A Modular Approach to Optimal Hybrid Bus Allocation for TCAT

Anmol Kabra (ak2426), Kenneth Fang (kwf37), Nicholas Kaye (nak64)

November 2017

Abstract

In this paper, we propose a series of models and algorithms to optimally allocate 8 new diesel-hybrid buses across several TCAT bus routes. With the addition of these hybrid buses, we aim to maximize the fuel efficiency of the TCAT bus system. We independently tackle the fuel consumption and bus allocation problems, and integrate our models and algorithms together to propose a modular approach to find the best distribution of the 8 hybrid buses across the TCAT bus system. Our design includes a model to minimize the total amount of fuel consumed by the TCAT bus system every day in order to cut TCAT's diesel expenditure and reduce its carbon footprint. This is accomplished by considering the mechanics of the buses' motion, while taking the road gradient into consideration. Next, we find the best allocation for buses on a route by using a novel algorithm that uses the least number of buses on a route, thus allowing those frequently-used buses to be replaced by hybrid buses to save fuel. Our experiments show that by allocating 6 hybrid buses to Route 82 and the remaining 2 hybrid buses to Route 11, the total fuel consumption is minimized and saves over 20 gallons of diesel fuel every day when compared to fuel consumption without the hybrid buses. Annually, this can result in financial savings of $\approx$ \$15,000 for TCAT which could then be used for a variety of other purposes.

Contents

1 Introduction **4**
- 1.1 Constraints and Goals for our Models 4
- 1.2 Properties of the Diesel-Electric Hybrid Buses 4

2 Models of Fuel Consumption **4**
- 2.1 Fuel Consumption - Model 1 5
  - 2.1.1 Route Assumptions 5
  - 2.1.2 Fuel Consumption Assumptions 5
  - 2.1.3 Notation 6
  - 2.1.4 Parameter Values and Justification 6
  - 2.1.5 Derivation of Fuel Consumption between Successive Stops 6
- 2.2 Fuel Consumption - Model 2 9
  - 2.2.1 Assumptions 9
  - 2.2.2 Notation 9
  - 2.2.3 Fuel Consumption between Successive Stops 9
  - 2.2.4 Issues with Model 2 10

3 Representation of Routes **10**
- 3.1 Complications in TCAT Routes 10
- 3.2 Simplifying TCAT Routes: Choosing Routes' Path 11
  - 3.2.1 Route 10 11
  - 3.2.2 Route 11 11
  - 3.2.3 Route 17 11
- 3.3 Schedules 12

4 Algorithms for Bus Allocation **12**
- 4.1 Assumptions 12
- 4.2 Bus Allocation - Algorithm 1 14
  - 4.2.1 Assumptions 14
  - 4.2.2 Motivation behind the Algorithm 15
- 4.3 Extending Algorithm 1 15
  - 4.3.1 Merits of this Extension over Algorithm 1 16

5 Integrating Fuel Consumption Models and Bus Allocation Algorithms **16**

6 Model Predictions **17**
- 6.1 Results from Algorithm 1 17
- 6.2 Results with the Extension to Algorithm 1 17

7 Validation and Robustness **18**

8 Strengths and Weaknesses **21**

9 Future Work **22**
- 9.1 Fuel Consumption Model Improvements 22
- 9.2 Bus Allocation Algorithm Improvements 22
- 9.3 Alternative Route Recommendations 22

10 Conclusion **23**

A Stops Selected for the Routes **23**
A.1 `.csv` File Format of Schedules for Bus Allocation Algorithms 26
A.2 Implementation of Fuel Consumption - Model 1 26
A.3 Implementation of Algorithm 1 in Java 27
A.4 Integrated Model with Extension to Algorithm 1 in MATLAB 30

1 Introduction

The Tompkins Consolidated Area Transit (TCAT) is considering to allocate hybrid buses on some of the routes to maximize fuel efficiency and reduce fuel consumption. With 33 bus routes and an estimated 4.2 million annual trips, the TCAT buses serve the citizens of the Tompkins County, NY.

Given the increasing demand for public transit in the county and increasing costs of diesel fuel, the TCAT administration is looking for optimal allocation of its 8 non-plug-in diesel-electric hybrid buses on 6 routes - Route 10, 11, 15, 17, 81, and 82.

1.1 Constraints and Goals for our Models

TCAT has conveyed some constraints on the assignment of their hybrid buses on different routes. These constraints also specify the goals of our models.

1. Use of hybrid buses will be restricted to routes 10, 11, 15, 17, 81, and 82. Our models should convey a priority-based allocation. Moreover, TCAT would like to use the 8 buses they already own, and so this is a constraint to our models.

2. TCAT will consider proposals to modify existing routes provided the advantages resulting from the new routes are significant. However, this requires a careful examination of commuters' footfalls in the TCAT transit, and relevant data is scarcely available. Our models work with existing routes and identify a good, if not best, allocation of buses on these routes.

1.2 Properties of the Diesel-Electric Hybrid Buses

According to the prompt, these are some properties of the hybrid buses that TCAT currently owns.

- The bus can be powered either by diesel or electricity at any moment.
- On very steep hills and on mildly wintry roads, the bus can only operate on diesel.
- There is Maximum Storage Capacity and Minimum Charge Level for the batteries. The bus cannot store more power/charge than the specified value, and it cannot operate on electricity below the Minimum Charge Level.
- When the bus is running in diesel mode, the electric batteries get slowly recharged.
- Hybrid buses have regenerative brakes to provide another source for recharging the batteries when brakes are applied. There is a maximum rate of recharge as well.

2 Models of Fuel Consumption

In order to find the most efficient allocation of buses, we modeled TCAT Routes 10, 11, 15, 17, 81, 82 as weighted graphs. In these models, every stop on the route map (marked by a black square on the TCAT route maps) corresponds to one vertex of our graph and the weights of the edges is the estimated fuel consumption between two stops.

These weights are a function of the elevations of the starting stop and destination stop, as well as the distance of the path between the two stops. Several other dependencies are included as the complexity of the models increases. We propose two models for fuel consumption:

Model 1 The fuel consumption expression is not a function of recharge rate. Also, we assume that a bus either operates in diesel-only or electric-only mode between two successive stops, but not both.

Model 2 To account for the assumptions in Model 1, Model 2 considers that a bus operates in both diesel-only and electric-only modes between two successive stops, and accounts for *some* properties of diesel-electric hybrid buses.

The models are discussed in length below.

2.1 Fuel Consumption - Model 1

2.1.1 Route Assumptions

- Route data was obtained from Google Maps. We assume that Google Maps and its APIs provide correct elevation of bus stops from sea level and the road distance between two stops. Elevation details were obtained from the website FreeMapTools, which uses the Google Maps API [1].
- We refer to the road distance when we talk about the distance between two bus stops (sometimes referred to as d). This data is approximated from Google Maps.
- We assume the roads have constant slope equal to the change in elevation divided by the path length: slope $= \frac{\Delta H}{d}$, where ΔH is the change in elevation and d is the road distance between two bus stops (or the length of our edge in the graph representation). Therefore, the slope gradient on an edge of our graph representation is constant throughout the edge.
- Buses will stop at every official stop as shown on the TCAT route website [2], regardless of whether commuters request drop-off or boarding.
- Buses will not accommodate extraneous requests and will stick strictly to the route on the TCAT route website.
- We ignore delays in the buses' schedules due to traffic and assume that detours are non-existent throughout the study period.
- Only the routes operating in the Cornell's Fall and Spring sessions are studied.
- We assume that wintery roads (covered with snow or ice) and roads during rains have equal change in friction forces.
- We assume that there are no stops between two successive bus stops. This also means that there are no traffic lights between two successive bus stops.
- Between two successive segments, the bus operates in either diesel-only or diesel-electic mode, but not in both.

2.1.2 Fuel Consumption Assumptions

- We assume that the fleet of TCAT buses is comprised of only the most recent bus purchases of diesel and hybrid transit buses: the Gillig Advantage T40 and Gillig Advantage T40 Hybrid [3].
- Fuel consumption is calculated based off an average fuel efficiency (e, units of km/litre) that is affected only by the slope of the road. Therefore, e is not affected by the quality of the fuel, traffic conditions, bus quality/conditions or other such factors.
- Fuel consumption varies only with the slope of the road ($\frac{\Delta H}{d}$) and with the distance covered (d).
- No slip condition for fuel consumption calculation: this means that the wheels are constantly under static friction. This is obviously not always true, especially in winter. To simplify our model, we do not consider any other friction forces operating between the buses tyres and asphalt.

- Similarly, air resistance and other resistive forces to the bus' motion are ignored.
- Bus moves at same speed uphill and downhill as on flat ground. The speed is averaged over the whole distance, so it takes into account the change in speed when buses stop at the bus stops.
- The fuel efficiency value of the hybrid bus [4] accounts for the effects of regenerative braking and the rate of recharge when using diesel power.
- We equate maximizing fuel efficiency with minimizing total fuel consumption when finding our optimal bus allocation. This may not be what TCAT demands, but assuming this correlation simplifies our model.

