at the end of time slots. Therefore, we can express the constraint that the battery level of bus v must always remain between 0 and the battery capacity C_{M_v} as

$$\forall v \in \mathcal{V}, \forall s \in \mathcal{S}: \; 0 < r(\mathcal{A}, v, s) - e(\mathcal{A}, v, s) \leq C_{M_v}. \quad (7)$$

Note that we can give vehicles an initial battery charge by adding "virtual" charging slots before the day starts.

2.2 Objective

Our objective is to minimize the energy use of the transit vehicles. We can use this objective to minimize both environmental impact and operating costs by imposing the appropriate cost factors on the energy use of liquid-fuel and electric vehicles. We let K^{gas} and K^{elec} denote the unit costs of energy use for liquid-fuel and electric vehicles, respectively. Then, by applying the earlier notation $e(\mathcal{A}, v, s)$ to all vehicles, we can express our objective as

$$\min_{\mathcal{A}} \sum_{v \in \mathcal{V}:\, M_v \in \mathcal{M}^{\text{gas}}} K^{\text{gas}} \cdot e(\mathcal{A}, v, s_\infty) + \sum_{v \in \mathcal{V}:\, M_v \in \mathcal{M}^{\text{elec}}} K^{\text{elec}} \cdot e(\mathcal{A}, v, s_\infty) \quad (8)$$

where s_∞ denotes the last time slot of the day.

3 Algorithms

Since this optimization problem is computationally hard (see proof sketch in an online appendix (Sivagnanam et al. 2020)), we first present an integer program to find optimal solutions for smaller instances (Section 3.1). Then, we introduce efficient greedy (Section 3.2) and simulated annealing algorithms (Section 3.3), which scale well for larger instances. Due to lack of space, we include less important subroutines in the online appendix (Sivagnanam et al. 2020).

3.1 Integer Program

Variables Our integer program has five sets of variables. Three of them are binary to indicate assignments and non-service trips. First, $a_{v,t} = 1$ (or 0) indicates that trip t is assigned to bus v (or that it is not). Second, $a_{v,(cp,s)} = 1$ (or 0) indicates that charging slot (cp, s) is assigned to electric bus v (or not). Third, $m_{v,x_1,x_2} = 1$ (or 0) indicates that bus v takes the non-service trip between a pair x_1 and x_2 of transit trips and/or charging slots (or not). Note that for requiring non-service trips (see Equations (1) to (4)), we will treat transit trips and charging slots similarly since they induce analogous constraints. There are also two sets of continuous variables. First, $c_s^v \in [0, C_{M_v}]$ represents the amount of energy charged to electric bus v in time slot s. Second, $e_s^v \in [0, C_{M_v}]$ represents the battery level of electric bus v at the start of time slot s (considering energy use only for trips that have ended by that time). Due to the continuous variables, our program is a mixed-integer program.

Constraints First, we ensure that every transit trip is served by exactly one bus:

$$\forall t \in \mathcal{T}: \; \sum_{v \in \mathcal{V}} a_{v,t} = 1$$

Second, we ensure that each charging slot is assigned at most one electric vehicle:

$$\forall (cp, s) \in \mathcal{C}: \sum_{\forall v \in \mathcal{V}:\, M_v \in \mathcal{M}^{\text{elec}}} a_{v,(cp,s)} \leq 1$$

Next, we ensure that Equations (1) to (4) are satisfied. We let $F(x_1, x_2)$ be *true* if a pair x_1, x_2 of transit trips and/or charging slots satisfies the applicable one from Equations (1) to (4); and let it be *false* otherwise. Then, we can express these constraints as follows:

$$\forall v \in \mathcal{V}, \forall x_1, x_2, \neg F(x_1, x_2): \; a_{v,x_1} + a_{v,x_2} \leq 1$$

When a bus v is assigned to both x_1 and x_2, but it is not assigned to any other transit trips or charging slots in between (i.e., if x_1 and x_2 are consecutive assignments), then bus v needs to take a non-service trip:

$$m_{v,x_1,x_2} \geq a_{v,x_1} + a_{v,x_2} - 1 - \sum_{x \in \mathcal{T} \cup \mathcal{C}:\, x_1^{\text{start}} \leq x^{\text{start}} \leq x_2^{\text{start}}} a_{v,x}$$

Note that if x_1 ends at the same location where x_2 starts, then the non-service trip will take zero time and energy.

