Maximum Throughput Dispatch for Shared Autonomous Vehicles Including Vehicle Rebalancing

Jake Robbennolt and Michael W. Levin

***Abstract*—Shared autonomous vehicles (SAVs) provide on demand point-to-point transportation for passengers. This service has been extensively studied using dispatch heuristics and agent based simulations of large urban areas. However, these approaches make no mathematical guarantees of the passenger throughput for the SAV network. This study builds on the dynamic queuing model design of Kang and Levin which provides a maximum stability dispatch policy for SAVs. This model is extended to include rebalancing of empty vehicles to regions of high demand. The modified dispatch policy is proven to maximize throughput. Simulation results show that this dispatch policy reduces waiting times (between vehicle dispatch and passenger pickup) compared to the original formulation. However, vehicle time traveling empty increases in some scenarios. Simulation results also show that rebalancing often reduces passenger waiting times, but not when too many vehicles rebalance at once and are not available for dispatch.**

***Index Terms*—Maximum throughout, shared autonomous vehicles, rebalancing.**

I. Introduction

THE use of shared autonomous vehicles (SAVs) can introduce many societal benefits. This transportation mode combines the emerging technology of autonomous vehicles with on demand taxi dispatching services using a shared fleet of vehicles to meet traveler demand. Several studies have shown benefits to the introduction of SAVs such as reduced parking needs, environmental benefits, and cost reductions [2], [3], [4], [5].

However, there are still open questions about the operation of SAVs, including the fleet size and the dispatch policy. Most studies of SAV operations have utilized agent based simulations to determine characteristics of network, demand, and dispatch policy that affect the fleet requirements. The results vary significantly with the dispatch policy used, so it is valuable to develop dispatch policies that achieve mathematical guarantees of performance. One such mathematical guarantee is stability of passenger queue lengths and/or waiting times, which also guarantees that SAV service rate will be equal to customer demand.

Kang and Levin [1] and Li et al. [6] have both developed dispatch policies with mathematical guarantees of maximum throughput. The dispatch policy developed by Kang and Levin [1] allows vehicle rebalancing and utilizes the same network structure as this paper. However, they rely on a planning horizon for their proof of stability. Li et al. [6] utilizes a different network structure and requires dispatch costs to be computed using an S-only algorithm which requires a probability distribution of service times that may not be known in practice. Their proposed policy also does not allow for vehicle rebalancing. Finally, Xu et al. [7] found a dispatch policy that can be proven to stabilize the network using the same network structure as Kang and Levin [1], but without the need for a planning horizon. As in Xu et al. [7], we define a fleet of single occupancy SAVs giving passengers rides on demand from specified origins to destinations. We build on this methodology by incorporating a vehicle rebalancing scheme into the dispatch policy. This is an important modification since it anticipates future demand and moves vehicles to it before travelers enter the network, reducing waiting times. However, such a policy must balance the reduction in waiting times with an increase in vehicles driving empty to locations where they are unnecessary. The proposed dispatch policy can still be proven to be stable and is shown to reduce waiting times for travelers.

The contributions of this paper are as follows: We formulate an SAV dispatch problem and propose a dispatch policy which incorporates rebalancing. We prove that the proposed dispatch policy can serve all demand if any policy can serve all demand. We demonstrate the minimum fleet size necessary to stabilize any demand (for any dispatch policy). We create a simulation model and use the Sioux Falls and Winnipeg networks to demonstrate the reduction of passenger waiting times when rebalancing is included. Finally, we compare the function of the proposed dispatch policy with another from the literature and show the benefits of the maximum throughput property.

II. Background

A. Max-Pressure Control

Tassiulas and Ephremides [8] first proposed a maximum stability policy called backpressure control to schedule data transfers in multihop radio networks. Later, Varaiya [9] and

Manuscript received 29 July 2022; revised 24 January 2023; accepted 10 April 2023. This work was supported by the National Science Foundation under Award 1935514. The Associate Editor for this article was J. Blum. *(Corresponding author: Jake Robbennolt.)*

Jake Robbennolt is with the Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: robbe091@umn.edu).

Michael W. Levin is with the Department of Civil, Environmental, and Geo-Engineering, University of Minnesota, Minneapolis, MN 55455 USA.

Digital Object Identifier 10.1109/TITS.2023.3268158

Wongpiromsarn et al. [10] realized the similarities to traffic networks and separately developed max-pressure traffic signal control for traffic networks. The controllers developed by Wongpiromsarn et al. and Varaiya are very similar, and both have been proven to have a maximum stability property. This means that even though they are decentralized, they can serve any demand that can be served by any signal timing scheme.

Based on these initial max-pressure traffic signal control schemes, several other authors made some modifications such as utilizing vehicle delay and travel times [11], [12]. In addition, Xiao et al. [13] included the jam-density of each turning movement. Later, Gregoire et al. [14] defined a pressure function which continued to take queue length as an input but also considered the buffer size of the link. Several authors suggested modifications to ensure the traffic signal control remained cyclic [15], [16], [17]. Finally, Sun and Yin [18] and [19] studied max-pressure traffic signal control in microsimulation to demonstrate travel time and queue length reduction. Simulation studies on max-pressure control have demonstrated the value of the maximum stability property. However, stability does not provide any guarantees on travel times, delay, or environmental impacts.

The proof of maximum stability has also been used to incorporate route choice into traffic signal timing [20], [21], [22], [23]. In addition, Chen et al. [24] considered max-pressure algorithms for pedestrian access to autonomous intersections. Finally, Levin et al. [25] considered how max-pressure algorithms can be used for dynamic lane reversals at intersections with autonomous vehicles. These studies provided a framework for adopting a similar methodology for SAV dispatch.

The fundamental idea behind max-pressure control is that when signal timing is insufficient to meet demand, queue lengths grow with no bound. Levin et al. [9] formulated traffic signal timing as a Markov decision process and proved that some traffic signal controllers cause the queue length to be bounded if any traffic signal timing can bound the queues. A similar methodology can be applied to a system of SAVs in which the fleet size is fixed and the goal is to stabilize the number of waiting passengers. In this paper we build on previous work on traffic stability to prove that a proposed SAV dispatch policy is stable.

B. SAVs

In general, the SAV dispatch problem is similar to a taxi routing problem or a dial-a-ride problem. Much previous work on SAVs has taken the form of agent based simulation. Fagnant and Kockelman [4] showed environmental benefits of SAVs and demonstrated that 1 SAV could replace 10 private vehicles. Chen et al. [26] examined electric SAVs and found them to be cost-competitive with other transit modes. Other agent based simulations have shown that SAVs have the potential to reduce congestion, parking needs, and emissions [2], [6], [27], [28], [29], [30].

Of particular concern is the vehicle dispatch policy which determines which vehicles to assign to each customer. This is very closely related to the taxi dispatching problem studied by Seow et al. [31] and Maciejewski and Nagel [32]. These problems involve assigning passengers to either the nearest idle taxi, or include taxis still en-route to drop off passengers. Ge et al. [33], Hyland and Mahmassani [34], Gurumurthy et al. [35], and Xu et al. [36] have applied similar strategies to optimize SAV dispatch. However, none of these approached have considered the long-run network throughput of the policy.

Some research has been done specifically on vehicle repositioning (rebalancing). This approach was first proposed in by Pavone et al. [37] and has since been studied in detail [38], [39]. Rebalancing is intended to bring vehicles to locations where they are needed before passengers request them in order to reduce waiting times. Though waiting times can often be reduced, research has suggested that repositioning can increase vehicle miles traveled and increase congestion [40], [41]. Based on app based ride hailing, these studies show up to 45% of vehicle miles are traveled empty. However, other work suggests that large benefits in mode switching outweigh these costs [42]. In addition, there are large travel time savings for passengers if the repositioning is optimized [43]. Though previous studies have shown benefits of rebalancing schemes, few have shown mathematical proofs of stability. This is a drawback because rebalancing can reduce the number of vehicles available to carry passengers for a given fleet size. This can result in an unstable network where the number of waiting passengers grows to an arbitrarily large number.

