Rebalancing of Shared Micromobility Services Through Dynamic Pricing

Talha Alvi
Electrical & Computer Engineering
University of Toronto
Toronto, Ontario
talha.alvi@mail.utoronto.ca

Farhan Wadia
Mechanical & Industrial Engineering
University of Toronto
Toronto, Ontario
farhan.wadia@mail.utoronto.ca

I. Introduction

Micromobility is an emerging trend shaping urban transportation today that consists of using small vehicles such as bikes, scooters, or skateboards to easily navigate in highly populated urban areas. Bike sharing and similar micromobility solutions help achieve sustainable mobility as a solution to the last-mile problem in transportation, while also helping to provide inexpensive and equitable transportation access to historically under-served communities [1]. With this project, the aim is to look at the challenges associated with providing optimal availability of a shared micromobility service to its users, and addressing the coupled issue of oversupply in areas with low demand and shortages in areas with high demand during operations. Specifically, this project will use data from the City of Toronto's Bike Share system (TBS) to address the problem of providing an adequate number of bikes at every service station, without an excess of bikes at some stations and shortage of bikes at other stations.

II. Problem Characterization and Background

The goals of this project are to look at some of the specific challenges associated with operating a shared micromobility system, and how some of those concerns can be alleviated by dynamic pricing and incentive schemes for users, or through manual intervention by operators, with a focus on using data from TBS. For casual users of TBS, the existing system has a tiered pricing structure offering options such as single trips ($3.25), unlimited daily passes ($7.00), and unlimited 3-day passes ($15). All trips must be completed within 30 minutes to avoid an overage fee of $4 per 30 minute interval exceeded. Additionally, TBS sells $99 and $115 annual usage passes for more frequent riders; the $99 pass allows for unlimited 30 minute trips before the overage fee applies, and the $115 pass allows for unlimited 45 minute trips before the overage fee applies [2].

TBS's existing pricing structure is relatively simple, and does not make use of maximum demand utilization strategies such as surge pricing during times of high demand, that other mobility operators like Uber & Lyft use. Use of dynamic pricing to take advantage of additional demand, or to induce demand during times and in areas of low usage would help TBS increase revenue, while still satisfying the existing demand from its users. Even though TBS is not designed to be revenue neutral, there are still opportunities to increase revenue by utilizing more intelligent pricing models. Casual riders of TBS effectively subsidize annual members; 2022 projections show that casual riders comprise 26% of the ridership, yet contribute 52% to total revenues. Casual rider revenue per trip is $3.66, whereas annual member revenue per trip is only $0.91. Given operating costs of $1.88 per trip, TBS is profitable on casual trips and unprofitable for annual member trips. On average, TBS's annual members take 120 trips per year [3].

In addition to dynamic pricing to take advantage of excess demand, a key operational challenge for bike share operators like TBS is addressing the supply and demand imbalances that build into the system at different times and different locations based on user ridership patterns. This causes some stations to have an excess of bikes, while others have no bikes available for users to rent. Resolving this issue requires manual intervention by the bike share operator to collect bikes from some stations and relocate them to other stations via truck transport. Although manual intervention is generally considered the most effective approach towards resolving this rebalancing problem and will always be needed, a newer approach is to also try and provide users incentives such as reduced or negative pricing, or extra usage time so that they move bikes between certain locations on-demand to rebalance the system [4] [5]. Bike share operator Vélib', based in Paris, France, is one example of a real world system which has tried the user incentive scheme to improve bike availabilities across their network; in particular, Vélib' gave users extra riding time rather than a fare discount [4]. For this project, the purpose is to evaluate how a dynamic pricing scheme can be implemented for TBS to increase bike availabilities across the network, reduce the extent of manual rebalancing required, and increase revenue by satisfying trips which would have previously gone unfulfilled.

For micromobility services such as TBS in general, the rebalancing problem is important to solve in order to increase the efficacy and sustainability of micromobility towards solving the last mile transportation problem, and to reduce congestion in dense, urban city centres. For TBS in particular,

improved solutions to the rebalancing problem would help to significantly reduce costs (the average cost of repositioning a single bike is $3 in 2009 dollars), and potentially even increase revenue by attracting customers and ensuring that bikes are always available to serve them [6].