Finally, we ensure that the battery levels of electric buses remain between zero and capacity. First, for each slot s and electric bus v, the amount of energy charged c_s^v is subject to

$$c_s^v \leq \sum_{(cp,s)\in\mathcal{C}} a_{v,(cp,s)} \cdot P(cp, M_v).$$

Then, for the $(n+1)th$ time slot s_{n+1} and for an electric bus v, we can express variable $e^v_{s_{n+1}}$ as

$$\begin{aligned} e^v_{s_{n+1}} = e^v_{s_n} + c^v_{s_n} - &\sum_{t\in\mathcal{T}:\, s_n^{\text{start}} < t^{\text{end}} \leq s_n^{\text{end}}} a_{v,t} \cdot E(v,t) \\ - &\sum_{x_1,x_2:\, s_n^{\text{start}} < x_2^{\text{start}} \leq s_n^{\text{end}}} m_{v,x_1,x_2} \cdot E(v, T(x_1,x_2)) \end{aligned}$$

where s_n is the $(n)th$ slot. Note that since $e^v_s \in [0, C_{M_v}]$, this constraint ensures that Equation (7) is satisfied.

Objective We can express Equation (8) as minimizing

$$\sum_{v\in\mathcal{V}} K^{M_v} \left[\sum_{t\in\mathcal{T}} a_{v,t} \cdot E(v,t) + \sum_{x_1,x_2\in\mathcal{T}\cup\mathcal{C}} m_{v,x_1,x_2} \cdot E(v, T(x_1,x_2)) \right]$$

where K^{M_v} is K^{elec} if $M_v \in \mathcal{M}^{\text{elec}}$ and K^{gas} otherwise.

Complexity The integer program contains both variables and constraints in the order of $\mathcal{O}(|\mathcal{V}| \cdot |\mathcal{T}|^2)$.

3.2 Greedy Algorithm

Next, we introduce a polynomial-time greedy algorithm. The key idea of this algorithm is to choose between assignments based on a *biased cost* instead of the actual cost.

Algorithm 1: BiasedCost$(\mathcal{A}, v, x, \alpha)$

```
if x ∈ T then
    cost ← E(v, x)
else
    cost ← 0
Earlier =
  { x̂ ∈ T ∪ C | ⟨v, x̂⟩ ∈ A ∧ x̂^end ≤ x^start }
if Earlier ≠ ∅ then
    x_prev = argmax_{x̂ ∈ Earlier} x̂^end
    m_prev ← T(x_prev^destination, x^origin)
    cost ← cost + E(v, m_prev) + α · (x^start − x_prev^end)
Later = { x̂ ∈ T ∪ C | ⟨v, x̂⟩ ∈ A ∧ x^end ≤ x̂^start }
if Later ≠ ∅ then
    x_next = argmin_{x̂ ∈ Earlier} x̂^start
    m_next ← T(x^destination, x_next^origin)
    cost ← cost + E(v, m_next) + α · (x^end − x_next^start)
Result: cost
```

Algorithm 2: Greedy$(\mathcal{V}, \mathcal{T}, \mathcal{C}, \alpha)$

```
A ← ∅
E ←
  {⟨v, t⟩ ↦ BiasedCost(A, v, t, α) | v ∈ V, t ∈ T}
while |T| > 0 and min[E] ≠ ∞ do
    MinimumCostAssignments ← argmin(E)
    ⟨v, t⟩ ← first(MinimumCostAssignments)
    A ← A ∪ {⟨v, t⟩}
    T ← T \ {t}
    E, A ← Update(A, T, C, E, v, t, α)
    ▷ update cost values E and and add charging slots
      to the assignments as necessary
Result: A
```

Biased Energy Cost Our greedy approach uses Algorithm 1 to compute a *biased energy cost* of assigning a bus v to a transit trip or charging slot x. If x is a transit trip (i.e., $x \in \mathcal{T}$), then the base cost of the assignment is $E(v, x)$. If x is a charging slot, then the base cost is zero. To compute the actual cost, the algorithm checks if bus v is already assigned to any earlier (or later) transit trips or charging slots. If it is, then it factors in the cost of the moving trip m_{prev} (and m_{next}) from the preceding (and to the following) assignment x_{prev} (and x_{next}). Finally, the algorithm adds a bias to the actual cost based on the waiting time between x^{end}_{prev} and x^{start} (if x_{prev} exists) and between x^{end} and x^{end}_{next} (if x_{next} exists). By adding these waiting times to the cost with an appropriate factor $\alpha > 0$, we nudge the greedy selection towards increasing bus utilization and minimizing layovers. The time complexity of this algorithm is $\mathcal{O}(|\mathcal{T} \cup \mathcal{C}|)$.