Finally, a distinction should be made between model based dispatch policies (such as the one developed here) and model free policies such as reinforcement learning. Many of the previously listed studies are model based, which means they use the known system dynamics to make optimal decisions. Such studies approach the problem either through mathematical optimization or through simulation [38], [44], [45], [46]. Reinforcement learning is a newer approach to solving the SAV dispatch problem that does not need to rely on system dynamics. Various studies have demonstrated that reinforcement learning dispatch policies can have strong performance [47], [48], [49], [50], [51], [52], [53]. However, these methods can lead to less intuitive results and they do not lend themselves well to an analytical characterization of their properties. This is of notable importance in comparison to the policy developed in this study which can be demonstrated to maximize throughput.

Three studies examined maximum throughput dispatch for SAVs. Kang and Levin [1] used a similar queuing model to that presented in this paper, though they utilized a planning horizon in their stability proof. Their model allows vehicle rebalancing, and their results show benefits when demand is asymmetric. However, the required planning horizon reduced the computational efficiency of their dispatch policy, and preemptive rebalancing was not an emphasis of their paper. Li et al. [6] examined the stability of an electrified SAV fleet. Though they include electric vehicle charging constraints in their model, they do not include any vehicle rebalancing. In addition, they modeled the vehicles and passengers as nodes in their network. The proof of stability requires dispatch costs to be computed using an S-only algorithm, assuming that network service times are known at all times. These service times are generally not known in advance, so this assumption is often unrealistic. Finally, Xu et al. [7] utilized a network of zones where passengers are picked up and dropped off and proved that their policy stabilized demand. They did not require a planning horizon in their proof, but did not include rebalancing. Other policies in the literature may maximize throughput, but have not yet been proven to. We extend the mixed integer linear program defined by Xu et al. [7] to include vehicle rebalancing between trips and prove that the network remains stable.

III. Basic Stability Analysis

In this section we define an SAV dispatch policy and prove that it is stable. We assume that passengers send requests for pickup on-demand and include a requested origin and destination. The dispatcher then sends a vehicle to pick them up and take them to their destination as soon as possible. In this study we assume that the dispatcher has full knowledge and control of each SAV in the network and that each SAV serves only 1 passenger at a time. We also assume that no passengers exit the network unserved, and that once an SAV is dispatched to serve a passenger it cannot be reassigned. Finally, we assume that links in the network have constant travel times (that SAVs do not affect the level of congestion).

A. Queueing Model

Consider a network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with nodes $\mathcal{N}$ and links $\mathcal{A}$. Passengers can be picked up or dropped off only at nodes. SAVs have unlimited parking at nodes when between trips and can carry only one passenger at a time. We also assume constant travel times on all arcs and define C_{qrs} to be the travel time from node q to node r to node s. Similarly, C_{qs} is the travel time from q to s. These travel times could include congestion, but are constant with respect to SAV dispatch.

Let $d_{rs}(t)$ be the new demand for travel from r to s at time t. $d_{rs}(t)$ is a random variable with mean $\bar{d}_{rs}$. $w_{rs}(t)$ is defined as the number of travelers at node r waiting to be picked up to travel to destination s and not yet assigned to a vehicle. To assign SAVs to each of these passengers the vehicle must travel from its starting location to pick up the passenger and then continue to the drop off. Let $v_{qrs}(t)$ be the number of SAVs assigned to travel from q to r to carry a passenger from r to s (it is possible to have $q = r$). Then, $w_{rs}(t)$ evolves as follows:

$$w_{rs}(t+1) = w_{rs}(t) + d_{rs}(t) - \sum_{q \in \mathcal{N}} v_{qrs}(t) \tag{1}$$

Vehicles dispatched on a trip from q to r to s, will travel for time C_{qrs} before arriving at their destination. In addition, we can imagine another set of vehicles that are dispatched for rebalancing trips straight from q to s without picking up any passengers (e_{qs}). This set of vehicles will travel through the network with time C_{qs} before also arriving at s. To track SAVs that are enroute, let $x_q^\tau(t)$ be the number of vehicles that are τ time steps away from q:

$$x_q^\tau(t+1) = \begin{cases} x_q^{\tau+1}(t) + \sum\limits_{(s,r)\in\mathcal{N}^2 : C_{srq}-1=\tau} v_{srq}(t) \\ + \sum\limits_{s\in\mathcal{N} : C_{sq}-1=\tau} e_{sq}(t) & \tau \geq 1 \\ x_q^0(t) + x_q^1(t) - \sum\limits_{(r,s)\in\mathcal{N}^2} v_{qrs}(t) \\ - \sum\limits_{s\in\mathcal{N}} e_{qs}(t) & \tau = 0 \end{cases} \tag{2}$$

When a SAV departs on trip $[q, r, s]$ at time t, the SAV will arrive at s at time $t + C_{qrs}$. Then at time $t + 1$ it will have $C_{qrs} - 1$ travel time remaining, so that SAV is added to $x_s^{C_{qrs}-1}(t+1)$. The same logic is used for rebalancing trips with travel times C_{qs}. From this formulation, we define $x_q(t) = x_q^0(t)$ to be the number of SAVs available at node q at time t.

Two additional constraints bound the dispatch of SAVs. Vehicles cannot be assigned to carry a passenger unless a traveler is waiting:

$$\sum_{q\in\mathcal{N}} v_{qrs}(t) \leq w_{rs}(t) \quad \forall r, s \in \mathcal{N}^2 \tag{3}$$

In addition, the number of waiting vehicles $x_q(t)$ bounds the number of SAVs that can carry passengers:

$$\sum_{(r,s)\in\mathcal{N}^2} v_{qrs}(t) \leq x_q(t) \quad \forall q \in \mathcal{N} \tag{4}$$

We assume that the fleet size of SAVs is a constant F. Therfore, the fleet size F can be related to SAV locations by summing over the number of SAVs at each location:

$$F = \sum_{q\in\mathcal{N}} \sum_{\tau=0}^{\infty} x_q^\tau(t) \tag{5}$$

The state consists of the waiting passengers and SAV locations ($\boldsymbol{w}(t)$ and $\boldsymbol{x}(t)$). The control is the vehicle dispatching via $v_{qrs}(t)$ and $e_{qs}(t)$, and the control space varies based on the available SAVs defined by $\boldsymbol{x}(t)$.

B. Stable Network

Using the network model defined in Section III-A, our goal is to define a dispatch policy for which the network remains stable. Stability is defined so that it is equivalent to serving all demand. Following the logic of Varaiya [9], we define the *stability* of the network as follows:

Definition 1: The network is stable if the expected number of waiting passengers remains bounded over time. Thus, the network is stable if there exists a $\kappa < \infty$ such that

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{(r,s)\in\mathcal{N}^2} \mathbb{E}\left[w_{rs}(t)\right] \leq \kappa \tag{6}$$

Definition 1 can only be achieved if SAVs are dispatched for passengers at the same rate at which passenger demand enters the network. Otherwise, the average number of waiting passengers would gradually increase over time past any bound.

C. Maximum-Stability Policy

A dispatch policy $\pi^\star$ is proposed which incorporates the network constraints listed in Section III-A. This policy seeks to minimize the total travel time SAVs need to travel to serve demand as well as minimizing a penalty to ensure rebalancing. This policy will be proven to be stable in Section III-E.