Some of the key challenges associated with solving the rebalancing problem via a dynamic pricing approach, which make the optimization problem ill-structured, are as follows:

- Difficulty determining optimal distribution of bikes or scooters at their respective stations around the city, and to balance supply and demand as a function of time.
- How to predict user behaviour and how it will evolve over time to be able to identify supply/demand patterns
- The effect of incentives on impacting user behaviour to fulfill desired outcomes for system rebalancing, and the price elasticity of demand with respect to these incentives

III. Literature Review

Rebalancing strategies for station-based bike sharing (SBBS) systems like TBS can be classified broadly into two categories: operator-based strategies, and user based strategies. With SBBS, bikes must always start from and return to a designated station, which is different from free-floating bike sharing (FFBS) where bikes do not necessarily have to be placed at designated stations [7].

A. Operator-Based Strategies

Operator-based strategies involve the use of repositioning trucks to load bikes and move them from areas with excess supply to areas with shortages. When rebalancing is done at times of low or no demand (i.e. overnight) to be able to prepare for the next day's operations, it is referred to as static rebalancing, whereas continuous rebalancing throughout the course of a day is referred to as dynamic rebalancing. Static rebalancing has the advantage of truck routing not being affected by congestion and other vehicular traffic, but it does not address supply issues throughout the course of the day. Bike share operators broadly use either or both of these approaches. Static rebalancing can be further classified into complete rebalancing, where all stations get set to meet the target inventory levels, or partial rebalancing where there are still some imbalances due to constraints such as lack of rebalancing time or not enough rebalancing vehicles available. The former is referred to as the Static Complete Rebalancing Problem (SCRP) and the latter is referred to as the Static Partial Rebalancing Problem (SPRP). User based strategies involve providing incentives to users such as reduced & negative pricing, or extra riding time to move bikes between locations of the operator's choice.

The initial approach to SCRP and SPRP was formulated by Raviv et al., and extended by Pfrommer et al. as well as several other researchers [7] [8] [9]. Raviv presented two different approaches, both of which are mixed integer linear programs with a convex objective function. The Raviv model itself is an extension of the one-commodity pickup-and-delivery travelling salesman problem (1-PDTSP) [8]. 1-PDTSP is an extension of the travelling salesman problem (TSP), where nodes correspond to customers that can either provide or require known amounts of a single commodity, and the objective is to visit each node exactly once using a vehicle of finite capacity, and satisfy the demand at each node while minimizing the total travel distance. Since the item is considered a commodity, an item picked up from any node can be delivered to any other node requiring that item [10]. An exact approach exists to solve 1-PDTSP for up to 50 nodes, and several other solutions using genetic algorithms and heuristics exist to solve the problem for larger numbers of nodes; the best-known approach is able to solve instances with up to 1000 nodes [11].

Pfrommer et al. extends the Raviv model in two main ways. The first way is by including a greeedy, promising route heuristic so that each truck chooses nodes that provide the highest utility per unit of additional travel time. Secondly, they tackle the issue of including a dynamic pricing scheme to incentivize users to modify their trips and route bikes between certain stations to rebalance the system better during operations [9]. Assuming a walking speed of half the cycling speed, Pfrommer et al. create a Voronoi partition to find the effective distance from a point to the closest station, and to determine which stations can be considered neighbours. The incentive to entice users to travel to a neighbouring station rather than a desired station is sampled from a uniform distribution, and customer behaviour is assumed to be rational in going to a neighbouring station if that maximizes utility. Although these works handle SCRP and SPRP for multiple vehicles, the mathematical formulation of the problem assumes only one vehicle exists. To resolve this, nodes must be decomposed into multiple redundant nodes at the same location, each with only unit absolute imbalance. In the decomposed network, the number of visits to each node will then be at most one in order to restore balance, since each unbalanced node would have a surplus or deficit of only one item [7]. However, increasing the number of nodes like this still makes the problem difficult to solve algorithmically in reasonable time since it remains NP-hard, and therefore heuristic approaches will always be needed to find feasible solutions.