Algorithm 2 shows our iterative greedy approach for assigning transit trips and charging slots to buses. The algorithm begins by computing the biased assignment cost for each pair of a bus v and transit trip t using **BiasedCost**$(\mathcal{A}, v, t, \alpha)$. Starting with an empty set $\mathcal{A} = \emptyset$, the algorithm then iteratively adds assignments $(v, t) \in \mathcal{V} \times \mathcal{T}$ to the set, always choosing an assignment with the lowest biased cost **BiasedCost**$(\mathcal{A}, v, t, \alpha)$ (breaking ties arbitrarily). After each iteration, the biased costs $\mathcal{E}$ for the chosen vehicle v are updated by **Update**, which also adds charging slot assignments as necessary (see the online appendix (Sivagnanam et al. 2020)). The algorithm terminates once all trips are assigned (or if it fails to find a solution). The time complexity of **Update** is $\mathcal{O}(|\mathcal{T}| \cdot |\mathcal{V}| + |\mathcal{T}| \cdot |\mathcal{C}| \cdot |\mathcal{X}| \ln |\mathcal{X}|)$, where $\mathcal{X} = \mathcal{T} \cup \mathcal{C}$. Since typically $|\mathcal{T}| \gg |\mathcal{V}|$, the complexity can be simplify into $\mathcal{O}(|\mathcal{T}| \cdot |\mathcal{C}| \cdot |\mathcal{X}| \ln |\mathcal{X}|)$. Accordingly, the time complexity of the greedy algorithm is $\mathcal{O}(|\mathcal{T}|^2 \cdot |\mathcal{C}| \cdot |\mathcal{X}| \ln |\mathcal{X}|)$.

3.3 Simulated Annealing

Finally, we introduce a simulated annealing algorithm, which improves upon the output of the greedy algorithm using iterative random search. Starting from a greedy solution, the search takes significantly less time than starting from random solution. The key element of this algorithm is choosing a random "neighboring" solution in each iteration.

Algorithm 3: RandomNeighbor$(\mathcal{A}, p_{swap})$

$NumberOfSwaps \leftarrow \mathbf{max}\{1, |\mathcal{A}| \cdot p_{swap}\}$
for $1, \ldots, NumberOfSwaps$ **do**
  $v_1, v_2 \leftarrow$ **UniformRandom**$(\mathcal{V})$
  $\mathcal{T}_1 \leftarrow \{t \in \mathcal{T} \mid \langle v_1, t \rangle \in \mathcal{A}\}$
  $\mathcal{T}_2 \leftarrow \{t \in \mathcal{T} \mid \langle v_2, t \rangle \in \mathcal{A}\}$
  $SplitTime \leftarrow$ **UniformRandom**$([s_1^{start}, s_\infty^{end}])$
  $\mathcal{T}_{noswap}, \mathcal{T}_{swap} =$
   SplitByStartTime$(\{\mathcal{T}_1, \mathcal{T}_2\}, SplitTime)$
  for $t \in \mathcal{T}_{swap}$ **do**
   if $t \in \mathcal{T}_1$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\langle v_1, t \rangle\} \cup \{\langle v_2, t \rangle\}$
   else
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\langle v_2, t \rangle\} \cup \{\langle v_1, t \rangle\}$
Result: $\mathcal{A}$

Random Neighbor Simulated annealing uses Algorithm 3 to generate a random neighbor for a candidate solution $\mathcal{A}$. The algorithm first chooses two vehicles v_1, v_2 at random from $\mathcal{V}$. Next, the algorithm enumerates all the trips $\mathcal{T}_1$ and $\mathcal{T}_2$ that are assigned to these vehicles in solution $\mathcal{A}$, chooses a random point in time *SplitTime* during the day, and splits all these trips into two sets $\mathcal{T}_{noswap}$ and $\mathcal{T}_{swap}$ based on the start times of the trip and *SplitTime*. Finally, the algorithms swaps all the trips in $\mathcal{T}_{swap}$ between v_1 and v_2 (i.e., trips that were assigned to v_1 are re-assigned to v_2 and vice versa). The algorithm then repeats this process from the beginning until the desired number of swap operations $|\mathcal{A}| \cdot p_{swap}$ is reached. The time complexity of this algorithm is $\mathcal{O}(|\mathcal{A}| \cdot (|\mathcal{T}| + |\mathcal{V}|))$.