2.1.3 Notation

- d m - road distance between two bus stops.
- $\frac{\Delta H}{d}$ - slope gradient of the road between two bus stops.
- f l - fuel consumption between two bus stops
- e km l^{-1} - fuel efficiency of the bus model.

2.1.4 Parameter Values and Justification

Parameter	Notation	Value	Justification
Fuel Efficiency of diesel-only buses	e_d	4.2 MPG	Derived in Academic Paper [4]
Fuel Efficiency of diesel-electric hybrid buses	e_h	4.8 MPG	Derived in Academic Paper [4]
Coefficient of static friction on dry roads	μ_d	0.7	Derived in Academic Paper [5]
Coefficient of static friction on wet roads	μ_w	0.4	Derived in Academic Paper [5]
Maximum Grade for Electric Bus	T	5%	Given by APTA [6]

Table 1: Parameters Used in Model 1: These are taken as correct constants when running simulations.

2.1.5 Derivation of Fuel Consumption between Successive Stops

With constant fuel efficiency, the fuel consumed to travel a distance d with is simply:

$$f = \frac{d}{e} \tag{1}$$

However, we want to account for variations in slope, as a purely downhill route would be much more fuel efficient than an uphill route. This means we need a method for relating the change in slope to the change in fuel consumed.

Internal combustion engines have a parameter called the Brake Specific Fuel Consumption, or BSFC, for short. the BSFC is defined as follows [8]:

$$BSFC = \frac{r_{fc}}{P}$$

where r_{fc} is the rate of fuel consumption ($\frac{\Delta G}{\Delta t}$), and P is the power. Power for driving wheel is given by the following equation:

$$P = \omega\tau$$

where ω is the angular velocity of the wheel[1], and τ is the torque exerted on the wheel(s). ω only depends on the speed of the bus, and τ depends on the amount of force F needed to propel the bus along the road, multiplied by the radius of the wheels, r. To summarize, we have the following equations:

$$BSFC = \frac{r_{fc}}{P} \tag{2}$$
$$r_{fc} = \frac{\Delta G}{\Delta t} \tag{3}$$
$$P = \omega \tau \tag{4}$$
$$\tau = rF \tag{5}$$

Combining these gives us the following equation for fuel consumption Δf:

$$\Delta f = (BSFC)\omega r F \Delta t$$

This tells us that fuel consumption is influenced by speed (ω), wheel size (r), running time Δt, and force against the ground F. This makes sense physically.

Now let's consider what changes as the slope of the road increases. We assume the bus proceeds at the same speed as before, so the only thing that changes has to be the force exerted on the ground; in order to move at the same speed uphill as on flat ground, the bus must exert more force, which translates to an increased fuel usage. Under this assumption, we can take the ratio of fuel consumed on flat ground versus fuel consumed on a slope and derive a formula for fuel consumed on a slope:

$$\frac{\Delta f_{slope}}{\Delta f_{flat}} = \frac{(BSFC)\omega r F_{slope} \Delta t}{(BSFC)\omega r F_{flat} \Delta t} \tag{6}$$
$$\frac{\Delta f_{slope}}{\Delta f_{flat}} = \frac{F_{slope}}{F_{flat}} \tag{7}$$
$$\Delta f_{slope} = \frac{F_{slope}}{F_{flat}} \Delta f_{flat} \tag{8}$$

Equation (8) changes to Equation (9) when we start from $t = 0$ as $f = 0$ at $t = 0$.

$$f_{slope} = \frac{F_{slope}}{F_{flat}} f_{flat} \tag{9}$$

This suggests that to get the fuel consumption on a slope, we can multiply the fuel consumption on flat ground by the ratio of the force exerted by the wheel along sloped ground versus flat ground. To find this, we begin by drawing a Free Body diagram of one bus wheel on flat ground versus one bus wheel on a slope with angle θ to flat ground:

In Figure 1, F_f is the force due to static friction due to the no slip assumption we made in Section 2.1.2.

As stated before, we want the total force acting parallel to the ground. This is trivial in the flat ground case:

$$\sum F_{||\text{ to road}} = F_f = \mu F_g \tag{10}$$

For the case with slope of the road equal to $\tan\theta$, we want the sum of components parallel to the road, so we should shift our coordinates to be parallel to F_f and F_n. In this basis, we see that

[1] ω is same for all wheels as the sizes of the wheels are equal and the speed of the bus is constant

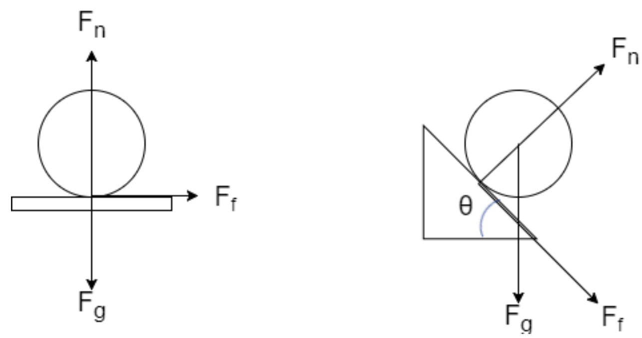


Figure 1: Free Body Diagrams of Bus Wheels. (left) Forces acting on a wheel on flat ground; (right) Forces acting on a wheel on a ground with slope = $\tan\theta$. F_g is a quarter of the weight of the bus (as the weight is distributed over 4 wheels), F_f is the frictional force acting on the wheels, and F_n is the normal force from the surface.

$$\sum F_{||\text{ to road}} = F_f + F_g \sin\theta = \mu F_n + F_g \sin\theta = \mu F_g \cos\theta + F_g \sin\theta = \mu F_g(\cos\theta + \sin\theta) \tag{11}$$

Combining Equations (1) and (9) to (11), we get the following:

$$f_{slope} = \frac{F_g(\mu\cos\theta + \sin\theta)}{\mu F_g} f_{flat} \tag{12}$$

$$f_{slope} = \frac{\mu\cos\theta + \sin\theta}{\mu} f_{flat} \tag{13}$$

$$f_{slope} = \left(\cos\theta + \frac{\sin\theta}{\mu}\right)\frac{d}{e} \tag{14}$$

Of course, this leaves us with the question of how to find θ. Since we assume constant slope over a certain distance, we can find θ using the following equation:

$$\sin\theta = \frac{\Delta H}{d} \tag{15}$$

where ΔH is the change in elevation from origin to destination. Equations (14) and (15) give us the complete equations for our model for finding the fuel consumption at any slope. And, as expected, at $\theta = 0$, our equation becomes Equation (1), on the ideal flat-ground case.

Using Equation (14) and e_d, we can use the information about the path to calculate the fuel consumption between two successive stops for the diesel buses.

$$f_{diesel} = \left(\cos\theta + \frac{\sin\theta}{\mu}\right)\frac{d}{e_d} \tag{16}$$

However, this same method, using e_h instead of e_d, cannot be used to calculate the fuel consumption between two successive stops for hybrid buses. This is because the hybrid buses can only use their electric motors to drive up inclines for grades below a given threshold. Thus we must adjust Equation (16) to account for when the hybrid bus must rely on diesel power for certain inclines. This results in a piece-wise function depending on θ:

$$f_{hybrid} = \begin{cases} \left(\cos\theta + \frac{\sin\theta}{\mu}\right)\frac{d}{e_h} & \text{if } \theta \leq T \\ \left(\cos\theta + \frac{\sin\theta}{\mu}\right)\frac{d}{e_d} & \text{if } \theta > T \end{cases} \tag{17}$$

Thus using Equation (16) and Equation (17) we are able to calculate fuel consumption between any two successive stops for both diesel and hybrid buses.

2.2 Fuel Consumption - Model 2

We see that Model 1 does not take the properties of the electric cells in the hybrid buses into consideration. Model 1 uses constant fuel efficiency parameters to calculate the fuel consumption for diesel-only and diesel-electric hybrid buses, and the properties of hybrid buses - Maximum Storage Capacity, Minimum Operating Charge, recharge on diesel mode, and recharge due to regenerative braking - are ignored.

With Model 2, we attempt to attend to the properties of diesel-electric hybrid vehicles by including parameters, constants, and markers in Equation (14). We come up with a model to represent the fuel consumption by a diesel-electric hybrid bus, as a function of different properties of the hybrid bus.

2.2.1 Assumptions

Most assumptions from Model 1 (Sections 2.1.1 and 2.1.2) also apply for Model 2. These assumptions differ from those of Model 1.

- We assume that fuel is only consumed when the bus is operating in diesel mode, and the bus can operate in electric mode for some distance in the route. Therefore, the bus would operate in diesel-only mode for the remaining distance.
- Buses have a constant energy conversation factor. 1 unit of charge in the battery always provides for the same distance traveled, regardless of the slope gradient and any other factors.
- Between two consecutive stops, the bus can operate in *both* diesel-only and electric-only mode, unlike the assumption in Model 1.