To be able to predict inventory and extend rebalancing to be dynamic, Schuijbroek et al. formulate inventory levels as a Markov Decision Process (MDP) where user behaviour is non-stationary and the rates of demand at a station can change over time, but that users pickup and dropoff bikes following a time-dependent exponential distribution. Based on this, inventory at any station can be modelled as a M/M/1 queue, and the probability of a station having a certain number of bikes at any given time can be calculated. They then use heuristics and similar methods as other researchers to solve the routing problem [12].

B. User-Based Strategies

In addition to the model developed by Pfrommer et al. which considers dynamic pricing, Fricker and Gast developed a model where users are presented a choice between two station destinations at the time of rental, with an incentive provided to go to the station with lower capacity. Their model assumes all stations have the same capacity for docking bikes, and they show that for every 25% increase in users deciding to go to the incentivized station, the proportion of unbalanced stations in the network decreases by a factor of 10. Like Schuijbroek et al., they predict inventory levels using a MDP [13].

Zulqarnain et al. view the dynamic pricing problem as a bi-level decision making process, and formulate the problem as an integer program that can be solved using genetic algorithms [4]. The operator's objective is to minimize the number of unbalanced stations, which leads them to modify prices provided to users. Users' objective is to minimize their travel cost. Based on the incentives provided, users will modify their travel patterns to help satisfy the operator's objective. An issue our team sees with this formulation is that it assumes users will be acting rationally, and in the interest of the bike share operator; in reality, users should not be assumed as being rational [14]. The model does not consider user inconvenience in modifying their start or end locations, the cost of how users value their own time, and the price elasticity of demand that would get users to consider switching their trip start and end points for a high enough incentive.

For FFBS, Zhang et al. developed a dynamic pricing model which considers negative prices (i.e. paying users) to complete desired trips. They consider walking, bus travel, and biking, and associate transferring and convenience costs between and for each of these transportation methods. They partition the search space into traffic analysis zones (TAZs), physically representing transport stations near each other as clusters, and form what they call a supernetwork out of it. The supernetwork is similar to a normal graph structure, but nodes corresponding to physical locations can be repeated to account for the transport option taken to or from the node (i.e. walking, bike, or bus). The model considers providing a negative price if users go from a TAZ with oversupply to a TAZ with undersupply, and positive prices otherwise. The positive price is fixed for all journeys it applies to, and the negative price linearly relates to the level of undersupply at the destination, up to a maximum. Negative pricing is expected to lead to two types of travel pattern changes in their model; the first is that users who might have otherwise decided to walk and not use a bike become incentivized to move a bike to an undersupplied location, and the second change is that users might change their travel paths if the benefit of the negative price outweighs the inconvenience of starting/ending further away from their originally desired travel points. By incentivizing users to move a bike to an undersupplied location, that bike then becomes available for another user's trip that can bring in positive revenue, whereas previously, the second user would not have been able to make a trip. The objective of their model is to minimize disutility (considering time, money, and comfort costs per transit mode), subject to network flow constraints on the graph [5].

Pan et al. show how the dynamic pricing solution to the rebalancing problem can be solved via deep reinforcement learning, and propose a divide-and-conquer algorithm called hierarchical reinforcement pricing (HRP) for this problem [15].

They model the problem mathematically by dividing the region spatially and temporally, and then defining functions for the supply of available bikes as a function of time, the number of bikes arriving at a station as a function of time, and the demand for bikes at a given station as a function of time. These functions are vectorized to consider all the station locations in the network. If a user is at a location from where they want to originate their trip, no price incentive is provided. However, if there is no bike available, but there is a bike available at an adjacent zone/station, then there is an inconvenience cost associated with the user having to walk to a neighbouring station. The cost function varies quadratically with the walking distance. Based on a real-world survey, user inconvenience costs are generally considered to have a convex form such as this. A user will accept an incentive to walk to an adjacent zone if the incentive is equal to or exceeds their inconvenience cost. The budget to provide these incentives is finite for each day. In the event that even all the adjacent stations have no available bikes, the user's trip will remain unfulfilled. To model the supply and demand dynamics at each station, supply at the next hour is considered to be equal to the supply at the beginning of the current hour, plus the net arrivals within the hour [15].