Algorithm 4: Simulated Annealing$(\mathcal{V}, \mathcal{T}, \mathcal{C}, \alpha, k_{max}, p_{start}, p_{end}, p_{swap})$

$\mathcal{A} \leftarrow$ **Greedy**$(\mathcal{V}, \mathcal{T}, \mathcal{C}, \alpha)$
$Solutions \leftarrow \{\mathcal{A}\}$
$\tau_{start} \leftarrow \frac{-1}{\ln p_{start}}$
$\tau_{end} \leftarrow \frac{-1}{\ln p_{end}}$
$\tau_{rate} \leftarrow \left(\frac{\tau_{end}}{\tau_{start}}\right)^{\frac{1}{k_{max}-1}}$
$\tau_k \leftarrow \tau_{start}$
$\delta_{avg} \leftarrow 0$
for $k = 1, 2, \ldots, k_{max}$ **do**
  $\mathcal{A}' \leftarrow$ **RandomNeighbor**$(\mathcal{A}, p_{swap})$
  $\delta_e \leftarrow$ **Cost**$(\mathcal{A}')$ − **Cost**$(\mathcal{A})$
  if $k = 1$ **then**
   $\delta_{avg} \leftarrow \delta_e$
  $AcceptProbability \leftarrow \exp\left(\frac{-\delta_e}{\delta_{avg} \cdot \tau_k}\right)$
  if Cost$(\mathcal{A}') <$ **Cost**$(\mathcal{A})$ **or** $AcceptProbability >$
   UniformRandom$([0, 1])$ **then**
   $\mathcal{A} \leftarrow \mathcal{A}'$
   $\delta_{avg} \leftarrow \delta_{avg} + \frac{\delta_e - \delta_{avg}}{|Solutions|}$
   $Solutions \leftarrow Solutions \cup \{\mathcal{A}\}$
  $\tau_k \leftarrow \tau_k \cdot \tau_{rate}$
$\mathcal{A}^* \leftarrow \mathrm{argmin}_{\mathcal{A}' \in Solutions}$ **Cost**$(\mathcal{A}')$
Result: $\mathcal{A}^*$

Algorithm 4 shows our simulated annealing approach. First, the algorithm obtains an initial solution $\mathcal{A}$ using Algorithm 2. Then, it follows an iterative process. In each iteration, the algorithm obtains a random neighboring solution $\mathcal{A}'$ of the current solution $\mathcal{A}$ using **RandomNeighbor**. If the energy cost **Cost** (see Equation (8)) of $\mathcal{A}'$ is lower than the energy cost of $\mathcal{A}$, then the algorithm always accepts $\mathcal{A}'$ as the new solution. Otherwise, the algorithm computes the probability *AcceptProbability* of accepting it based on a decreasing temperature value τ_k and the cost difference between $\mathcal{A}'$ and $\mathcal{A}$, and then accepts $\mathcal{A}'$ at random. The algorithm terminates after a fixed number of iterations k_{max} and returns the best solution found up to that point. The time complexity of this algorithm is $\mathcal{O}(k_{max} \cdot |\mathcal{T}| \cdot (|\mathcal{T}| + |\mathcal{V}|))$.

4 Numerical Results

We evaluate our algorithms using data collected from CARTA. We will release the complete dataset as well as our implementation publicly.

4.1 Dataset

Public Transit Schedule We obtain the schedule of the transit agency in GTFS format, which includes all trips, time schedules, bus stop locations, etc. Trips are organized into 17 bus lines (i.e., bus routes) throughout the city. For our numerical evaluation, we consider trips served during weekdays (Monday to Friday) since these are the busiest days. Each weekday, the agency must serve around 850 trips using 3 electric buses of model BYD K9S, and 50 diesel and hybrid buses of various models.

Energy Use Prediction To estimate the energy usage of each transit and non-service trip, we use a neural network based prediction model, which we train on high-resolution historical data. CARTA has installed sensors on its mixed-fleet of vehicles, and it has been collecting data continuously for over a year at 1-second intervals from 3 electric, 41 diesel, and 6 hybrid buses. To train the predictor, we select 6 months of data from 3 electric vehicles (BYD K9S buses) and 3 diesel vehicles (2014 Gillig Phantom buses). We obtain 0.1 Hz timeseries data from onboard telemetry devices, which record location (GPS), odometer, battery current, voltage, and charge (for EVs), and fuel level and usage (for diesel). In total, we obtain around 6.6 million datapoints for electric buses and 1.1 million datapoints for diesel buses (fuel data was recorded less frequently).

We augment this dataset with additional features related to weather, road, and traffic conditions to improve our energy-use predictor. We incorporate hourly predictions of weather features, which are based on data collected using Dark Sky API (Sky 2019) at 5-minute intervals. Weather features include temperature, humidity, pressure, wind speed, and precipitation. We include road-condition features based on a street-level map of the city obtained from OpenStreetMap. We also include road gradients, which we compute along transit routes using an elevation map that is based on high-accuracy LiDAR data from the state government. Finally, we incorporate predictions of traffic conditions, which are based on data obtained using HERE Maps API (HERE 2020). Note that we present more detailed description of the dataset in (Ayman et al. 2021).