The proposed policy $\pi^\star$ is defined as follows. At each time step t, solve the integer linear program:

$$\min \sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}(t)C_{qs} + \delta(t) \times \sum_{s\in\mathcal{N}} a_s(k, z_s(t), x_s(t), e_{qs}(t), v_{qrs}(t)) \quad (7a)$$

$$\text{s.t.} \sum_{q\in\mathcal{N}} v_{qrs}(t) \le w_{rs}(t) \quad \forall (r,s) \in \mathcal{N}^2 \quad (7b)$$

$$\sum_{(r,s)\in\mathcal{N}^2} v_{qrs}(t) + \sum_{s\in\mathcal{N}} e_{qs}(t) \le x_q(t) \quad \forall q \in \mathcal{N} \quad (7c)$$

$$\sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}(t) = \min\left\{ \sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t), \sum_{q\in\mathcal{N}} x_q(t) \right\} \quad (7d)$$

$$v_{qrs}(t) \in \mathbb{Z}_+ \quad \forall (q,r,s) \in \mathcal{N}^3 \quad (7e)$$

$$e_{qs}(t) \in \mathbb{Z}_+ \quad \forall (q,r,s) \in \mathcal{N}^3 \quad (7f)$$

$$\delta(t) \in \{0,1\} \quad (7g)$$

To avoid the solution $v_{qrs}(t) = 0$, equation (7d) sets the number of vehicles dispatched equal to the number of waiting passengers or the number of available vehicles, whichever is more limiting. In addition, this term ensures that as the number of waiting passengers increases the rebalancing term will approach zero. If there are fewer available vehicles than there are waiting passengers, all vehicles will be assigned to waiting passengers. This property is important in the proof of stability as described in Section III-D and III-E.

In the objective function, $a_s(\cdot)$ is a penalty for a large imbalance of vehicles (too few or too many) at node s. $a_s(\cdot)$ is used to encourage rebalancing to avoid these large imbalances. This penalty is left general for now, and an example is provided later on. The penalty will be a function of z_s, an exogenous variable denoting the ideal number of vehicles in each zone s, as well as $x_q(t)$, $v_{qrs}(t)$, and $e_{qs}(t)$. This term sets a penalty for a large imbalance of vehicles, and can be controlled by the scalar parameter k which can be optimized in simulation.

The parameter δ is a function of $\boldsymbol{w}(t)$ and $\boldsymbol{x}(t)$. It needs to be defined each timestep, but is constant with respect to this integer linear program. We define $\delta = 0$ if $\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) \ge \sum_{q\in\mathcal{N}} x_q(t)$. Otherwise, $\delta = 1$. Then, the penalty a_s will be removed unless $\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) \le \sum_{q\in\mathcal{N}} x_q(t) \le F$, which is an important requirement for stability as shown in Section III-E. Also important for the proof of stability is the requirement that a_s be bounded when $\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) \le \sum_{q\in\mathcal{N}} x_q(t) \le F$. An example of a function matching these requirements is given in Section III-F.

The resulting optimal solution $v^\star_{qrs}(t)$ is used to determine which SAVs to dispatch and which passengers they are assigned to serve. $e^\star_{qs}(t)$ is used to determine which vehicles will be rebalanced and to where. The stability properties of $\pi^\star$ are proved in Section III-E.

Before proceeding to the stability analysis, it is important to note the relationship between the exogenous demand and the number of rebalancing trips. As demand increases the number of rebalancing trips must decrease, leading to Proposition 1:

Proposition 1: As the number of people waiting to be picked up $w_{rs}(t)$ increases, the number of rebalancing trips $e_{qs}(t)$ will become zero.

Proposition 1 and future Propositions and Lemmas are proved in the Appendix.

D. Stable Region

For any dispatch policy, it is possible for demand to be so high that it is impossible to serve all passengers. This results in an unstable network where passenger queues grow with no bound. We will prove that $\pi^\star$ stabilizes the network whenever possible. In order to prove that the proposed policy $\pi^\star$ achieves maximum stability, we first must establish when it is possible for the network to be stable.

The SAV dispatch policy defines vehicle trips $v_{qrs}(t)$ and $e_{qs}(t)$ each time step. We define $\bar{v}_{qrs}$ and $\bar{e}_{qs}$ to be the average number of SAVs dispatched per time step. We use $\bar{v}_{qrs}$ and $\bar{e}_{qs}$ to refer to the dispatch behavior from any dispatch policy. The stable region is intended to describe the possible passenger service of any dispatch policy so we can compare $\pi^\star$ to it analytically.

$$\bar{v}_{qrs} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} v_{qrs}(t) \quad (8)$$

$$\bar{e}_{qs} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} e_{qs}(t) \quad (9)$$

The average dispatch rate to pick up passengers $\bar{v}_{qrs}$ can be related to passenger demand. This makes it possible to determine a fleet size F necessary to serve the average demand rate $\bar{\boldsymbol{d}}$. Since it is necessary to serve all demand:

$$\sum_{q\in\mathcal{N}} \bar{v}_{qrs} = \bar{d}_{rs} \quad \forall (r,s) \in \mathcal{N}^2 \quad (10)$$

Constraint (10) is given with strict equality because SAVs trips $v_{qrs}(t)$ are made only when a passenger is waiting and all rebalancing trips are encompassed by $e_{qs}(t)$. SAVs must obey conservation of flow:

$$\sum_{(q,r)\in\mathcal{N}^2} \bar{v}_{qrs} + \sum_{q\in\mathcal{N}} \bar{e}_{qs} \quad (11)$$

$$= \sum_{(q,r)\in\mathcal{N}^2} \bar{v}_{sqr} + \sum_{r\in\mathcal{N}} \bar{e}_{sr} \quad \forall s \in \mathcal{N} \quad (12)$$

The fleet size F bounds the number of SAVs that can be dispatched at any time step. Dispatched vehicles v_{qrs} and e_{qs} travel for C_{qrs} and C_{qs} time steps respectively. Over a long time horizon T:

$$\sum_{t=1}^{T}\left[\sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}(t)C_{qs}\right] \leq F \times T \tag{13}$$

Taking the limit as $T \to \infty$:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\left[\sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}(t)C_{qs}\right] \leq F \tag{14}$$

Equivalently:

$$\sum_{(q,r,s)\in\mathcal{N}^3} \bar{v}_{qrs}C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} \bar{e}_{qs}C_{qs} \leq F \tag{15}$$

To simplify this equation we eliminate the $\sum_{(q,s)\in\mathcal{N}^2} \bar{e}_{qs}C_{qs}$ term since the stable region should not decrease as rebalancing trips are added.

Proposition 2: The stable region will stay the same if $\sum_{(q,s)\in\mathcal{N}^2} \bar{e}_{qs}C_{qs} = 0$.