Note: We still assume that no traffic lights are present between two successive stops and that the speed of the bus is averaged between two successive stops.

2.2.2 Notation

- d_d m - road distance covered while operating in diesel-only mode.
- d_e m - road distance covered while operating in electric model.
- Q_{max} C - Maximum Storage Capacity.
- Q_{min} C - Minimum Operating Charge.
- D_q $\mathrm{m\,C^{-1}}$ - road distance covered by the bus while in electric mode. Specifically, D_q is the distance covered by the bus per unit charge of the batter in the bus.
- ρ $\mathrm{C\,m^{-1}}$ - increase in the charge level of the battery per unit distance covered in diesel-only mode.

2.2.3 Fuel Consumption between Successive Stops

Let's say that the battery is charged ρ C per 1 meter distance covered while operating in diesel-only mode. Also, let D_q m be the distance covered by the bus per 1 Coulomb charge consumed from the battery. Additionally, we know that $d = d_e + d_d$, where d is the road distance between two successive stops on the bus' route.

It is trivial to see that a bus can cover $D_q(Q_{max} - Q_{min})$ m solely on the charge present in the battery. Additionally, the bus can also cover $D_q(\rho d_d)$ m distance from the slow recharge generated while running in diesel-only mode. Thus, the total distance covered in electric-only mode is[2]:

$$d_e = D_q\bigg((Q_{max} - Q_{min}) + \rho d_d\bigg) \tag{18}$$

Since $d = d_e + d_d$,

$$d_d = d - D_q(Q_{max} - Q_{min}) - D_q\rho d_d \tag{19}$$

$$\bigg(1 + D_q\rho\bigg)d_d = d - D_q(Q_{max} - Q_{min}) \tag{20}$$

$$d_d = \frac{d - D_q(Q_{max} - Q_{min})}{1 + D_q\rho} \tag{21}$$

Equation (14) thus changes to:

$$f_{slope} = \left(\cos\theta + \frac{\sin\theta}{\mu}\right)\frac{d_d}{e} \tag{22}$$

Extension of Equation (22) corresponding to the addition of threshold follows similar to Equation (17).

2.2.4 Issues with Model 2

As discussed, Model 2 conveys a lot more than Model 1. However, using Model 2 to estimate the fuel consumption of a diesel-electric hybrid bus can be unstable and undeterminstic. This is particularly because several parameters in Equation (21) are unknown for the TCAT buses. **Moreover, parameters such as D_q, ρ, Q_{max}, and Q_{min} vary from bus model to bus model, and we couldn't find a trustworthy source to estimate these values for the TCAT buses.**

This is the main reason why **we estimate fuel consumption using Model 1**, even though Model 2 can potentially provide us more accurate measures of the fuel consumption by diesel-electric hybrid buses. More discussion in Section 9.1.

3 Representation of Routes

We see that our fuel consumption models require information such as road distance between two successive stops (represented by an edge between two vertices) and difference in elevation of the two connected vertices on the graph, which represent two successive bus stops.

In addition to the data and parameter values discussed in Sections 2.1.1 and 2.1.2, we collect the elevation of bus stops and the road distance between two successive stops from the Google Maps and its APIs [1].

3.1 Complications in TCAT Routes

While approaching this requirement, we recognized that we will require route data from TCAT on routes 10, 11, 15, 17, 81, and 82. However, we soon found that these routes have different schedules based on different time of the day and different days of the week.

For instance, Route 11 (Figure 2) has different routes for Mon-Fri days and Mon-Sat late nights. Moreover, snow detours affect the route during different times of the year. Whereas the latter complication is addressed in our models' assumptions (Section 2.1.2), we did not assume any route simplifications previously. Consequently, we discuss some simplifications/assumptions to the route data below.

[2]As noted in Model 2 assumptions, we ignore the recharge generated due to regenerative braking

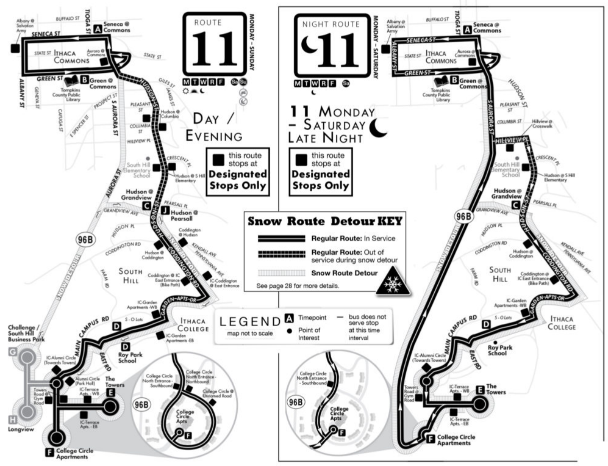


Figure 2: Route 11 Path from TCAT's website [2].

3.2 Simplifying TCAT Routes: Choosing Routes' Path

Since TCAT routes are complicated - the same route taking varying paths depending on the time of the day and days of the week, we simplify all routes for our models to handle. These can be considered assumptions to our models, even though they were not discussed in Section 2.1.1. In the next few subsections, we propose the routes that our models will work with and justify our assumptions' soundness.

We discuss Routes 10, 11, and 17 as they describe all types of the routes among others - loop routes, direct routes (17 is the only one), and *quirky* loop routes. We do not show the graph representations of other routes due to time constraints.

3.2.1 Route 10

Route 10 is a Mon-Fri shuttle plying during work hours originally. There are no temporary or seasonal detours and the buses cover all stops marked on the map.

We convert the original map in Figure 3b into our proposed graph representation (Figure 3a) for our models to conveniently interpret them.

3.2.2 Route 11

Route 11 has numerous routes and schedules including Mon-Fri day schedule, Mon-Fri night schedule, and Sunday schedules, originally. There are a few temporary or seasonal detours and all buses do not cover all stops even in the same day.

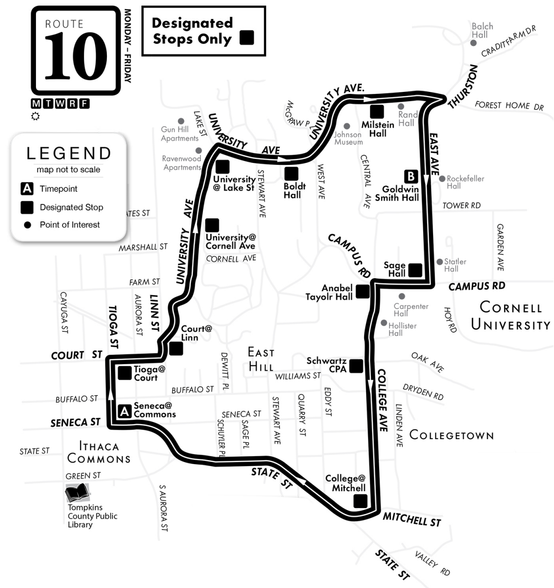
We consider the Mon-Fri day route and simplify it like this:

1. The buses start from Seneca@Commons, loop the Commons and go to IC through Hudson Road.
2. After IC Alumni Circle, the buses go to IC Terrace Apts → The Towers → IC Terrace Apts → College Circle → College Circle Apts → College Circle@Unnamed Rd. → College Circle → IC Terrace Apts → The Towers → IC Terrace Apts → Back to Commons through stops.

3.2.3 Route 17

Buses on Route 17 leave TCAT Garage inbound to Commons and return to TCAT Garage at the end of the day. Therefore, all buses - whether diesel-only or diesel-electric hybrid - have to take this route. Consequently, there is no scope of optimization or fuel efficiency reduction on Route 17 (Figure 5).

(a) Graph Representation of Route 10



(b) Route 10 Path from TCAT's website [2]

Figure 3: Graph Representation and TCAT Map of Route 10: Buses on this trip follow a simple loop on Mon-Fri days at specified time intervals. Bus frequency increases at peak hours (between 1000 and 1300 hours) and slowly decreases as the day wanes. Hueristically, it would make sense to allocate hybrid buses to the bus that loops most often on Route 10.

We may safely ignore Route 17 as there is no scope of optimization here.

3.3 Schedules

For our algorithms in Section 4, we need schedules of trips on a route, obtained from the TCAT's website - similar to that in Figure 6a.