In total, we use 26 different features to train neural network models for energy prediction. Our neural network has one input, two hidden, and one output layer, all using sigmoid activation. We chose this architecture based on its accuracy after comparing it to various other regression models. We train a different prediction model for each vehicle model, which we then use to predict energy use for every trip. For 30-minute trips, the mean absolute percentage error of the energy predictor is 8.4% for diesel and 13.1% for electric vehicles. For an entire day of service, the error is only 1.5% and 2.6%, respectively.

Non-Service Trips Since non-service trips are not part of the transit schedule, we need to plan their routes and estimate their durations. For this, we use the Google Directions API, which we query for all 2,070 possible non-service trips (i.e., for every pair of locations in the network) for each 1-hour interval of a selected weekday from 5am to 11pm. The response to each query includes an estimated duration as well as a detailed route, which we combine with our other data sources and then feed into our energy-use predictors.

Charging Rate, Energy Costs, and CO_2 Emissions Electric buses of model BYD K9S have a battery capacity of 270 kWh, and the charging poles of the agency can charge a BYD K9S model bus at the rate of 65 kW/h. We consider 3 charging poles for our numerical evaluation. Based on data from the transit agency, we consider electricity cost to be $9.602 per 100 kWh and diesel cost to be $2.05 per gallon. Finally, based on data from EPA (EPA 2020a), we calculate CO_2 emissions for diesel vehicles as 8.887 kg/gallons and for electric vehicles as 0.707 kg/kWh.

4.2 Results

For all experiments, we set the length of time slots to be 1 hour. For experiments with small problem instances, we set the wait-time factor (see Algorithm 1) to $\alpha = 0.00004$ for electric buses and to $\alpha = 0.00002$ for liquid-fuel buses; swapping rate to $p_{swap} = 0.03$ (see Algorithm 3); simulated-annealing iterations, initial probability, and final probability to $k_{max} = 50,000$, $p_{start} = 0.1$, and $p_{end} = 0.07$, respectively (see Algorithm 4). For experiments with complete daily schedules, we set wait-time factor (see Algorithm 1) to $\alpha = 0.001$ for electric buses and to $\alpha = 0.0002$ for liquid-fuel buses; swapping rate to $p_{swap} = 0.05$; simulated-annealing iterations, initial probability, and final probability to $k_{max} = 50,000$, $p_{start} = 0.2$, and $p_{end} = 0.09$, respectively.

We found these to be optimal configuration based on a grid search of the parameter space. Due to the lack of space, we include these search results in an online appendix (Sivagnanam et al. 2020).

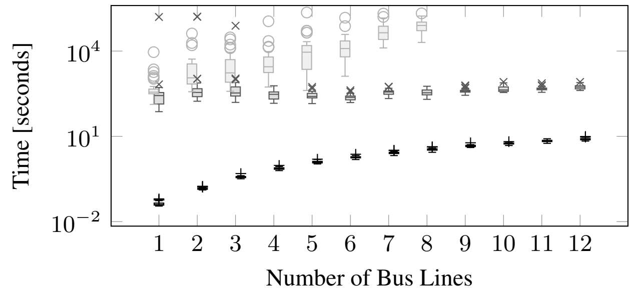

Figure 1: Computation times using integer program (■○), simulated annealing (■x), and greedy algorithm (■+). Please note the logarithmic scale on the vertical axis.

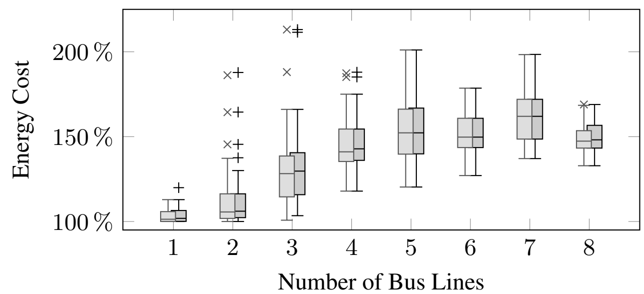

Figure 2: Energy cost for assignments using simulated annealing (■x) and the greedy algorithm (■+) compared to optimal assignments (found using the integer program).

Computational Performance We first study how well our algorithms scale with increasing problem sizes. To this end, we measure the computation times of our algorithms with 1 to 12 bus lines (selected from the real bus lines), and 10 selected trips for each line. For each case, we evaluate the algorithms on 35 different samples and present statistical results. For cases with 11 and 12 bus lines, we assign the entire vehicle fleet, which consists of 3 electric and 50 liquid-fuel buses. For cases with fewer bus lines, we assume that the agency has 3 electric buses but only 5 times as many liquid-fuel buses as bus lines. We solve the integer program (IP) using IBM CPLEX. We run all algorithms on a machine with a Xeon E5-2680 CPU, which has 28 cores, and 128 GB of RAM. Figure 1 shows the computation times for the IP, greedy, and simulated annealing. As expected, the time to solve the IP is significantly higher and increases rapidly with the number of lines, becoming infeasible at around 10 lines. On the the hand, the greedy and simulated annealing algorithms are orders of magnitude faster and scale well.