Proposition 2 and Equation (15) yield:

$$\sum_{(q,r,s)\in\mathcal{N}^3} \bar{v}_{qrs}C_{qrs} \leq F \tag{16}$$

Based on equation (16), we can define an average time $\bar{C}_{rs}$ required to serve a passenger from r to s. This time includes C_{rs} and also the empty travel required to send a vehicle to r, but does not include travel times for rebalancing vehicles:

$$\bar{C}_{rs} = \frac{\sum_{q\in\mathcal{N}} \bar{v}_{qrs}C_{qrs}}{\sum_{q\in\mathcal{N}} \bar{v}_{qrs}} \tag{17}$$

Using the average time to serve each passenger, equation (16) can be rewritten as:

$$\sum_{(r,s)\in\mathcal{N}^2} \bar{C}_{rs}\sum_{q\in\mathcal{N}} \bar{v}_{qrs} \leq F \tag{18}$$

Using equation (10), the demand can be substituted for the average dispatch:

$$\sum_{(r,s)\in\mathcal{N}^2} \bar{C}_{rs}\bar{d}_{rs} \leq F \tag{19}$$

Let the stable region $\mathcal{D}$ be the set of demands for which there exists a $\bar{\boldsymbol{v}}$ and $\bar{\boldsymbol{e}}$ satisfying constraints (10), (12), and (16). Let $\mathcal{D}^0$ be the interior of $\mathcal{D}$, (where constraint (16) holds with strict inequality). Then there exists an $\epsilon > 0$ such that

$$\sum_{(r,s)\in\mathcal{N}^2} \bar{C}_{rs}\bar{d}_{rs} - F = \sum_{(q,r,s)\in\mathcal{N}^3} \bar{v}_{qrs}C_{qrs} - F \leq -\epsilon \tag{20}$$

Proposition 3: If $\bar{\boldsymbol{d}} \notin \mathcal{D}$, then there does not exist a stabilizing control.

Proposition 3 shows that any demand outside of $\mathcal{D}$ cannot be stabilized, i.e. it cannot be served by any dispatch policy. In Section III-E we show that the dispatch policy $\pi^\star$ serves any demand $\bar{\boldsymbol{d}} \in \mathcal{D}^0$. Though no policy can serve demand on the boundary of $\mathcal{D}$, $\pi^\star$ serves as much demand as any other dispatch policy and achieves maximum throughput.

E. Stability Analysis

For any demand $\bar{\boldsymbol{d}} \in \mathcal{D}^0$, the dispatch policy $\pi^\star$ defined in Section III-C is now proven to be stable. This means that it can serve all demand whenever possible by any dispatch policy. Proposition 3 proved that any demand $\bar{\boldsymbol{d}} \notin \mathcal{D}$ cannot be stabilized, so this will prove that $\pi^\star$ achieves the maximum stability property. Based on Theorem 2 of Leonardi et al. [54], Definition 1 can be achieved by proving the following:

Lemma 1: When policy $\pi^\star$ is used and $\bar{\boldsymbol{d}} \in \mathcal{D}^0$, there exists a Lyapunov function $\nu(\boldsymbol{w}(t)) \geq 0$ and constants $\kappa > 0$, $\epsilon > 0$ such that

$$\mathbb{E}\left[\nu(\boldsymbol{w}(t+1)) - \nu(\boldsymbol{w}(t))\,|\,\boldsymbol{w}(t),\boldsymbol{x}(t)\right] \leq \kappa - \epsilon|\boldsymbol{w}(t)| \tag{21}$$

Finally, based on Lemma 1, Proposition 4 demonstrates that the proposed dispatch policy $\pi^\star$ can serve all demand if any policy can:

Proposition 4: When policy $\pi^\star$ is used and $\bar{\boldsymbol{d}} \in \mathcal{D}^0$, the network is stable.

F. Rebalancing Penalty Function

In Section III-C, the term a_s was left general in the objective function. This allowed a proof of stability for a class of rebalancing strategies with different a_s terms. This term is a penalty on an imbalance of vehicle throughout the network. The only requirement of this function is that it be bounded when $\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) \leq \sum_{q\in\mathcal{N}} x_q(t) \leq F$.

Such a function is simple to create when bounded by the fleet size. For example, we define $a_s(t)$ as a constant times the absolute difference between the ideal vehicles in each zone and the number parked or rebalancing there. Recall that z_s is an exogenous variable denoting the ideal number of vehicles in each zone. a_s can be controlled by the scalar parameter k which can be optimized in simulation. We note that $x_s + \sum_{(q)\in\mathcal{N}} e_{qs}(t) - \sum_{(q)\in\mathcal{N}} e_{sq}(t)$ denotes the number of vehicles at or rebalancing to node s. Thus, $z_s - \left(x_s + \sum_{(q)\in\mathcal{N}} e_{qs}(t) - \sum_{(q)\in\mathcal{N}} e_{sq}(t)\right)$ is bounded by at most the fleet size. Thus, we define $a_s(t)$:

$$a_s(t) = k \times \left| z_s - \left(x_s + \sum_{(q)\in\mathcal{N}} e_{qs}(t) - \sum_{(q)\in\mathcal{N}} e_{sq}(t)\right)\right| \quad \forall (s)\in\mathcal{N} \tag{22}$$

To linearize this function, we add the following constraints to policy $\pi^\star$:

$$a_s(t) \geq k \times \left(z_s - \left(x_s + \sum_{(q)\in\mathcal{N}} e_{qs}(t) - \sum_{(q)\in\mathcal{N}} e_{sq}(t)\right)\right) \quad \forall(s) \in \mathcal{N} \tag{23}$$

$$a_s(t) \geq -k \times \left(z_s - \left(x_s + \sum_{(q)\in\mathcal{N}} e_{qs}(t) - \sum_{(q)\in\mathcal{N}} e_{sq}(t)\right)\right) \quad \forall(s) \in \mathcal{N} \tag{24}$$

There are many other functions satisfying the proof of stability above. However, Equations (23) and (24) are used to define $a_s(t)$ in the numerical examples below.

G. Fleet Size Optimization

One of the important implications of the stability analysis is that the minimum stable fleet size ($F^\star$) to serve all demand can be determined exactly. That solution is presented below:

$$\min F \tag{25a}$$

$$\text{s.t.} \sum_{q,r,s\in\mathcal{N}^3} C_{qrs}\bar{v}_{qrs}(t) \leq F \tag{25b}$$

$$\sum_{(q)\in\mathcal{N}} \bar{v}_{qrs}(t) = \bar{d}_{rs} \quad \forall r, s \in \mathcal{N}^2 \tag{25c}$$

$$\sum_{(q,r)\in\mathcal{N}^2} \bar{v}_{qrs}(t) = \sum_{(q,r)\in\mathcal{N}^2} \bar{v}_{sqr}(t) \quad \forall s \in \mathcal{N} \tag{25d}$$

$$\bar{v}_{qrs}(t) \in \mathbb{Z}_+ \quad \forall(q, r, s) \in \mathcal{N}^3 \tag{25e}$$

We note that this formulation does not include the rebalancing vehicles e_{qs} since maximum stability can be achieved without rebalancing as proven in Proposition 2.

IV. Simulation and Results

The Sioux Falls and Winnipeg networks were used for simulation to demonstrate the impact of rebalancing on performance [55]. Since stability with and without rebalancing has already been proven, the purpose of these results is to show that the inclusion of rebalancing improves performance when the fleet size is greater than the minimum ($F > F^\star$). The Sioux Falls network includes 24 nodes and 76 links, and the Winnipeg network has 1052 nodes and 2836 links. For the Sioux Falls network, two demand scenarios are included to demonstrate the effects of symmetric and asymmetric demand. The first is the base scenario from [55] which has a demand of 15,025 trips per hour. Modifying this slightly, we used 150% of the demand from nodes 1–13 and only 50% of the demand from nodes 13–24 to create an asymmetric demand scenario. The Winnipeg network is included to demonstrate the performance of the dispatch policy on a more realistic network. This scenario has a total of 64,784 trips per hour. In both cases demand is stochastic and added following a Poisson distribution. These scenarios are run an a simulation created in Java using IBM ILOG CPLEX version 12.9 to solve the dispatch optimization. The dispatch policy is real-time implementable, as the optimization program takes approximately 13.3 seconds to run per timestep (in a real-world implementation, dispatch would take place every 15–30 seconds). Simulations were run on a laptop computer with an Intel Core i7-1165 at 2.80 GHz and 16 GB of RAM.