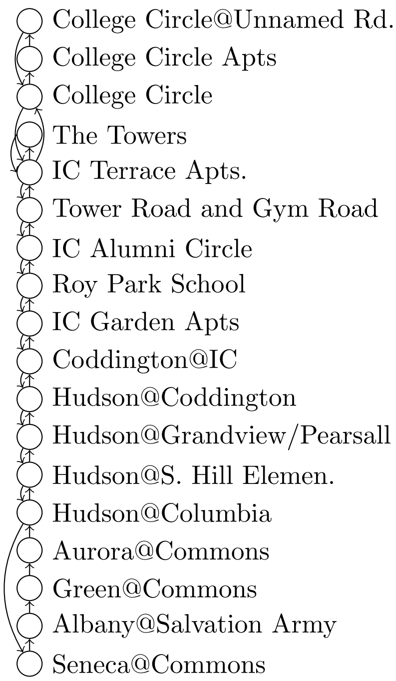
As discussed previously, we only consider the Mon-Fri trips in the day time to simplify our models. With this in mind, we compiled `.csv` files of schedules with the trips' start times, end times, origin stop, and destination stops for the appropriate routes and used our algorithms in Section 4 to calculate the number of trips per bus, the number of buses needed, and the fuel usage for the day on specific routes, categorized by buses.

4 Algorithms for Bus Allocation

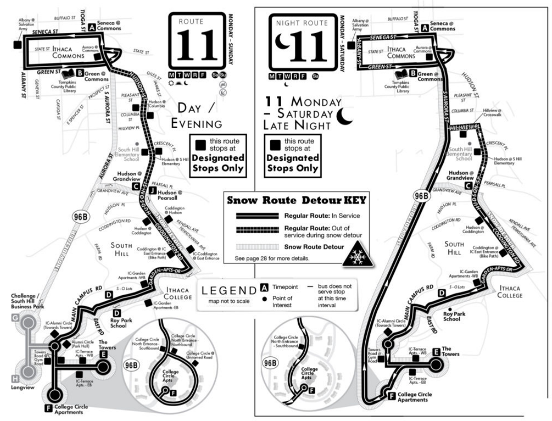
With the models for fuel consumption, we can estimate the fuel consumed between two successive stops, or in one edge of the graph representation. They can be used to compute the fuel consumption from one start stop to an end stop along that route. However, in order to get overall fuel consumption and analyze how much fuel is saved by allocating hybrid buses, we must model how many buses are used in one route per day, and how much fuel each of those buses consumes per day.

4.1 Assumptions

- Upon reaching their final stop on a trip, buses are able to begin a new trip at any stop, instantaneously. This means that even if a bus finishes at 7:20 AM, and a new trip starts across campus at 7:21AM, we assume the same bus can cover both trips.



(a) Graph Representation of Route 11



(b) Route 11 Path from TCAT's website [2]

Figure 4: Graph Representation and TCAT Map of Route 11: A bus' trip on Route 11 is convoluted - it visits multiple stops near Ithaca College (IC) more than once on a trip. This is why we call this a 'quirky' loop route. Not many routes in consideration have this type of convolution.

Figure 5: Route 17 Path from TCAT's website [2]: Route 17 is a direct route without loops. Moreover, all buses must cover this route regardless of their types. Therefore, there is no scope of increased fuel efficiency if TCAT maintains all of its buses and doesn't replace diesel-only buses with hybrid buses.

Route 10 Weekday • Cornell / Commons Shuttle

[A] Seneca @ Commons	[B] Goldwin Smith Hall	[A] Seneca @ Commons
7:12 A	7:20 A	7:31 A
7:24 A	7:32 A	7:43 A
7:36 A	7:44 A	7:55 A
7:48 A	7:56 A	8:07 A
8:00 A	8:09 A	8:19 A
8:06 A	8:15 A	8:25 A
8:12 A	8:21 A	8:31 A
8:18 A	8:27 A	8:37 A
8:24 A	8:33 A	8:43 A

(a) Schedule of Route 10 from TCAT's website [2]: This schedule has unchanging start and end stops.

Route 15 Weekday • Southside Shopper • Wegmans • Tops • Walmart • Titus Towers 1 • Commons

[A] Seneca @ Commons	[B] Titus Towers	Wegmans	[C] Tops	[D] Walmart	[B] Titus Towers	[E] Green @ Commons	Continues as Route #
7:21 A	7:28 A	Yes	7:37 A	7:42 A	7:47 A	7:55 A	32
8:21 A	8:28 A	Yes	8:37 A	8:42 A	8:47 A	8:55 A	32
9:21 A	9:28 A	Yes	9:37 A	9:42 A	9:47 A	9:55 A	32
10:21 A	10:28 A	Yes	10:37 A	10:42 A	10:47 A	10:55 A	32
11:21 A	11:28 A	Yes	11:37 A	11:42 A	11:47 A	11:55 A	32
12:21 P	12:28 P	Yes	12:37 P	12:42 P	12:47 P	12:55 P	32
1:21 P	1:28 P	Yes	1:37 P	1:42 P	1:47 P	1:55 P	32

(b) Schedule of Route 15 from TCAT's website [2]: This schedule often has different start and end stops.

Figure 6: Schedules of Different Routes: The start times, end times, origin stop, and destination stop are copied from the website and fed into our programs as `.csv` files.

- Drivers do not require any breaks. This can also be seen as drivers take negligible time to switch out with one another.
- All buses hit their stops exactly on time. We do not account for any possible mishaps along the trip that add delays.

4.2 Bus Allocation - Algorithm 1

We approach the problem of calculating the number of buses needed for a route in a day's operation by maintaining 'repositories' of buses. More specifically, we maintain a repository of buses that are available (not operating on the route) and a repository of buses plying on the route. The former repository is modeled with a Last-In-First-Out `stack` and the latter is modeled with a First-In-First-Out `queue`.

4.2.1 Assumptions

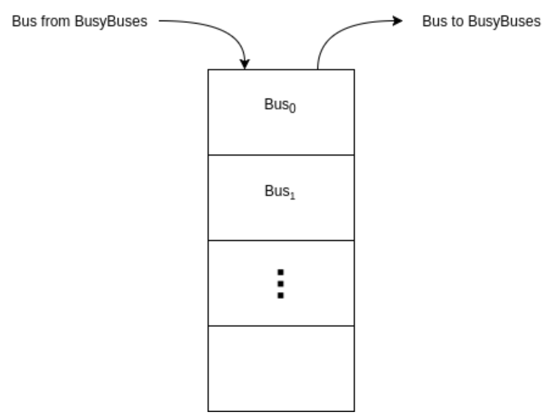
1. This algorithm assumes that the start times represent the times at an unchanging origin, the end times represent the times at an unchanging destination.

4.2.2 Motivation behind the Algorithm

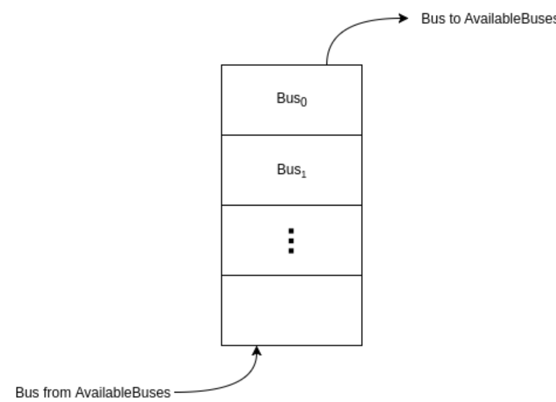
From the TCAT's website, we can obtain the start and end times of a trip at the starting and ending locations. Following Assumption 1 of this algorithm, this algorithm would simply overestimate the number of trips if the data procured from TCAT's website has different starting and ending times for a route. Algorithm 2 in Section 4.3 tackles this assumption. However, it uses Algorithm 1's features to build its algorithm.

Stack Let's consider the `stack` repository of available buses first. When a 'busy' bus completes the route, it becomes 'available' for use again on the same route (shown in Figure 7a). If we had inserted this bus into the end of our available-buses repository, then we wouldn't have used the minimum number of buses possible. This is a valid constraint as TCAT owns a limited fleet and it would want to use as few buses as possible on a route.

Fewer Buses-More Trips/Bus or More Buses-Fewer Trips/Bus This is a design decision on TCAT's side - it could either have more buses doing fewer trips per bus, or fewer buses doing more trips per bus. **As you will see in the following sections, we propose the latter - one could replace the diesel-only bus doing a lot of trips with a diesel-electric hybrid bus to drastically decrease fuel consumption.**



(a) Stack: This represents the 'available-buses' repository. Buses from the 'busy-buses' repository are "pushed" to this stack, and buses are "pulled" out.



(b) Queue: The queue represents the 'busy-buses' repository, ordered by end time (time at which they reach their destination). Buses starting trips are added, and buses ending trips are removed.

Figure 7: Pictorial Representation of the Concepts in this Algorithm: When its time for a new bus to start at the origin, Algorithm 1 checks if any 'busy-bus' has reached its destination, pushes them onto the 'available-buses' stack. Buses that start operating on the route are added to the queue.

Queue The busy-buses repository is nicely modeled by a `queue` in which a bus is added to the end of the queue with the end time recorded (shown in Figure 7b). A bus is removed from the head of the queue when it is available after completing the trip (end time of that bus is lower than the current start time in the calendar). This bus is pushed back to the available-buses `stack` as it is available for another trip now.