As a reinforcement learning problem, the environment (or state) at each timestep would be the set of: supply at each station, remaining budget available for incentives, demand at the previous timestep, net arrivals at the previous timestep, expense of incentives provided in the previous timestep, and the cumulative amount of unfulfilled trips up to the current timestep. The action is to provide a sequence of incentives, which lead to an immediate reward of the number of satisfied requests in the region at that timestep. A transition probability is associated with moving from one state to another, and the goal is to learn a policy such that the overall discounted rewards over the entire time horizon are maximized (the discount factor essentially defines how much future rewards are discounted relative to immediate rewards) [15].

IV. Problem Formulation & Modeling

The modelling strategy implemented is to model the problem as a Markov Decision Process (MDP). A MDP is a framework for modelling decision making situations for which the outcome is probabilistic. It represents a system as a series of states and transitions between those states, with a corresponding reward for each transition. The goal in a MDP formulation is to find the optimal policy, which is a sequence of decisions or actions that maximizes the expected reward over time. The MDP formulation for the bike share system is

outlined as follows, similar to approach outlined by Pan et. al, but adapted to a docked system with a simplified modelling to reduce state complexity [15].

Environment: The system spatial and temporal environment can be represented as an area divided into n zones represented by a vector $Z = (z_1, z_2, ..., z_n)$. Each day is divided into T time slots of equal length represented as $T = (1, 2, 3, ..., 24)$. The supply and demand in each zone can be represented as a vector $S_i(t)$, where denotes the supply in each zone i at a time slot t. The total user demand and bike arrivals at each zone z_i are represented. by $D_i(t)$ and $A_i(t)$ respectively. The number of the users at each time step going from one zone z_i to another z_j is represented by $d_ij(t)$.

Pricing: The users are offered a price incentive if the demand is in excess of the current supply. The price incentive incentivizes users to walk to the nearest zone that has excess supply. The nearest zone can be denoted by $N(r_i)$ with some distance x representing the distance to the zone from the initial zone r_i. The pricing incentive is subject to a total rebalancing budget B, which is reset each day. When the budget is spent for the day, no additional incentives can be offered and all excess demand that may have been filled is assumed lost.

User Cost: The user in this model is represented as an independent agent that accepts a given trip, based on the cost to him being lower than the price offered. If the price is higher than the cost borne as a result of walking to the nearest zone, the trip is lost. This cost can be represented as a quadratic function with respect to the distance x, and can be scaled to the preferred dollar value by adjusting the coefficients of the quadratic polynomial. This cost can be represented formally as $c(i, j, x)$. If this cost is lower than the price $p_{ij}(t)$ offered to the user, that means the user is obtaining some positive utility, based on the offered price and therefore accepts the trip. Conversely, if the cost to the user is higher than the trip is rejected.

Supply & Demand: The supply and demand dynamics of the system are as follows: $d_{ij}(t)$ denotes the number of users taking a trip from from zone i to j and $a_{ji}(t)$ represents the arrivals from each zone j to zone i in each time step. The travel time is assumed to be uniform for all trips and is fulfilled within one time step.

$$S_i(t+1) = S_i(t) - \sum_{j=1} d_{ij}(t) + \sum_{j=1} a_{ji}(t) \tag{1}$$

Model Summary: The MDP model is summarized as follows: the state of the environment is represented as $s(t) = (\mathbf{S}(t), \mathbf{D}(t-1),\ \mathbf{D}_f(t-1), B(t))$, where $\mathbf{S}(t)$ represents the supply vector for the environment, $\mathbf{D}(t-1)$ represents the demand vector at the previous time step, $\mathbf{D}_f(t-1)$ represents the fulfilled demand vector at the previous time step and $B(t)$ represents the overall remaining budget. The action is represented by $a(t) = (p_{1t}, p_{2t}, ..., p_{nt})$, which represents the price determined by the bike share operator, also referred to as the agent that learns the optimal policy. The agent receives some rewards $R(s_t, a_t)$, based on the state, action set, which is the number of trips that are fulfilled for the excess demand. The goal of this problem is to learn an optimal policy $\pi_\theta(s_t)$ that maps the state of the environment to an action that represents the best possible action for the given state. The policy shall maximize the overall discounted rewards starting from s_0 over some time horizon. The discount factor $\gamma \in [0, 1]$ represents the discount factor, which optimizes rewards over time, with 0 representing immediate reward and 1 representing rewards over an infinite horizon.