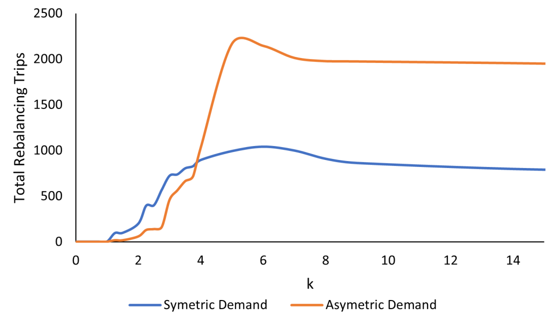

Fig. 1. Sioux Falls - Total rebalancing trips throughout the simulation.

A. Impact of Demand Distribution

These results will examine the effects of modifying the parameter k on performance for the symmetric and asymmetric demand scenarios for the Sioux Falls network. Included metrics are the total number of rebalancing trips, the average time between vehicle dispatch and passenger pickup, the total trip time for a passenger (includes waiting time post-dispatch and driving time) and the average driving time per trip (includes driving empty to a pickup, driving with a passenger to destination, and average rebalancing time).

To ensure there are sufficient SAVs available for rebalancing, the minimum fleet size is increased by 5% of that found using the minimization in Section III-G. A fleet size of $F = 2520$ SAVs was used for the symmetric scenario, and $F = 2880$ SAVs was used for the asymmetric demand scenario.

Figure 1 demonstrates how the number of rebalancing trips changes as the parameter k increases. In both the symmetric and asymmetric demand there is a maximum number of rebalancing trips based on the available vehicles. In both cases when k reaches about 5 this maximum is hit and the number of trips levels out to keep the network stable. This is the effect of the δ term in the rebalancing penalty. This demonstrates that regardless of the rebalancing policy, the network will remain stable if it was stable with no rebalancing.

When rebalancing vehicle are included in the simulation, vehicles are more likely to arrive at the pickup location prior to a passenger. This means a large portion of trips v_{qrs} will have a larger portion of $q = r$, reducing average times between vehicle dispatch and passenger pickup. On the other hand, when too many vehicles are rebalanced, they will all be moving and not available for dispatch, which is counter productive. As shown in Figure 2, there is a clear valley in dispatch-pickup time for the asymmetric demand before it starts to climb again. This means that when $k = 5$, waiting times for passengers will be minimized. On the other hand, the

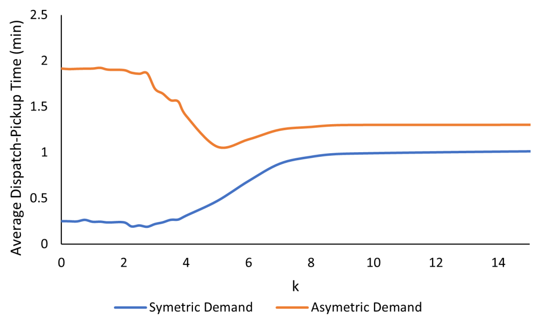

Fig. 2. Sioux Falls - Average time between vehicle dispatch and passenger pickup.

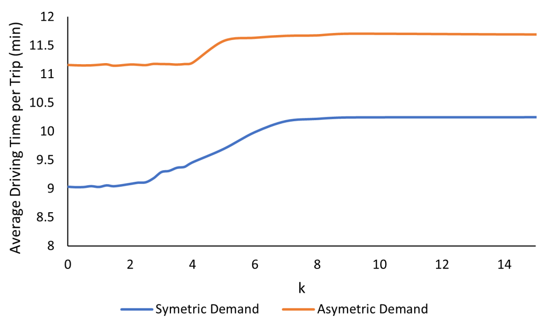

Fig. 3. Sioux Falls - Average driving time per passenger including driving to pickup, driving with passenger to destination, and rebalancing.

symmetric demand shows less of a positive impact due to the rebalancing. This is because symmetric demand already has a large portion of trips where the vehicle starting location q is already the same as the pickup location r (hence the much lower dispatch-pickup times). In this case, rebalancing is often counterproductive as the SAVs are moving, and not available for dispatch.

It is also important to consider the total time spent driving for each passenger trip and the average travel time for each passenger. These values are shown in Figures 3 and 4. We see in these figures that no rebalancing can reduce the driving time. However, there is a significant decrease in the travel time when rebalancing is high for the asymmetric demand scenario. As above, there is only slight improvement when demand is symmetric. Figures 3 and 4 demonstrate a trade-off between the additional driving time and the travel time reduction for passengers when rebalancing is included. This is because SAVs rebalance to locations that aren't exactly where they are needed so end up driving slightly more as a result. As k increases more, SAVs may even be rebalancing multiple times between passenger trips, further increasing time traveling empty. It is interesting to note that for the asymmetric demand $k = 4$ scenario, there is a significant decrease in travel time for passengers without much increase in overall driving for SAVs.

To help visualize the effects of demand symmetry on the rebalancing policy, a graph of the network is shown with the net rebalancing vehicles displayed at each node for each demand scenario (positive means vehicles are rebalancing to this node and negative means vehicles are leaving this node). As Figures 5 and 6 show, the demand pattern has a large influence on where rebalancing vehicles are moving

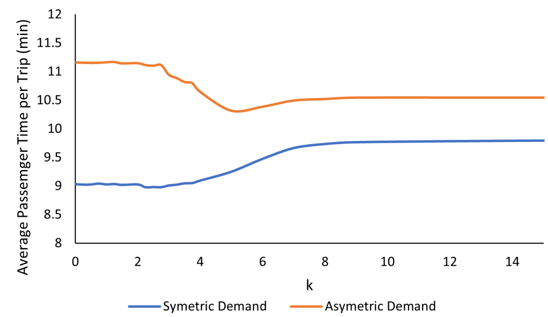

Fig. 4. Sioux Falls - Average time for passenger to make a trip including waiting and driving time.

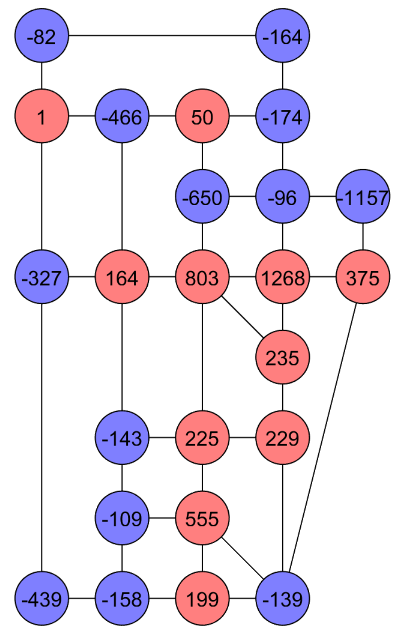

Fig. 5. Sioux Falls - Net number of vehicles rebalancing to each node in the symmetric demand scenario (k = 5, F = 2520).

throughout the network and the magnitude of the flows. In addition, even though the k value and total demand for these two scenarios was the same, the net rebalancing is much higher in the asymmetric demand scenario. This demonstration highlights the need for a policy that is adaptable to different demand patterns as they can change across days, weeks, and years for a real network.

B. Impact of Fleet Size on Performance

To dig deeper into the results for the asymmetric demand scenario on the Sioux Falls Network, the same metrics are used to examine the impacts of increasing fleet size. The fleet size was varied between 2520 SAVs and 3072 SAVs to see how rebalancing changed. Also shown are 5 different values of the parameter k to demonstrate the relationship with fleet size. The metrics calculated here are only for passengers arriving at their destination, so wait time are even longer for passengers when the fleet size is below 2800 vehicles when the network is no longer stable.