4.3 Extending Algorithm 1

This is an extension of Algorithm 1, and does not assume unchanging start and end bus stops. From the TCAT schedules on their website, we can record the starting stop and ending stops of each trip, in addition to the start and end times of that same trip - as discussed in Section 3.3.

Algorithm 1 Calculating # Trips on a Route

```
function CalcTrips(N_B, C)                ▷ N_B: Total # of Buses, C is the Calendar of the Route
    b_available ← Stack
    b_busy ← Queue
    for i = 0, 1, ..., N_B − 1 do
        b_available.push(new Bus(i))
    for s, e in C do
        if b_busy.isempty() then
            b ← b_available.pop()
            b.setEndTime(e)
            b_busy.insert(b)
        else
            while b_busy.peek().getEndTime() < s do
                b_available.push(b_busy.remove())
            if b_available.isempty() then
                "Too Few Buses!"
                break
            else
                b ← b_available.pop()
                b.setEndTime(e)
                b_busy.insert(b)
```

This information lets us calculate the fuel consumption by each bus with more precision. Then, instead of calculating the number of trips each bus makes, we can instead dynamically compute the amount of fuel consumption as each bus takes a trip (gets added to the queue). So, our final algorithm takes in the TCAT schedule, elevation of stops, and distance between stops, and computes the number of buses as well as the fuel consumed by each bus, under the stated assumptions.

4.3.1 Merits of this Extension over Algorithm 1

- This extension enables our algorithm to estimate the fuel consumption per bus per route to discern the total savings for the day.
- Algorithm 1 is a naive way of calculating the number of trips by a bus, and overestimates the number of trips as all trips in the schedule are not the same length (different start and end times). This is tackled by this extension.
- We use the graph representation of routes explained in Section 3.2 to sum up segments of fuel usage and produce a more accurate result than obtained with a naive application of Algorithm 1 on the schedules.

The results of this extension are discussed in Section 6.

5 Integrating Fuel Consumption Models and Bus Allocation Algorithms

Now that we have a model for estimating fuel consumption between two successive stops (or an edge in the graph representation), and an algorithm to find the most efficient bus allocation (provided our assumptions are correct), we can integrate both features.

We **select Model 1** (Section 2.1) of our Fuel Consumption as we do not currently have confident parameter values of Equation (22) for Model 2 (Section 2.2). We could have tried estimated values for those parameters, but the model could have been difficult to bring to correct values given that we couldn't find good sources for the parameters' values.

The **extension to Algorithm 1 potentially provides better approximations** to total fuel consumption. Using the schedules of different routes, we find the number of buses and trips that a bus needs to make, and thus dynamically calculate the fuel consumption per trip, and per day eventually.

6 Model Predictions

6.1 Results from Algorithm 1

This algorithm only calculates the number of trips taken by different buses on a route, and overestimates the number of trips if the calendar contains start and end times at changing origins and destinations. We say that it overestimates the number of trips as we don't think of shorter trips as equal to longer, full trips. Nevertheless, the model provides a good estimate of the *maximum* number of trips required by each bus.

Bus Number	Route 10	Route 11	Route 15	Route 81	Route 82
1	35	14	14	19	12
2	25	14	2	8	12
3	15			2	11
4	7				11
5					11
6					8
7					4

Table 2: Results from Algorithm 1: Considering all start times as times at origin, and end times as times at destination stops, the number of trips per bus is shown in this table.

6.2 Results with the Extension to Algorithm 1

After integrating both the model and the algorithm, we ran the programs on the derived schedules of different routes.

Bus Number	Route 10	Route 11	Route 15	Route 81	Route 82
1	25.04	22.91	15.63	12.16	22.93
2	20.49	22.91	2.23	3.61	22.43
3	9.86			0.90	23.93
4	3.79				21.93
5					21.93
6					21.93
7					3.99

Table 3: Diesel Bus Allocation by Route and Fuel Consumption: We see that some buses, which run more than other buses on the same route, consume more fuel than those others.

Table 3 lists each route and the daily fuel consumption per bus serving that route, in gallons. The bus number corresponds to how early it starts, with bus number 1 starting at the very first trip of the day, and the following buses filling in for any trips where all the earlier buses are occupied. Under our assumptions, we try to reuse the first buses for as many trips as possible, as we only have several hybrids. This strategy is beneficial as if one bus does more work than others serving that route, more fuel can be saved by swapping it out with a hybrid bus.

Table 4 displays daily fuel consumption of hybrid buses serving the same routes:

Bus Number	Route 10	Route 11	Route 15	Route 81	Route 82
1	23.12	20.56	13.71	10.77	20.15
2	18.92	20.56	1.96	3.19	19.71
3	9.11			0.80	21.03
4	3.50				19.28
5					19.28
6					19.28
7					3.51

Table 4: Hybrid Bus Allocation by Route and Fuel Consumption: A similar trend observed when the threshold gradient $T = 5\%$ or $2.86°$ (see Table 1 for T).

We can see that overall fuel consumption drops by a 1-3 gallons every day for several buses. Table 5 shows the difference between the fuel consumption of a diesel bus and the fuel consumption of a hybrid bus. From these results, we can determine where to allocate our hybrid buses in order to save the most fuel.

Bus Number	Route 10	Route 11	Route 15	Route 81	Route 82
1	1.92	2.35	1.92	1.39	2.78
2	1.57	2.35	0.27	0.42	2.72
3	0.75			0.10	2.90
4	0.29				2.65
5					2.65
6					2.65
7					0.48

Table 5: Fuel Saved per Bus Categorized by Route

From Table 5, we can infer that we should allocate 6 hybrid buses to Route 82 and 2 hybrid buses to Route 11. Provided TCAT sends out buses similarly to how we simulated them, this allocation will save 21.05 gallons of fuel per day. With the current average of \$2.83 per gallon of diesel fuel, this amounts to \$59.57 saved per weekday from the Monday-Friday schedules alone. This gives a weekly savings of ≈\$300 per week, which translates to ≈ \$15000 annual savings, disregarding holidays or reduced bus schedules.

7 Validation and Robustness

We performed sensitivity analysis on several of the key parameters that may influence the fuel efficiency and fuel consumption of the TCAT buses on each of the different routes. For modeling the diesel buses, consumption may depend on the coefficient of friction between the tyres and the road, μ, and the baseline fuel efficiency levels, e_d and e_h.

We first varied the value of μ. In Figure 8, each column represents a different value of μ from 0.1 to 0.9 for each bus route. Since our model uses μ to scale the effect of the road's gradient, we would expect the effects of a change in μ to cancel out for a closed loop. As we see in Figure 8, for all closed loop routes, the change in μ has no effect and for Route 17, the effect of the path's gradient is exaggerated. Thus the results of our model align with our initial predictions. Thus slight error in measuring the coefficient of friction between the road and the bus tyres should not have a significant affect on our optimal solution.

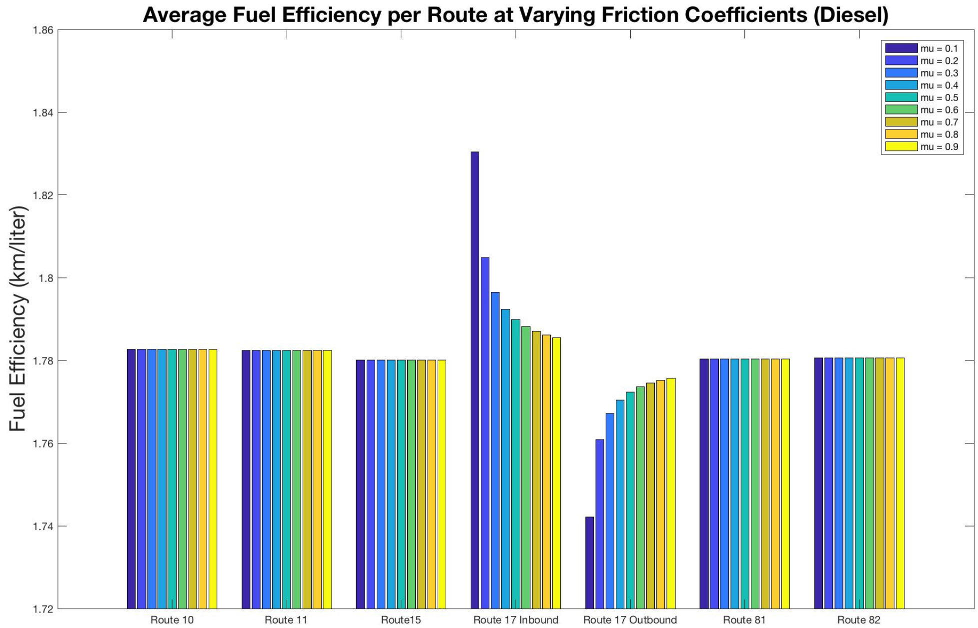


Figure 8: Average Fuel Efficiency per Route at Varying Friction Coefficients (Diesel)

We then varied the baseline fuel efficiency that we had previously assumed [4]. As expected, the increase in average fuel efficiency resulted in equal shifts of the average fuel efficiencies for each route, but did not distort the differences between the routes. Thus our results should not be significantly affected by small errors in the measure of the buses' baseline average fuel efficiency and our recommended strategy would still hold.