V. Solution

A. Implementation Details

The above model is implemented for the Toronto Bike Share system using the OpenAI Gym API specification. The specification allows modelling of reinforcement learning type algorithms using a uniform approach, and differs from the Hierarchical Reinforcement Pricing (HRP) Algorithm developed in [15]. The specification initializes an action space, and an observation space, which represent the action $a(t)$ and state $s(t)$ vectors for the problem. A reset and a step function a used to run the agent, where at each step the demand for each zone is calculated, based on the available City of Toronto Bike Share data set. The data set includes information for start time, end time, start station, and end station. In the problem the stations are split up into square zones of a uniform size of 300 m x 300 m, which represents one zone. The station data is correlated to the zone using geographic coordinates of each station to the geographic boundaries that represent each zone. Python Geopandas and Shapely libraries are used to aid this computation. After demand is calculated for the given time step, the demand is compared to the existing supply for each of the zones to find excess demand that can not be fulfilled due to supply ¡ demand at that zone. In that case, the nearest geographic zones are calculated for each of the zones where there is excess demand. For each of the nearest zones, the euclidean distance is calculated and the available supply is tallied up as well. The user is offered a price form the agent for zone where the initial trip is started from and the user accepts the trip from the nearest zone if the price offered is less than the cost determined, based on the distance. The cost in this specific implementation is represented as the quadratic function: $2.2685x^2 - 0.000645x + 0.8418$. The coefficients here can be adjusted to alter the specific trade offs for the distance in comparison to the cost. The shifted demand for the user is updated as fulfilled demand for the nearest zone for which the trip was accepted. Using this information the reward for the time step is calculated as the sum of excess demand for each zone that is fulfilled, i.e. the number of trips that were shifted to other zones due to low supply.

B. Algorithms

Two different reinforcement learning algorithms A2C and SAC were used to try and learn the optimal policy, based on the MDP formulation implemented using the Gym environment. These are relatively popular RL algorithms and the implementation used is from on the Stable Baselines 3 (SB3) library.

1) Advantage Actor Critic (A2C): In the actor-critic methods, the actor is responsible for determining the action to take in a given state, while the critic is responsible for evaluating the quality of those actions. The critic estimates the value of each state-action pair and provide this information to the actor, which can use it to improve its policy. The A2C algorithm borrows from temporal difference (TD) learning as well to use the difference between the predicted value of a state and the actual reward received to update the value function. The A2C algorithm uses the outputs of the both actor and the critic to update the parameters of the policy and value functions. A2C is an on policy algorithm, meaning that it uses the current policy to collect the data and update the policy. This can be more stable for learning, but it is a trade off as it is more sensitive to the initial policy choice. Further details for the algorithm can be reviewed as part of the paper Mnih et. al. [16].

2) Soft Actor Critic (SAC): The SAC algorithm again uses an actor critic method for RL, but it combines the maximum entropy principle for learning. In SAC, the actor is trained to maximize the trade off between the expected reward and the entropy of the policy, while the critic estimates the value of each state-action pair. The entropy in this case represents randomness or uncertainty, which allows the actor the explore the environment, which can lead to more efficient learning. The algorithm employs a linear rate of entropy that decreases linearly with the number of training steps, this allows it to do more exploration at the start of training and more exploitation towards the end. This is an off-policy algorithm that uses old data to update its policy. The figure below provides the pseudo code for the algorithm.

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$
  end for
end for

Fig. 1: SAC Algorithm [17].

C. Evaluation & Training

1) Training: Training is completed using the A2C and SAC algorithms for at least 2000 time steps. The training results are outlined below in terms of average reward over each training episode and the normalized actor and critic training loss to see how the training loss improves over the training time.