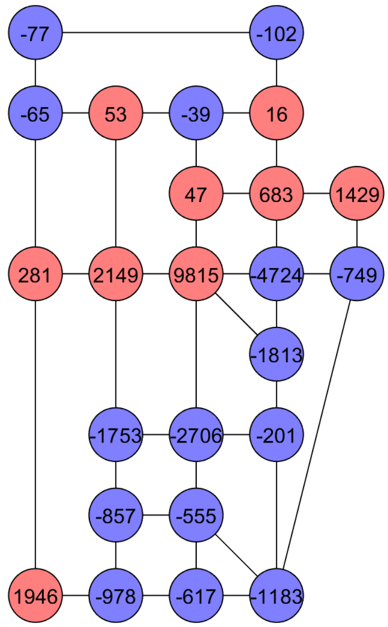

Fig. 6. Sioux Falls - Net number of vehicles rebalancing to each node in the asymmetric demand scenario (k = 5, F = 2880).

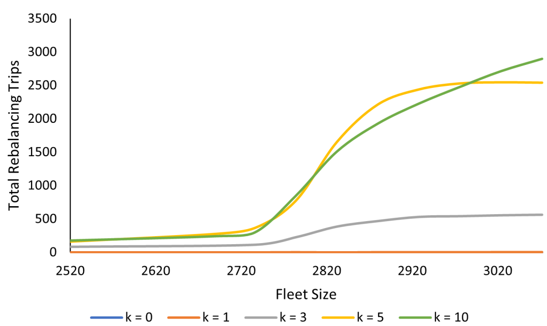

Fig. 7. Sioux Falls - Total rebalancing trips throughout the simulation.

Similar to the figures shown above, Figure 7 shows that as k increases, the number of rebalancing trips increases. However, it is also important that there be sufficient vehicles for rebalancing, so additional fleet size also vastly increase the number of rebalancing trips. This suggests that as demand changes throughout the day k should be modified to match those changes.

The fleet size also has a very big difference in the dispatch-pickup time for passengers. The smallest fleet size that stabilizes this demand is 2688, and this fleet size allows for almost no rebalancing. This means that even large values of k do not reduce the dispatch-pickup time significantly. However, as the fleet size increases, there is sufficient demand to rebalance some vehicles while still serving all passengers.

Finally, Figures 9 and 10 demonstrate the relationship between the time it takes a vehicle to serve a passenger and the time it takes a passenger to make a trip. Based on these figures $k = 10$ is demonstrated to not be the best policy; it has the most driving time for vehicles but does not improve the passenger time over the $k = 5$ value. Again, at low values of k there is a trade-off between the travel time reduction for passengers and the extra time that vehicles spend driving empty. We also note that without any rebalancing, the time to serve passengers does not increase even as the fleet size increases. However, when rebalancing is included the increases in fleet size benefits the passengers to a much greater extent.

Finally, we note that increasing rebalancing does not always reduce passenger time. This is because we do not dispatch vehicles that are enroute on a rebalancing (or passenger pickup) trip; they have to finish their trip before they can be rebalanced. Therefore, higher k values result in longer waiting times for vehicles to finish their rebalancing trips before they can be dispatched to pick up passengers.

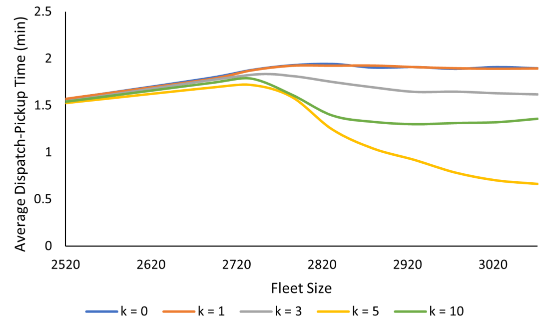

Fig. 8. Sioux Falls - Average time between vehicle dispatch and passenger pickup.

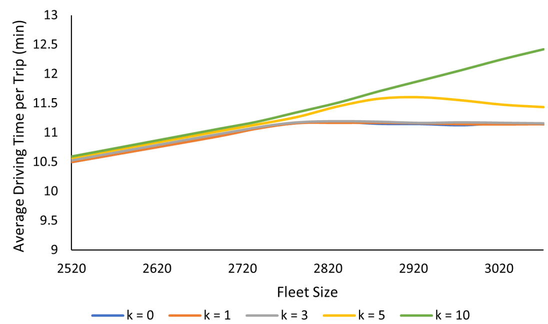

Fig. 9. Sioux Falls - Average driving time per passenger including driving to pickup, driving with passenger to destination, and rebalancing.

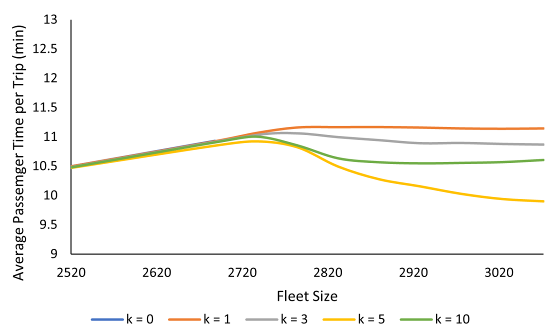

Fig. 10. Sioux Falls - Average time for passenger to make a trip including waiting and driving time.

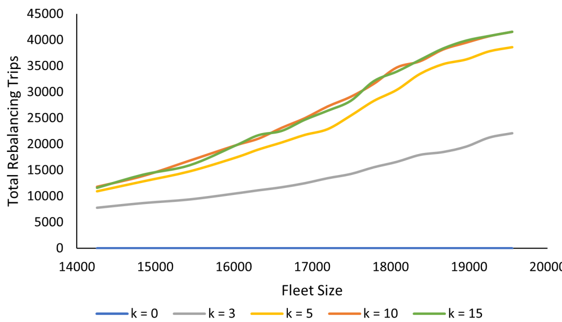

Fig. 11. Winnipeg - Total rebalancing trips throughout the simulation.

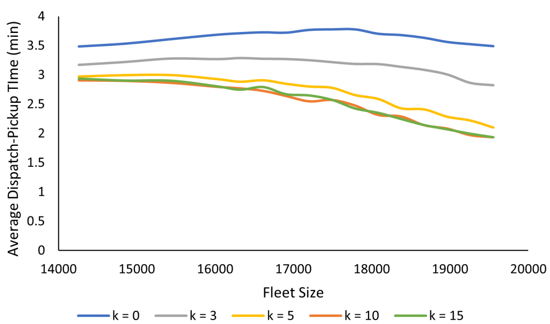

Fig. 12. Winnipeg - Average time between vehicle dispatch and passenger pickup.

C. Realistic Traffic Scenario - Winnipeg Network

The same performance metrics can be evaluated for the Winnipeg network using a realistic demand scenario. These results show very similar trends as those for the Sioux Falls Network. This network has longer average trip lengths and much higher demand, so the fleet size is correspondingly higher.

Figure 11 shows that k values greater than 10 stop affecting the number of vehicles rebalancing (as all vehicles that can rebalance are already assigned to rebalance). Taken to an extreme, we confirm that for a fleet size of 17,000 and a k value of 100, there are still about 25,000 rebalancing trips and the driving time for trip is still approximately 16 minutes.

As in the case of Sioux Falls, adding rebalancing can reduce the time between dispatch and pickup. Since the demand in this network is more symmetric, there is always a reduction in dispatch-pickup times as k increases for this specific network and demand scenario. For low fleet sizes the time can be reduced by 30 seconds, but that drops further as fleet size grows and more vehicles are available for rebalancing trips. Based on this reduction, the results in Figures 13 and 14 are intuitive; there is a decrease in travel times for passengers but an increase in driving time for the vehicles. As with the Sioux Falls network, we note a trade off between empty driving and reduced travel times for passengers as more rebalancing trips are added. This trade-off has been shown in multiple other simulation studies where waiting times for passengers are reduced at the expense of more vehicle miles traveled [5], [4], [56].