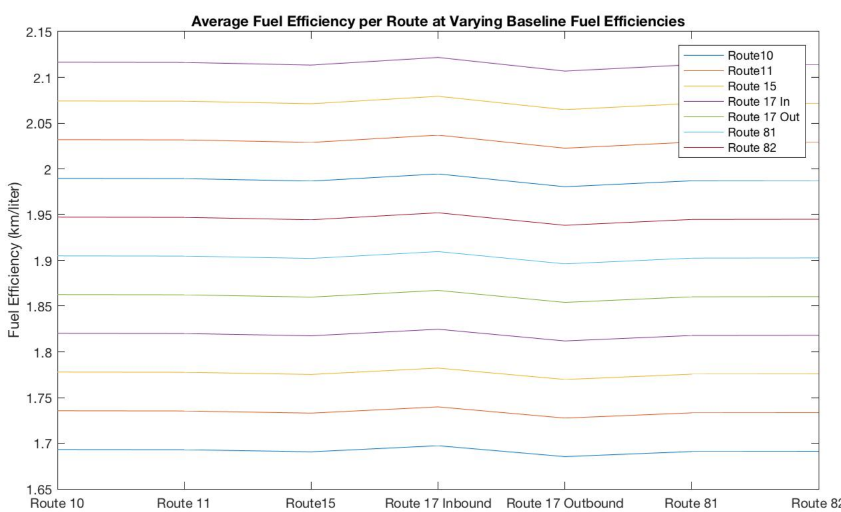


Figure 9: Average Fuel Efficiency per Route at Varying Baseline Fuel Efficiencies

Additionally, for the hybrid buses, an important variable is the maximum gradient that a hybrid bus can climb without relying on its diesel engine. Different values of the variable may distort the bus efficiencies for different routes, since the routes have different topographical characteristics. By Figure 10, we see that Routes 10, 11, and 82 are most sensitive to changes in this maximum value.

If the value of the maximum gradient is not a constant in all situations, as it may decrease in inclement weather, then there may be additional complications in assigning buses to these routes. However, as we do not expect the maximum grade to change drastically from 5% regardless of weather conditions, we do not expect our results to change significantly either. This is especially true because the greatest change in Figure 10 comes from increasing the maximum gradient and inclement weather would ultimately decrease that value, which is less sensitive.

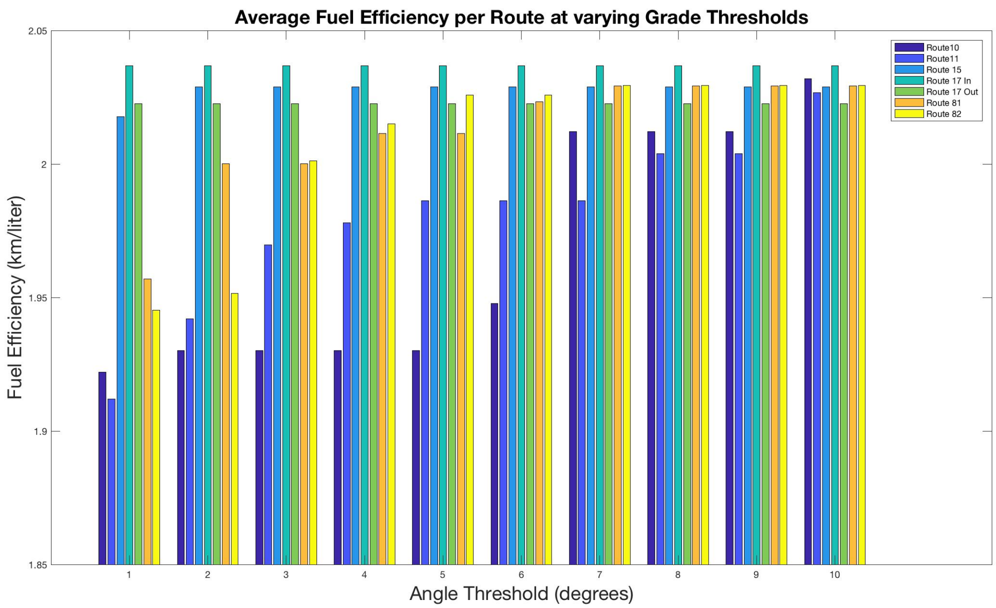


Figure 10: Average Fuel Efficiency per Route at varying Grade Thresholds

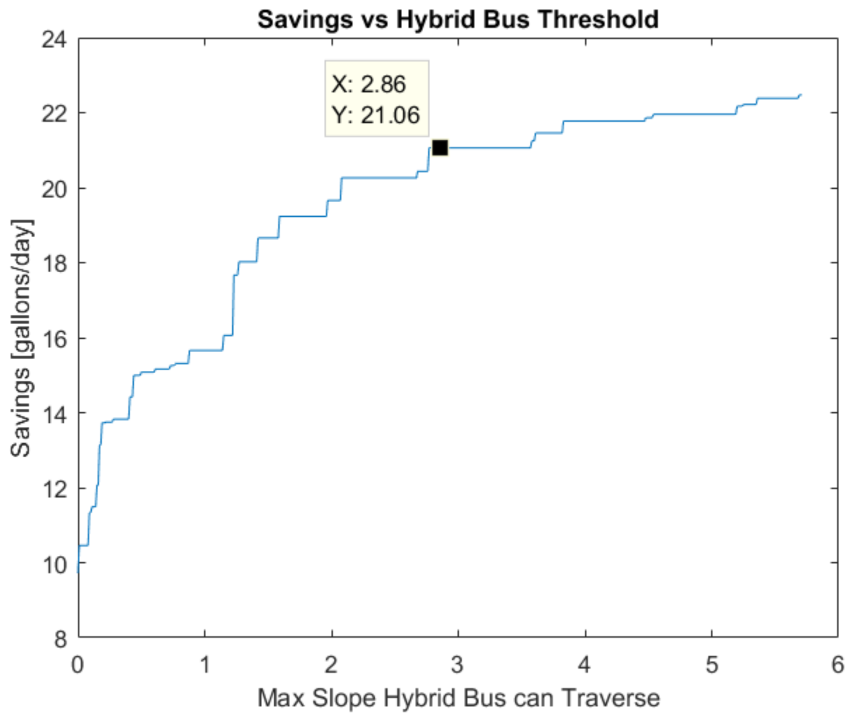


Figure 11: Savings vs Climb Threshold: The marked value is the threshold value 2.86 degrees that we used in our conclusion. The X-axis is in degrees. At 0° or 0% gradient, the buses can only save fuel on downhill roads - around 10 gallons per day. On the other hand, the buses can save about 22 gallons per day with 5.72° or 10% gradient.

We also calculated the gallon savings for varying maximum gradients that the hybrid buses could climb on electricity, a plot of which is shown in Figure 11. **One can see that there are diminishing marginal returns for increased climb threshold, but there is still room for more powerful hybrid buses to improve fuel savings.** Additionally, we can get an estimate of how hybrid buses will perform under worse conditions that lower the threshold at which they can use electric power. Unless the bus is unable to climb even the slightest slope (around half a degree) on electric power, our strategy will still save at least 15 gallons per day.

8 Strengths and Weaknesses

One of the greatest strengths of our model is that our methods of allocating buses and modeling fuel consumption is very modular and can easily be expanded upon. While our model only accounts for the stops on each route without detours, stops can easily be added to our graph of each route. Once this information is added, our algorithms will still work as they did previously and return the new fuel consumption information. Our model can easily be applied to a variety of different paths beyond those explicitly used in this simulation.

Similarly, our function for calculating fuel consumption is independent of the graphs for each route, that function can be improved upon without jeopardizing the efficacy of our route model. Including these parts of the model as independent allows us to revisit, improve, and expand certain parts of our model when new information is revealed without affecting the other parts of our model.

Another strength of our model is the general level of precision in our graphs for each bus stop along each route. While we were not able to account for all weather detours or traffic detours, we were able to get good data on the road length between each bus stop and the change in elevation between them. Assuming constant change in elevation between stops, which may be relatively accurate given the distances are generally quite short, we were able to factor in gradient into our fuel consumption model.

However, our model does have several weaknesses. One of these weaknesses is that we are reliant on several key assumptions that may not hold true in reality. The first of these is that we do not account for traffic flow or potential delays to the buses. The model does not account for traffic signals or congestion at various times that may affect a bus' actual fuel consumption.

Another weakness of our model is that we do not have specific measures for some of the technological aspects of the hybrid bus' engines and batteries. We could not find enough accurate data to model the recharging of the batteries from the diesel engine running or from the regenerative brakes. The effects of regenerative brakes and engine recharge could be magnified by various traffic patterns or from increased braking during downhill segments of the bus route. We made a simplifying assumption that the average fuel efficiency value $e_h = 4.8$MPG accounted for these effects. With additional information about the specific engine and battery systems, we could generate more exact estimates of fuel consumption for each route and create an overall more precise fuel consumption estimate.