Solution Quality Next, we evaluate the performance of our algorithms with respect to solution quality, that is, energy cost. Note that we present CO_2 results in an online appendix (Sivagnanam et al. 2020). We use the exact same setting as in the previous experiment (Figure 1). Note that for larger instances, solving the IP is infeasible. Figure 2 shows that simulated annealing performs slightly better than greedy; however, neither perform as well as IP (which is op-

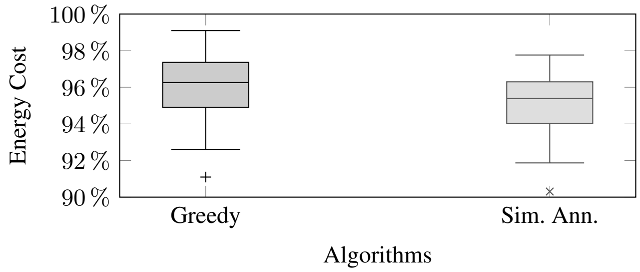

Figure 3: Energy costs for assignments using the greedy algorithm and simulated annealing for complete daily schedules, compared to existing real-world assignments.

timal). On the bright side, the cost ratio between IP and our heuristics remains within the range of 1.5 to 1.6 even for larger instances.

Comparison to Existing Assignments Finally, we compute assignments for the complete daily schedule of the agency using 3 electric and 50 liquid-fuel buses using greedy and simulated annealing algorithms. In Figure 3, we compare greedy and simulated annealing assignments with real-world assignments for 50 different sample days based on energy costs. Our results shows that both algorithms attain lower energy costs than existing real-world assignments. On average, real-world assignments cost $8187 with 35.58 metric tons of CO_2 emission, greedy approach costs $7863 with 34.33 metric tons of CO_2 emission, and simulated annealing algorithm costs $7788 with 34.00 metric tons of CO_2 emission. We were able to assign the full schedule using greedy algorithm in around 6 minutes; meanwhile, simulated annealing runs for around 8 hours (around 50,000 iterations). Since an agency might need to find a new assignment urgently (e.g., because some buses are unavailable due to maintenance), the greedy algorithm can be a better option.

5 Related Work

Previous research efforts predict energy consumption through simulation models using spatial and temporal data (e.g., Wang et al. (2018), Tian et al. (2016), Wang et al. (2017)) collect GPS data, bus stop data, bus transaction data, traffic data, and electricity consumption data). Unlike the previous works, we derive realistic energy estimates using our energy predictors based on vehicle locations, traffic, elevation, and weather data. Further, some works consider fixed energy cost and emission with respect to the miles traveled by the bus (e.g., Santos et al. (2016), Paul and Yamada (2014), Sassi, Cherif, and Oulamara (2015)). Li (2014) computes the distance of non-service trips using point-to-curve matching, then obtain optimal paths using Dijkstra's algorithm. But these assumptions limit applicability or performance for real-world implementation.

Researchers have applied various approaches such as genetic algorithm (Yang and Liu (2020), Sun et al. (2020), Santos et al. (2016)), simulated annealing (Zhou et al. (2020)), and column generation (Li (2014)) in the domain of energy-efficient bus scheduling with either liquid-fuel buses or electric buses. But only few research efforts (e.g., Santos et al. (2016), Zhou et al. (2020)) focused on transit networks operating mixed-fleets of buses.

Other researchers have applied approaches such as integer programming (Nageshrao, Jacob, and Wilkins (2017), Lotfi et al. (2020), Picarelli et al. (2020)), Markov decision processes (Wang et al. (2018)), greedy algorithms (Jefferies and Göhlich (2020)), genetic algorithms (Gao et al. (2018), Chao and Xiaohong (2013)), space-time networks (Olsen, Kliewer, and Wolbeck (2020)), and dynamic programming (Wang, Kang, and Liu (2020)) to optimally assign electric buses to charging stations.

Jahic, Eskander, and Schulz (2019) use preemptive, quasi-preemptive, and non-preemptive approaches to effectively handle the load in charging stations or garages. Since charging duration occupies a reasonable portion of the routine, Chao and Xiaohong (2013) propose a battery replacement technique, but this approach is inefficient for transit agencies operating with a few electric buses. Murphey et al. (2012) present the development of a machine learning framework for energy management optimization in an HEV, developing algorithms based on long- and short-term knowledge about the driving environment.