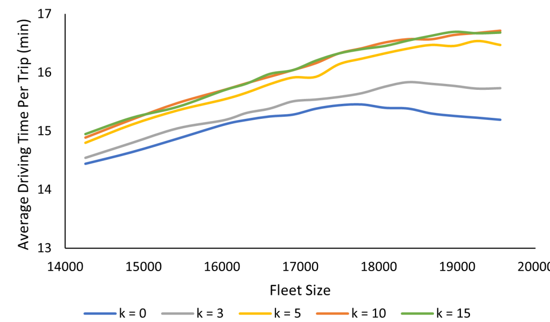

Fig. 13. Winnipeg - Average driving time per passenger including driving to pickup, driving with passenger to destination, and rebalancing.

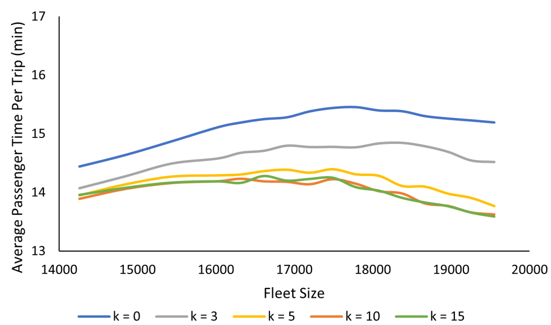

Fig. 14. Winnipeg - Average time for passenger to make a trip including waiting and driving time.

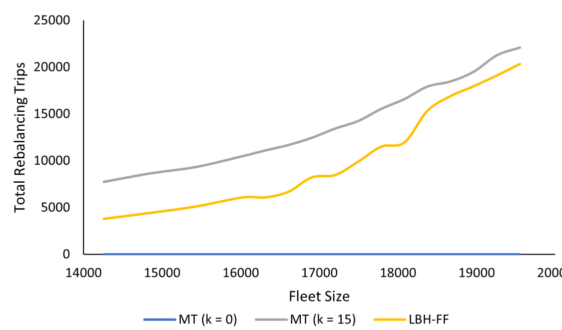

Fig. 15. Winnipeg - Total rebalancing trips throughout the simulation.

D. Comparison With Literature

Finally, the dispatch policy developed in this study is compared with a policy in the literature. The chosen policy (load balancing heuristic) assigns all requests to the closest vehicle if there are more available vehicles than requests or all vehicles to the closest request if there are more requests. Then, the optimal rebalancing flow are computed using a feedforward fluidic optimal rebalancing policy. These policies were proposed in Bischoff and Maciejewski [57] and Pavone et al. [37] respectively and have been studied in more depth and compared to additional literature in Hörl et al. [43] and Ruch et al. [58].

In Figures 15, 16, 17 and 18, the same metrics are plotted as above. These plots show the maximum throughput policy (MT) with $k = 0$ and $k = 15$, as well as the results from the load balancing heuristic with feedforward fluidic

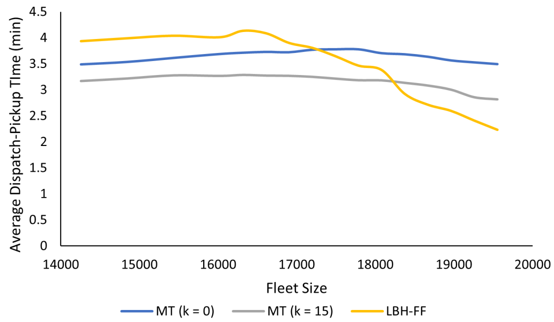

Fig. 16. Winnipeg - Average time between vehicle dispatch and passenger pickup.

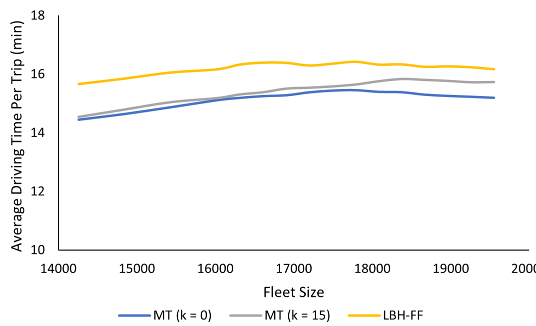

Fig. 17. Winnipeg - Average driving time per passenger including driving to pickup, driving with passenger to destination, and rebalancing.

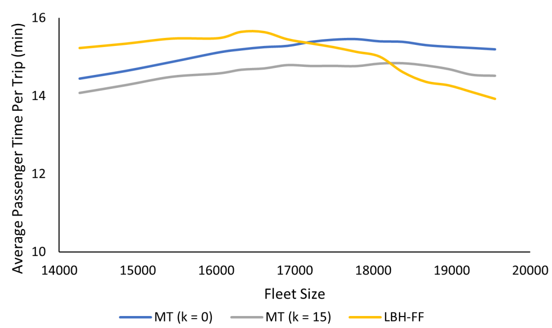

Fig. 18. Winnipeg - Average time for passenger to make a trip including waiting and driving time.

rebalancing (LBH-FF) for the Winnipeg network. When it comes to the rebalancing vehicles, the calculated optimal number of rebalancing trips by the LBH-FF policy is relatively high. In addition, even though there are fewer rebalancing trips than the MT (k=15) policy, they tend to be longer as evidenced by the higher driving times per trip for all fleet sizes.

It is interesting to note that at a higher fleet size the LBH-FF policy performs better than either of the other two in terms of time for passengers. However, this is not the case at low fleet sizes. The issues with the LBH-FF at low fleet sizes are compounded since the calculated statistics only consider vehicles that have completed their trips, and in some scenarios there are a significant number of passengers who have yet to be assigned a vehicle. To highlight this issue we plot the number of passengers that have not been assigned a vehicle as a function of time in Figure 19 for a fleet size of 17500 vehicles.

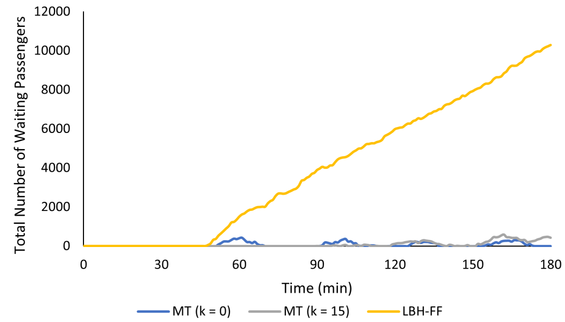

Fig. 19. Winnipeg - Number of waiting passengers throughout a single simulation (F = 17500).

The network is not stable when the LBH-FF policy is used since the queues of passengers are shown to increase over time. Though there are time periods when queues form for the MT policy these queue eventually dissipate and the policy remains stable.

Though the proof of maximum throughput does not make any guarantees on travel times or empty driving amounts, there are some clear benefits. In particular, the total time for a passenger to be served is the time for a vehicle to be dispatched, the dispatch-pickup time, and then the driving time. However, with the LBH-FF policy in the scenario above, as the queue grow very large waiting times or a vehicle to be dispatched also grow. This causes total travel times to be much larger even when passenger travel times from dispatch to drop-off are very similar. This example highlights the importance of maintaining stability wherever possible in the dispatch decision.