Our model also does not account for variation in the routes due to detour or seasonal adjustments. While some of the routes have variations in the case of snow, we did not have adequate information to account for when the routes will change to accommodate these detours. We therefore excluded them from our model and assumed that the routes will operate as normal during the time of our model.

9 Future Work

Our model is very modular and abstract, which makes it simple to expand on. Here are some improvements that can be implemented in the future:

9.1 Fuel Consumption Model Improvements

As stated before, we developed a second model for the fuel consumption of a hybrid bus, but we were unable to use it due to a lack of knowledge of key parameters of the TCAT hybrid buses. Given these parameters, such as maximum TCAT bus charge, mileage per charge, and maximum regeneration rate from regenerative breaking, we would be able to use our second fuel consumption model instead, which would provide more precision in calculating fuel consumption. This can be directly swapped out for our simpler model that we used to make our conclusions, and could potentially change our conclusions.

9.2 Bus Allocation Algorithm Improvements

We focused on optimizing one day of the TCAT Monday-Friday Day schedules (where applicable). However, given more time, we could construct a database including night and weekend schedules, as well as schedules that vary by season. This would allow us to formulate annual fuel consumption predictions and provide better time dependent advice by accounting for changes in the schedule during holidays or university breaks. Hypothetically, with a complete database of every schedule per year, we could use our approach to compute the annual fuel consumption by each bus.

9.3 Alternative Route Recommendations

Our weighted directed graph representation can be incredibly powerful when considering alternative routes. Alternative routes would be represented as additional nodes on the graph, with edges pointing to them. This creates alternative paths along the graph between two points. From this abstraction, we can apply Dijkstra's Shortest Path Algorithm between any two points on our graph to find which alternative routes are better, if any. Dijkstra's Algorithm takes a weighted graph and computes the path between two nodes that has the least total weight. Typically the weights would correspond to distances and the nodes would correspond to different locations, so that Dijkstra's Algorithm could be used to find the shortest path between two nodes. However, since our model weights edges based on gas consumption, the shortest path between two nodes in our model would correspond to the path that consumes the least amount of fuel. Supposing TCAT had a list of alternative stops, we could easily compute the optimal routes with these stops included.

10 Conclusion

In this paper, we created models for each of the bus routes 10, 11, 15, 17, 81, and 82 as well as algorithms for calculating the fuel consumption of diesel and hybrid buses on each route and allocating buses to each route throughout the day. By using our bus allocation algorithm to find which buses are used most frequently and our fuel consumption algorithm to find which routes require the most fuel, we are able to combine these results to find the specific buses that should be replaced with hybrid buses.

By replacing the specific buses with the greatest fuel discrepancy from diesel to hybrid, we create the optimal hybrid bus allocation strategy for TCAT to implement. Our optimal strategy replaces 6 of the buses for Route 82 with hybrid buses and 2 of the buses for Route 11. Making these changes would result in consuming about 21 fewer gallons of diesel fuel every day, which is savings of about $15000 annually.

References

[1] FreeMapTools. "Elevation Finder." https://www.freemaptools.com/elevation-finder.htm. Accessed November 10, 2017.

[2] Tompkins Consolidated Area Transit. https://www.tcatbus.com/. Accessed November 10, 2017.

[3] Wikipedia. "Tompkins Consolidated Area Transit." https://en.wikipedia.org/wiki/Tompkins_Consolidated_Area_Transit#Current_fleet. Accessed November 10, 2017.

[4] Hallmark, Shauna, et al. "Evaluation of In-Use Fuel Economy for Hybrid and Regular Transit Buses." *Journal of Transportation Technologies*, vol. 3, Jan. 2013, pp. 52-57. https://file.scirp.org/pdf/JTTs_2013012517210256.pdf. Accessed November 10, 2017.

[5] Al-Grafi, Mubarak, et al. "Analysis of Vehicle Friction Coefficient by Simulink/Matlab." *International Journal of Control, Automation and Systems*, vol. 2, no. 2, July 2013. http://researchpub.org/journal/jac/number/vol2-no2/vol2-no2-7.pdf. Accessed November 10, 2017.

[6] "Designing Bus Rapid Transit Running Ways." *American Public Transportation Association*, October 2010. http://www.apta.com/resources/standards/Documents/APTA-BTS-BRT-RP-003-10.pdf. Accessed November 10, 2017.

[7] "Gas Prices." AAA. http://gasprices.aaa.com/. Accessed November 12, 2017.

[8] Wikipedia. "Brake Specific Fuel Consumption." https://en.wikipedia.org/wiki/Brake_specific_fuel_consumption. Accessed November 10, 2017.

A Stops Selected for the Routes

As discussed in Section 3.3, our algorithms take in a `.csv` file of start times, end times, origin stops, and destination stops. The latter two are indexed between 1 and N where N is the number of bus stops on the route (stops are counted twice if a bus visits them twice in a loop). For example, Route 10 has 13 stops - it starts from Seneca@Commons in the direction of Tioga@Court and returns to Seneca@Commons after

Stop #	Route 10	Route 81	Route 15
1	Seneca@Commons	A Lot	Seneca@Commons
2	Tioga@Court	Tennis Courts	Geneva@State
3	Court@Linn	Jessup@TripHammer	McGraw House (across)
4	University@Cornell Ave	Risley Hall	Clinton@Fayette
5	University@Lake Street	Goldwin Smith	Plain@Center
6	Boldt Hall	Uris Hall	Plain@Wood
7	Milstein Hall	Corson/Mudd	Titus Towers
8	Goldwin Smith	Bradfield Hall (Opposite)	Elmira@Reuse Center
9	Sage Hall	Dairy Bar	Ithaca Plaza
10	Anabel Taylor Hall	BTI	South Meadow@Wood
11	Schwartz CPA	Dairy Bar (across)	Wegmans
12	College@Mitchell	Bradfield Hall	Tops
13	Seneca@Commons	Kennedy Hall	Memorial Pkwy@Pet Smart
14		Uris Hall (across)	FairGround@Lowes
15		Rockefeller Hall	Walmart
16		Helen Newman	Elmira@Pudgies
17		Credit Farm@Pleasant Grove	Plain@Wood
18		A Lot	Plain@Center
19			Clinton@Fayette
20			McGraw House
21			Green@Commons

Table 6: Stops for Routes 10, 81, and 15.

visiting College@Mitchell, and completes the loop. There are only 12 distinct stops, but a bus returns to Seneca@Commons and reaches its destination.

Tables 6 and 7 lists the stops that our algorithms use in the graph representation.

Stop #	Route 11	Route 82
1	Seneca @ Commons	Hasbrouck
2	Albany and Salvation Army	Appel Commons
3	Green and Commons	Balch@Cradit Farm
4	Aurora and Commons	Goldwin Smith
5	Hudson and Columbia	Uris Hall
6	Hudson and S. Hill Elementary	Corson/Mudd
7	Hudson and Grandview	Bradfield Hall (across)
8	Hudson and Coddington	Dairy Bar
9	Coddington at IC	BTI
10	IC Garden Apts W8	Dryden Rd@Humphrey's Crosswalk
11	Roy Park School	Fairview/Maplewood Park Apts
12	IC Alumni Circle	East Ithaca Apts
13	Tower Road and Gym Road	CISER
14	IC Terrace Apt. E8	Unnamed
15	The Towers	East Hill Plaza
16	IC Terrace Apt. E8	East Hill Office Building
17	College Circle	CISER
18	College Circle Apts	East Ithaca Apts
19	College Circle and Unnamed Road	Fairview/Maplewood Park Apts
20	College Circle	Dryden Rd@Humphrey's Crosswalk
21	IC Terrace Apt. E8	Vet School
22	The Towers	Dairy Bar (across)
23	IC Terrace Apt. E8	Bradfield Hall
24	Tower Road and Gym Road	Kennedy Hall
25	Alumni Circle (Park Hall)	Uris Hall (across)
26	Roy Park School	Rockefeller Hall
27	IC Garden Apts E8	Balch@Thurston
28	Coddington at IC	Jessup@Triphammer
29	Hudson and Coddington	RPCC
30	Hudson and Pearsall	Jessup@Pleasant Grove
31	Hudson and S. Hill Elementary	Hasbrouck
32	Hudson and Columbia	
33	Seneca @ Commons	

Table 7: Stops for Routes 11, 82.

A.1 `.csv` File Format of Schedules for Bus Allocation Algorithms

The first column is that start time at the origin stop; the second column is the end time at the destination stop; the third column is the index of the origin stop (1 represents the 1^{st} stop on TCAT's schedule listing [2]); the fourth column is the index of the destination stop.

Here's an example of the CSV file, containing data for Route 11.