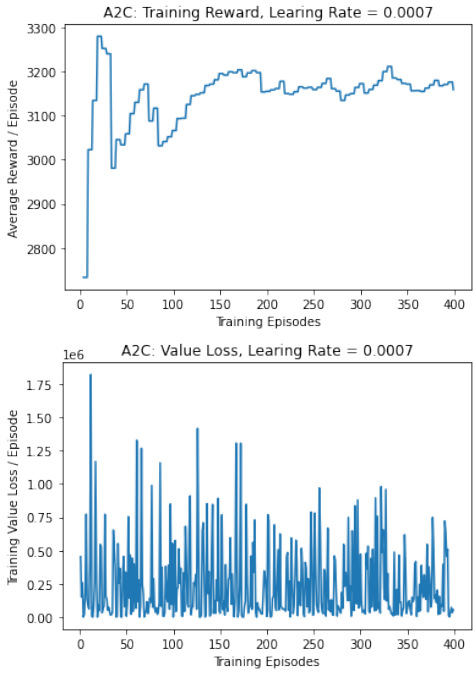

Fig. 2: Training Reward & Loss for A2C

Fig. 2 shows the training reward and loss for the A2C model. The model eventually begins to converge towards an average reward of around 3200 / episode, but takes a long time and still has quite a few large swings as it is still attempting to learn an optimal policy. The value loss for the training period does start to go down eventually, but there are still huge changes in magnitude. This is an indication that the model is not able to learn the optimal policy and that the errors in the learned model will be high.

Fig. 3 shows the reward and loss for the SAC model over the training horizon. As shown, the training reward is more stable in its convergence in comparison to the A2C model. The training reward does dip at around middle of training, but that is likely due to the algorithm exploring its state space, due to its use of a stochastic policy to learn and a higher learning rate. The model starts to converge immediately afterwards to an optimal average reward. The chart showing the critic loss over the training period shows that the SAC model is able to learn an optimal policy much more effectively over its training period.

Fig. 4 outlines the training rewards and loss for SAC with a learning rate that is a magnitude lower than that of the previous figure. As can be seen, the model is more stable in the way it learns compares to a learning rate of 0.0007. The downside of a smaller learning rate would be that it may take a longer time to converge to an optimal value, but it does provide more stability.

2) Evaluation: Evaluation is completed for the trained SAC model and results are detailed below in Fig. 5. The evaluation