Some works propose solutions that reduce energy costs by changing bus schedules or routes (e.g., Hassold and Ceder (2014), Wang et al. (2018)), which can cause inconvenience to passengers, while Santos et al. (2016) schedule buses ensuring that service level is unchanged. Kliewer, Mellouli, and Suhl (2006), Kliewer, Gintner, and Suhl (2008), Li, Lo, and Xiao (2019) allow a bus to serve multiple lines instead of limiting it to a single line, which can reduce energy cost; we also implement a similar approach. Li, Lo, and Xiao (2019) group the trips as origin-destination pairs and assign them to vehicles, which also reduces energy costs. Liao, Liu, and Fu (2019) optimize the routing of conventional and electric vehicles using separate models; in contrast, we optimize mixed fleets of vehicles using a single, integrated model.

6 Conclusion

Due to the high upfront costs of EVs, many public transit agencies can only afford to operate mixed fleets of EVs, HEVs, and ICEVs. In this paper, we formulated the novel problem of minimizing operating costs and environmental impact through assigning trips and scheduling charging for mixed fleets of public transit vehicles; and we provided efficient greedy and simulated annealing algorithms. Based on real-world data from CARTA, we demonstrated that these algorithms scale well for larger instances and can provide significant savings in terms of energy costs and CO_2 emission. Even though our approaches perform better than existing real-world assignments, there remains a significant gap to optimal solutions (at least for smaller instances). In future work, we will strive to improve our algorithms to close this gap and to provide further saving to transit agencies. We are also publicly releasing our dataset to facilitate open research in this direction.

Acknowledgments

We thank the anonymous reviewers of this paper who provides their valuable feedback and suggestions. This material is based upon work supported by the Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE), under Award Number DE-EE0008467. Disclaimer: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This work was completed in part with resources provided by the Research Computing Data Core at the University of Houston and in part using cloud research credits provided by Google.

Ethics Statement

Our research did not involve any personally identifiable information.

Our objective formulation minimizes fuel and electricity usage (both measured as energy), both of which are scaled by factors (K^{gas} and K^{elec}). In the evaluation, we set these factors to the prices paid by the agency for fuel and electricity; hence, minimizing energy usage minimizes the agency's monetary cost. Please note that we could tweak these factors to consider other goals, e.g., minimize environmental impact by setting the factors to the environmental footprint of fuel and electric energy usage.

References

Ayman, A.; Sivagnanam, A.; Michael, W.; Pugliese, P.; Dubey, A.; and Laszka, A. 2021. Data-Driven Prediction and Optimization of Energy Use for Transit Fleets of Electric and ICE Vehicles. *ACM Transactions on Internet Technology* In press.

Chao, Z.; and Xiaohong, C. 2013. Optimizing battery electric bus transit vehicle scheduling with battery exchanging: Model and case study. *Procedia-Social and Behavioral Sciences* 96: 2725–2736.

EIA. 2019. U.S. Energy Information Administration: Use of energy explained – Energy use for transportation (2019). https://www.eia.gov/energyexplained/use-of-energy/transportation.php, Accessed: September 9th, 2020.

EPA. 2020a. Greenhouse Gases Equivalencies Calculator - Calculations and References. https://www.epa.gov/energy/greenhouse-gases-equivalencies-calculator-calculations-and-references, Accessed: September 9th, 2020.

EPA. 2020b. U.S. Transportation Sector Greenhouse Gas Emissions. https://nepis.epa.gov/Exe/ZyPDF.cgi?Dockey=P100ZK4P.pdf, Accessed: September 9th, 2020.

Gao, Y.; Guo, S.; Ren, J.; Zhao, Z.; Ehsan, A.; and Zheng, Y. 2018. An electric bus power consumption model and optimization of charging scheduling concerning multi-external factors. *Energies* 11(8): 2060.

Hassold, S.; and Ceder, A. 2014. Improving Energy Efficiency of Public Transport Bus Services by Using Multiple Vehicle Types. *Transportation Research Record* 2415(1): 65–71.

HERE. 2020. HERE Maps API. https://developer.here.com/, Accessed: January 21st, 2020.

Jahic, A.; Eskander, M.; and Schulz, D. 2019. Preemptive vs. non-preemptive charging schedule for large-scale electric bus depots. In *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, 1–5. IEEE.

Jefferies, D.; and Göhlich, D. 2020. A Comprehensive TCO Evaluation Method for Electric Bus Systems Based on Discrete-Event Simulation Including Bus Scheduling and Charging Infrastructure Optimisation. *World Electric Vehicle Journal* 11(3): 56.

Kliewer, N.; Gintner, V.; and Suhl, L. 2008. Line change considerations within a time-space network based multi-depot bus scheduling model. In *Computer-aided Systems in Public Transport*, 57–70. Springer.