Listing 1: data_files/11.csv

```
657,748,1,15
727,818,1,15
757,848,1,15
827,918,1,15
857,948,1,15
927,1018,1,15
957,1048,1,15
1027,1118,1,15
1057,1148,1,15
1127,1218,1,15
1157,1248,1,15
1227,1318,1,15
1257,1348,1,15
1327,1418,1,15
1357,1448,1,15
1427,1518,1,15
1457,1548,1,15
1527,1618,1,15
1557,1648,1,15
1627,1718,1,15
1657,1748,1,15
1727,1818,1,15
1757,1848,1,15
1827,1918,1,15
1857,1948,1,15
1927,2018,1,15
1957,2048,1,15
2027,2118,1,15
```

A.2 Implementation of Fuel Consumption - Model 1

Listing 2: data_files/gasForHybridPath.m

```
function litres = gasForHybridPath(elevation1,elevation2,distance,threshold)
%GASFORPATH estimates gas consumption for path between two stops for diesel
%bus
%   elevation1 is the elevation of the starting stop, and elevation2 is the elevation of
    the
%   destination stop.

mu = 0.7; %For dry road conditions
```

```

fuel_efficiency = 4.8*1.6/3.78541; %Fuel efficiency in km/litre
fe_diesel = 1.78;

%threshold = 2.86;

slope = (elevation2-elevation1)/distance;
angle = asin(slope);
degrees = angle*360/(2*pi);
if degrees < threshold
    k = (mu*cos(angle)+sin(angle))/mu;
    litres = (distance/fuel_efficiency)*k;
else
    k = (mu*cos(angle)+sin(angle))/mu;
    litres = (distance/fe_diesel)*k;
end

end
```

A.3 Implementation of Algorithm 1 in Java

Listing 3: data_files/Bus.java

```
public class Bus {
    private int id;
    private boolean used;
    private int endTime;
    private int trips;

    public Bus(int id, int endTime) {
        this.id = id;
        this.used = false;
        this.endTime = endTime;
        this.trips = 0;
    }

    public int getID() {
        return this.id;
    }

    public int getTrips() {
        return this.trips;
    }

    public int getEndTime() {
        return this.endTime;
    }

```

```
    public boolean isUsed() {
        return this.used;
    }

    public void incTrips() {
        this.trips++;
    }

    public void setUsed(boolean b) {
        this.used = b;
    }

    public void setEndTime(int time) {
        this.endTime = time;
    }

    @Override
    public String toString () {
        return "id = " + this.id + " used = " +
                this.used + " endTime = " + this.endTime;
    }
}
```

Listing 4: data_files/simulateBusesOnRoute.java

```
import java.util.*;
import java.io.*;

public class simulateBusesOnRoute {

    public static Bus[] estimateNumberOfBuses(int totalNumBuses,
            Integer [][]calendar) {

        Stack<Bus> availableBuses = new Stack<>();

        for (int i = 0; i < totalNumBuses; ++i)
            availableBuses.push(new Bus(i, 0));

        Queue<Bus> busyBuses = new LinkedList<>();

        for (int i = 0; i < calendar.length; ++i) {
            if (busyBuses.isEmpty()) {
                Bus newBus = availableBuses.pop();
                newBus.setUsed(true);
                newBus.setEndTime(calendar[i][1]);
                newBus.incTrips();
                busyBuses.add(newBus);
            } else {
                // some buses are already running
```

```
            while (!busyBuses.isEmpty()
                    && busyBuses.peek().getEndTime() < calendar[i][0]) {
                availableBuses.push(busyBuses.remove());
            }

            if (availableBuses.isEmpty()) {
                System.out.println("Too few buses!");
                break;
            } else {
                Bus newBus = availableBuses.pop();
                newBus.setEndTime(calendar[i][1]);
                newBus.setUsed(true);
                newBus.incTrips();
                busyBuses.add(newBus);
            }
        }
    }
    while (!busyBuses.isEmpty())
        availableBuses.push(busyBuses.remove());

    return availableBuses.toArray(new Bus[(availableBuses.size())]);
}

public static Integer[][] readCSVfile(String csvFileName, int skipLines) {

    String line = "";
    ArrayList<Integer[]> startEndTimes = new ArrayList<>();
    try {
        BufferedReader br = new BufferedReader(new FileReader(csvFileName));
        while (skipLines-- > 0) br.readLine();
        while ((line = br.readLine()) != null) {

            String[] data = line.split(",");
            Integer[] currStartEndTimes =
                {Integer.parseInt(data[0]), Integer.parseInt(data[1])};
            startEndTimes.add(currStartEndTimes);
        }
    } catch (FileNotFoundException e) {}
    catch (IOException e) {}

    return startEndTimes.toArray(new Integer[startEndTimes.size()][]);
}

public static void writeToFile(String fname, ArrayList<Integer> arr) {

    try {
        BufferedWriter writer =
            new BufferedWriter(new FileWriter(fname, false));
        writer.append("# simulations using simulateBusesOnRoute.java\n");
```

```
            writer.append("bus_id,trip\n");
            for (int i = 0; i < arr.size(); ++i)
                writer.append(i + "," + arr.get(i) + "\n");
            writer.close();
        } catch (IOException e) {}
    }
    public static void main(String []args) {

        if (args.length < 2)
            System.out.println("Too few arguments!");
        else {
            Integer [][] calendar = readCSVfile(args[0],
                Integer.parseInt(args[1]));

            Bus[] buses = estimateNumberOfBuses(55, calendar);

            ArrayList<Integer> idToTripsMap = new ArrayList<>();
            for (int i = 0; i < buses.length; ++i) {

                int thisBusTrips = buses[i].getTrips();
                if (thisBusTrips > 0) {
                    // this bus was used
                    idToTripsMap.add(thisBusTrips);
                }
            }

            Collections.sort(idToTripsMap);
            writeToFile("simBusesRouteFull_" + args[0], idToTripsMap);
        }
    }
}
```

A.4 Integrated Model with Extension to Algorithm 1 in MATLAB

Listing 5: data_files/gasForHybridPath.m

```
function litres = gasForHybridPath(elevation1,elevation2,distance,threshold)
%GASFORPATH estimates gas consumption for path between two stops for diesel
%bus
%   elevation1 is the elevation of the starting stop, and elevation2 is the elevation of the
%   destination stop.

mu = 0.7; %For dry road conditions

fuel_efficiency = 4.8*1.6/3.78541; %Fuel efficiency in km/litre
fe_diesel = 1.78;

```

```
%threshold = 2.86;

slope = (elevation2-elevation1)/distance;
angle = asin(slope);
degrees = angle*360/(2*pi);
if degrees < threshold
    k = (mu*cos(angle)+sin(angle))/mu;
    litres = (distance/fuel_efficiency)*k;
else
    k = (mu*cos(angle)+sin(angle))/mu;
    litres = (distance/fe_diesel)*k;
end

end
```

Listing 6: data_files/getGasForHybridRoute.m

```
function gas = getGasForHybridRoute( TCAT_data ,threshold)
%GETGASFORROUTE Gets gas consumption for a whole TCAT route in one day.
%   TCAT_data should be a 2 column matrix [elevation, distance to next stop]
num_stops = length(TCAT_data(:,1))-1;
gas = 0; %total gas in litres
for i = 1 : num_stops;
        gas = gas + gasForHybridPath(TCAT_data(i,1),TCAT_data(i+1,1),TCAT_data(i,2),
            threshold);
end
end
```

Listing 7: data_files/getGasForTripHybrid.m

```
function gas = getGasForTripHybrid(startIndex,endIndex,keys, data,threshold)
%GETGASFORTRIP Summary of this function goes here
%   Detailed explanation goes here

start = find(keys(:,1)==startIndex);
start = keys(start,2);
destination = find(keys(:,1)==endIndex);
destination = keys(destination,2);

dataSubset = data(start:destination,:);

gas = getGasForHybridRoute(dataSubset,threshold); %Change method used to calculate for
    different buses
gas = gas * 0.264172;
end
```

Listing 8: data_files/gasForHybridDay.m

```
function [buses] = gasForHybridDay(calendar,key,data,threshold)
```

```matlab
%GASFORDAY Summary of this function goes here
%   Detailed explanation goes here

buses = zeros(0,2);
available = 0;
startTimes = calendar(:,1);
endTimes = calendar(:,2);

for i = 1:length(calendar(:,1))

    gas = getGasForTripHybrid(calendar(i,3),calendar(i,4),key,data,threshold);
    available = length(find(buses(:,2)<startTimes(i)));

    if(available == 0)
        buses = [buses;gas,endTimes(i)]; % add a bus that does this trip
    else
        busIndices = find(buses(:,2)<startTimes(i));
        nextBus = busIndices(1);%Get first bus that's available
        buses(nextBus,1) = buses(nextBus,1) + gas; %add gas consumption by this bus
        buses(nextBus,2) = endTimes(i); %update finish time for the bus
    end
end
```