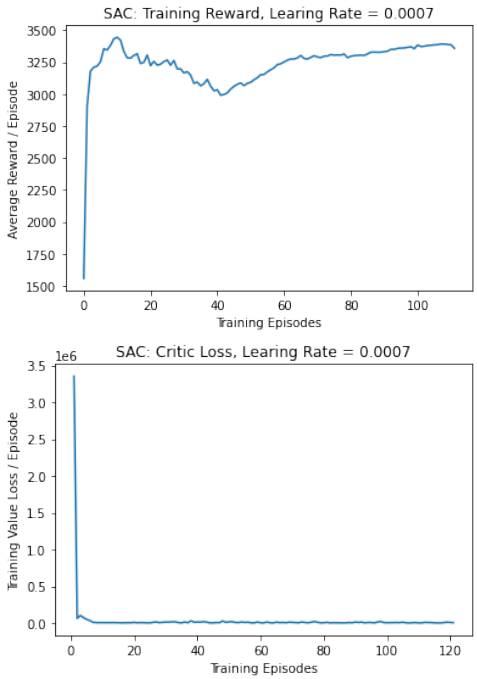

Fig. 3: Training Reward & Loss for SAC, LR=0.0007

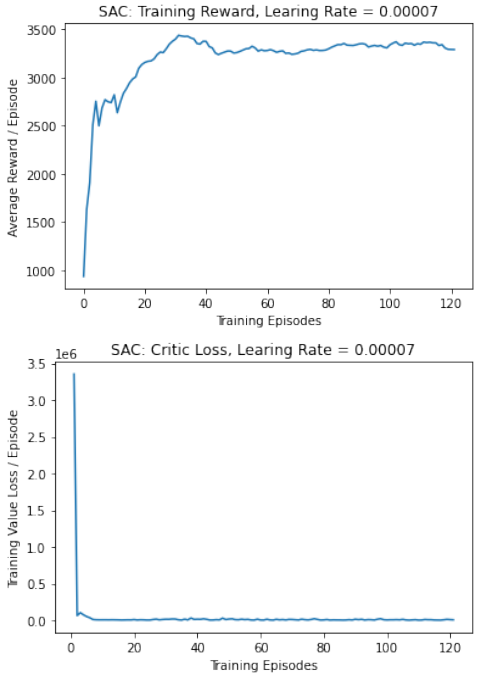

Fig. 4: Training Reward & Loss for SAC, LR=0.00007

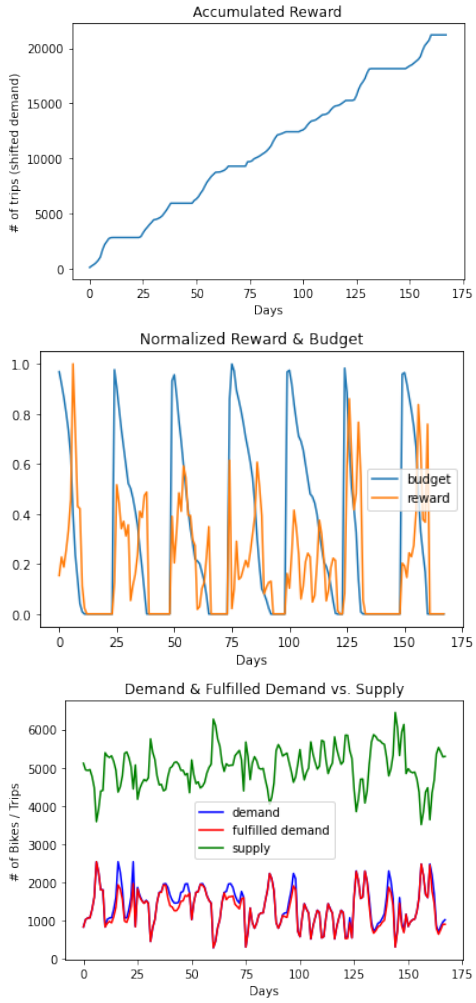

Fig. 5: Training Results for 7 Day Evaluation Period

focuses on the rewards collected over a period of one week or 168 episodes. The accumulated rewards chart shows the sum total number of trips that are fulfilled by the operator that are in excess of supply at each zone. The reward and budget graph shows the comparison between the normalized budget and corresponding rewards over the evaluation period. The budget slowly decreases overtime to the incentives offered, the reward goes to zero as well, since the budget has run out. This shows that sufficient budget needs to be allocated in order to meet the overall excess demand. This is something that can be experimentally determined, based on training data and seeing at what point the average rewards stop decreasing if the budget is further increased. The optimal budget to set for each episode or day can help the operator make the trade off between fulfilling as much excess demand as possible, while minimizing the amount spent each day offering incentives. The last graph in the figure shows the demand curve in comparison to the fulfilled demand over the evaluation period. The overall supply at each step can also be seen. It should be noted that this graph fails to represent the spatial supply and demand, as that is what is really needed to reflect where the supply and

demand mismatch is and how that is resolved through this modelling.

VI. Conclusion

To summarize, this project shows how the rebalancing problem for a micromobility service provider such as TBS can be partially alleviated through providing user incentives, while simultaneously providing a new revenue stream of trips which were previously unfulfilled due to spatial and temporal imbalances in supply and demand. Although the use of static rebalancing trucks will always be needed for any micromobility service provider, and the majority of literature related to this topic focuses on the operator-based rebalancing approach, dynamic pricing is a great complementary policy to improve the user experience as well as operator profitability. Prior literature shows how dynamic pricing can be modelled as an integer program or network flow problem, but reinforcement learning as shown in the literature and in this paper allows for the ability to form a more nuanced model while still being solvable.

With this project specifically, over the 7 day evaluation period used for the SAC model, 21,213 additional trips were fulfilled that would have previously gone unfulfilled. The incentives provided for these trips total $21,000 ($3,000 per day for 7 days). Although TBS has multiple pricing options based on membership types, assuming for simplicity that the fare for each of these rides was $3.25, then the total revenue from these additional trips is $68,942.25, and the net profit for the week thanks to these additional fulfilled trips is $47,942.25. Although TBS is not profitable on an annual basis since annual member trips have a net loss of $0.97 per trip while making up 74% of ridership, the remaining 26% of casual users provide a net profit of $1.78 per trip [3]. Therefore, implementing the dynamic pricing strategy to provide to casual riders would clearly help to increase profits, while improving rider satisfaction and availability of bikes throughout the network.