Kliewer, N.; Mellouli, T.; and Suhl, L. 2006. A time–space network based exact optimization model for multi-depot bus scheduling. *European journal of operational research* 175(3): 1616–1627.

Li, J.-Q. 2014. Transit bus scheduling with limited energy. *Transportation Science* 48(4): 521–539.

Li, L.; Lo, H. K.; and Xiao, F. 2019. Mixed bus fleet scheduling under range and refueling constraints. *Transportation Research Part C: Emerging Technologies* 104: 443–462.

Liao, W.; Liu, L.; and Fu, J. 2019. A comparative study on the routing problem of electric and fuel vehicles considering carbon trading. *International Journal of Environmental Research and Public Health* 16(17): 3120.

Lotfi, M.; Pereira, P.; Paterakis, N.; Gabbar, H. A.; and Catalão, J. P. S. 2020. Optimizing Charging Infrastructures of Electric Bus Routes to Minimize Total Ownership Cost. In *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, 1–6.

Murphey, Y. L.; Park, J.; Chen, Z.; Kuang, M. L.; Masrur, M. A.; and Phillips, A. M. 2012. Intelligent hybrid vehicle power control—Part I: Machine learning of optimal vehicle power. *IEEE Transactions on Vehicular Technology* 61(8): 3519–3530.

Nageshrao, S. P.; Jacob, J.; and Wilkins, S. 2017. Charging cost optimization for EV buses using neural network based energy predictor. *IFAC-PapersOnLine* 50(1): 5947–5952.

Olsen, N.; Kliewer, N.; and Wolbeck, L. 2020. A study on flow decomposition methods for scheduling of electric buses in public transport based on aggregated time–space network models. *Central European Journal of Operations Research* 1–37.

Paul, T.; and Yamada, H. 2014. Operation and charging scheduling of electric buses in a city bus route network. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2780–2786. IEEE.

Picarelli, E.; Rinaldi, M.; D'Ariano, A.; and Viti, F. 2020. Model and Solution Methods for the Mixed-Fleet Multi-Terminal Bus Scheduling Problem. *Transportation Research Procedia* 47: 275–282.

Santos, D.; Kokkinogenis, Z.; de Sousa, J. F.; Perrotta, D.; and Rossetti, R. J. 2016. Towards the integration of electric buses in conventional bus fleets. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 88–93. IEEE.

Sassi, O.; Cherif, W. R.; and Oulamara, A. 2015. Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs. *International Journal of Mathematical and Computational Sciences* 9(3): 171–181.

Sivagnanam, A.; Ayman, A.; Wilbur, M.; Pugliese, P.; Dubey, A.; and Laszka, A. 2020. Minimizing Energy Use of Mixed-Fleet Public Transit for Fixed-Route Service. *arXiv preprint arXiv:2004.05146* .

Sky, D. 2019. API Documentation. https://darksky.net/dev/docs, Accessed: January 21st, 2020.

Sun, Q.; Chien, S.; Hu, D.; Chen, G.; and Jiang, R.-S. 2020. Optimizing Multi-Terminal Customized Bus Service With Mixed Fleet. *IEEE Access* 8: 156456–156469.

Tian, Z.; Jung, T.; Wang, Y.; Zhang, F.; Tu, L.; Xu, C.; Tian, C.; and Li, X.-Y. 2016. Real-time charging station recommendation system for electric-vehicle taxis. *IEEE Transactions on Intelligent Transportation Systems* 17(11): 3098–3109.

Wang, G.; Xie, X.; Zhang, F.; Liu, Y.; and Zhang, D. 2018. bCharge: Data-Driven Real-Time Charging Scheduling for Large-Scale Electric Bus Fleets. In *2018 IEEE Real-Time Systems Symposium (RTSS)*, 45–55. IEEE.

Wang, J.; Kang, L.; and Liu, Y. 2020. Optimal scheduling for electric bus fleets based on dynamic programming approach by considering battery capacity fade. *Renewable and Sustainable Energy Reviews* 130: 109978.

Wang, Y.; Zhang, D.; Hu, L.; Yang, Y.; and Lee, L. H. 2017. A data-driven and optimal bus scheduling model with time-dependent traffic and demand. *IEEE Transactions on Intelligent Transportation Systems* 18(9): 2443–2452.

Yang, X.; and Liu, L. 2020. A Multi-Objective Bus Rapid Transit Energy Saving Dispatching Optimization Considering Multiple Types of Vehicles. *IEEE Access* 8: 79459–79471.

Zhou, G.-J.; Xie, D.-F.; Zhao, X.-M.; and Lu, C. 2020. Collaborative Optimization of Vehicle and Charging Scheduling for a Bus Fleet Mixed With Electric and Traditional Buses. *IEEE Access* 8: 8056–8072.