References

[1] A. Khamis, "4.5 - Micromobility," in *Smart mobility: Exploring foundational technologies and wider impacts*, New York, NY: Apress, 2021, pp. 93–94.

[2] "Pass Options," *Bike Share Toronto*. [Online]. Available: https://bikesharetoronto.com/pricing/. [Accessed: Oct. 16, 2022].

[3] J. Hanna, "Bike Share Business Update Q2 YTD," *Toronto.ca*, Jul. 26, 2022. [Online]. Available: https://www.toronto.ca/legdocs/mmis/2022/pa/bgrd/backgroundfile-229023.pdf. [Accessed: Oct. 16, 2022].

[4] Z. Haider, A. Nikolaev, J. Kang and C. Kwon, "Real-time Dynamic Pricing for Bicycle Sharing Programs", U.S. Department of Transportation, 2014. [Online]. Available: http://utrc2.org/sites/default/files/Final-Report-Real-time-Dynamic-Pricing-Bicycle-Sharing.pdf. [Accessed Oct. 16, 2022].

[5] J. Zhang, M. Meng, and D. Z. W. Wang, "A dynamic pricing scheme with negative prices in dockless bike sharing systems," *Transportation Research Part B: Methodological*, vol. 127, pp. 201–224, 2019.

[6] P. DeMaio, "Bike-sharing: History, impacts, models of provision, and future," *Journal of Public Transportation*, vol. 12, no. 4, pp. 41–56, 2009.

[7] A. Pal and Y. Zhang, "Free-floating bike sharing: Solving real-life large-scale static rebalancing problems," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 92–116, 2017.

[8] T. Raviv, M. Tzur, and I. A. Forma, "Static repositioning in a bike-sharing system: Models and solution approaches," *EURO Journal on Transportation and Logistics*, vol. 2, no. 3, pp. 187–229, 2013.

[9] J. Pfrommer, J. Warrington, G. Schildbach and M. Morari, "Dynamic Vehicle Redistribution and Online Price Incentives in Shared Mobility Systems," *Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1567–1578, 2014.

[10] H. Hernández-Pérez and J.-J. Salazar-González, "The one-commodity pickup-and-delivery travelling salesman problem," *Combinatorial Optimization — Eureka, You Shrink*, pp. 89–104, 2003.

[11] H. Hernández-Pérez, I. Rodríguez Martín, and J.-J. Salazar-González, "Introduction to Pickup-and-Delivery Problems," *Pickup-and-delivery site*, 30-Dec-2020. [Online]. Available: http://hhperez.webs.ull.es/PDsite/#:%20:text=The%201%2DPDTSP%20is%20a,minimizing%20the%20total%20travel%20distance. [Accessed: Oct. 16, 2022].

[12] J. Schuijbroek, R. C. Hampshire, and W.-J. van Hoeve, "Inventory rebalancing and vehicle routing in bike sharing systems," *European Journal of Operational Research*, vol. 257, no. 3, pp. 992–1004, 2017.

[13] C. Fricker and N. Gast, "Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity," *EURO Journal on Transportation and Logistics*, vol. 5, no. 3, pp. 261–291, 2016.

[14] D. Kahneman, "Maps of Bounded Rationality: Psychology for Behavioral Economics," *The American Economic Review*, vol. 93, no. 5, pp. 1449–1475, 2003.

[15] L. Pan, Q. Cai, et. al., "A Deep Reinforcement Learning Framework for Rebalancing Dockless Bike Sharing Systems," *arXiv*, 1802.04592v4 [cs.AI] 2 Dec 2018

[16] V. Mnih, A. Badia, et. al., "Asynchronous Methods for Deep Reinforcement Learning," *arXiv*, 1602.01783 [cs.LG] 16 Jun 2016

[17] T. Haarnoja, A. Zhou et. al., "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor"*arXiv*, 1801.01290 [cs.LG] 08 Aug 2018