V. Conclusion

This paper builds on the work of Xu et al. [7] to develop a maximum stability dispatch policy for SAVs including rebalancing. We prove the stability of this dispatch policy and show that it can serve as much demand as any other policy. This is an important improvement on previous research on rebalancing because there is a mathematical guarantee that this policy is throughput optimal. This is particularly valuable since rebalancing vehicles take away fleet capacity from the total vehicles service passenger trips each timestep but in this policy we prove that the stable region is not affected. This policy is also an improvement on previous work on stable dispatch since rebalancing vehicles reduce passenger waiting times. We also demonstrate that the minimum fleet size for vehicles using this policy can be determined exactly for a given level of demand.

The simulation model demonstrates the superior performance of this policy compared one where rebalancing is not allowed. We demonstrate a trade-off between the number of rebalancing trips and the travel time benefits to passengers. In particular, this dispatch policy can reduce waiting times for passengers between dispatch and pickup when demand is asymmetric and the fleet size is greater than the minimum. Future simulation work should examine additional dispatch policies to find ways to further reduce travel times for passen-

TABLE I
LIST OF NOTATION

Notation	Meaning
$\mathcal{N}$	List of Nodes.
$\mathcal{A}$	List of Links.
C_{qrs}	Travel time from node q to r to s.
C_{qs}	Travel time from node q directly to s.
$\bar{C}_{rs}$	Average time to serve a passenger going from node q to s.
$d_{rs}(t)$	Demand from node r to s at time t.
$\bar{d}_{rs}$	Average demand from node r to s.
$\bar{\boldsymbol{d}}$	Vector of average demand.
$\mathcal{D}$	Stable demand region
$\mathcal{D}^0$	Interior of stable demand region
$w_{rs}(t)$	Number of travelers waiting at node r to travel to s at time t.
$v_{qrs}(t)$	SAVs at node q assigned to take a traveler from r to s at time t.
$\bar{v}_{qrs}$	Average number of SAVs at node q assigned to take a traveler from r to s.
$v^\star_{qrs}(t)$	Optimal number of SAVs at node q assigned to take a traveler from r to s at time t.
$\bar{\boldsymbol{v}}$	Vector of average SAV dispatch decisions.
$e_{qs}(t)$	SAVs at node q assigned to rebalance to node s at time t.
$\bar{e}_{qs}$	Average number of SAVs at node q assigned to rebalance to node s.
$e^\star_{qs}(t)$	Optimal number of SAVs at node q assigned to rebalance to node s at time t.
$\bar{\boldsymbol{e}}$	Vector of average SAV rebalancing decisions.
$x^\tau_q(t)$	Number of vehicles τ timesteps from node q at time t.
$x_q(t)$	Number of vehicles parked at q at time t.
F	Fleet Size.
$\pi^\star$	Proposed dispatch policy.
$\delta(t)$	Binary variable indicating whether available vehicles exceeds waiting customers.
$a_s(t)$	Penalty function for a large imbalance of vehicles.
$z_s(t)$	Optimal number of vehicles in each zone.
k	Scalar parameter to control the amount of rebalancing.

gers while limiting empty traveling. In addition, as this policy is centralized, a comparison with a decentralized controller or multiple competing control strategies would be of interest.

There are many extensions to this maximum stability dispatch policy to be considered in the future. This study assumes constant travel times on links, so future work should examine the impacts of congestion. We also assumed that each SAV only had capacity for one passenger, which eliminated the complication of ridesharing. In addition, this formulation assumed that vehicles could not reroute once dispatched. Relaxing these constraints would allow for better performance at low fleet sizes and as the number of rebalancing trips increase, but stability would need to be proved under this new set of assumptions. Finally, more study should be done on the rebalancing policy. We assumed that the optimal number of SAVs in each zone was known, which may not always be true in practice. This assumption can be relaxed for future modifications to the rebalancing penalty function. This study proved that a class of rebalancing policies are stable, but only tested one example in simulation. There are many other rebalancing strategies fitting the requirements in Section III-F which might perform better in certain scenarios.

APPENDIX A
NOTATION

See Table I.

APPENDIX B
PROOF OF PROPOSITION 1

Proof: If the number if waiting customers is low, i.e.

$$\sum_{rs\in\mathcal{N}^2} w_{rs}(t) \le \sum_{q\in\mathcal{N}} x_q(t) \tag{26}$$

Then, we know based on equation (7d):

$$\sum_{q\in\mathcal{N}} v_{qrs}(t) = \sum_{rs\in\mathcal{N}} w_{rs}(t) \tag{27}$$

This means $\sum_{rs\in\mathcal{N}} w_{rs}(t)$ is bounded by the fleet size F as defined in (5). There will also be rebalancing trips in the range $0 \le \sum_{qs\in\mathcal{N}} e_{qs}(t) \le F - \sum_{qrs\in\mathcal{N}} v_{qrs}(t)$.

However, when the number of waiting passengers is high:

$$\sum_{rs\in\mathcal{N}^2} w_{rs}(t) \ge \sum_{q\in\mathcal{N}} x_q(t) \tag{28}$$

Then, we know based on equation (7d):

$$\sum_{q\in\mathcal{N}} v_{qrs}(t) = \sum_{q\in\mathcal{N}} x_q(t) \tag{29}$$

This means there are no vehicles left to assign to rebalancing trips. ∎

APPENDIX C
PROOF OF PROPOSITION 2

Proof: When rebalancing trips are included, demand is served by vehicles which first rebalance and then serve passengers. However, this rebalancing behavior (described by $\bar{e}_{qs}$) can be replaced with the with the rebalancing inherent in $\bar{v}_{qrs}$.

Define an intermediate node i. A rebalancing trip from q to i via $\bar{e}_{qi}$ then a trip from i to r to s via $\bar{v}_{irs}$ is equivalent to $\bar{e}_{qi} = 0$ with the trip described by $\bar{v}_{qrs}$. Thus, by setting $\bar{e}_{qi} = 0$, there are still the same number of trips from q to r to s.

Using this logic, equation (10) can rewritten as:

$$\sum_{(q,i)\in\mathcal{N}^2} \bar{e}_{qi} + \sum_{i\in\mathcal{N}} \bar{v}_{irs} \tag{30}$$

$$= \sum_{q\in\mathcal{N}} \bar{v}_{qrs} \quad \forall (r,s) \in \mathcal{N}^2 \tag{31}$$

$$= \bar{d}_{rs} \quad \forall (r,s) \in \mathcal{N}^2 \tag{32}$$

By assumption, $C_{qi} + C_{irs} \ge c_{qrs}$ so that equation (15) is still satisfied. ∎

APPENDIX D
PROOF OF PROPOSITION 3

Proof:

When $\bar{\boldsymbol{d}} \notin \mathcal{D}$, equation (10) must be rewritten to account for some additional demand $\delta \ge 0$:

$$\sum_{q\in\mathcal{N}} \bar{v}_{qrs} + \delta \le \bar{d}_{rs} \qquad \exists (r,s) \in \mathcal{N}^2 \tag{33}$$

We can then return to equation (1) to track the progression of waiting passengers between timesteps:

$$w_{rs}(t+1) - w_{rs}(t) = d_{rs}(t) - \sum_{q\in\mathcal{N}} v_{qrs}(t) \tag{34}$$

The expected value of this formulation summed from $t = 0$ to $t = \tau - 1$ and all locations can be rewritten as:

$$\mathbb{E}\left[\sum_{t=0}^{\tau-1} \sum_{(r,s)\in\mathcal{N}^2} (w_{rs}(t+1) - w_{rs}(t))\right] \tag{35}$$

$$= \mathbb{E}\left[\sum_{(r,s)\in\mathcal{N}^2} (w_{rs}(\tau) - w_{rs}(0))\right] \tag{36}$$

$$= \mathbb{E}\left[\sum_{(r,s)\in\mathcal{N}^2} (\bar{d}_{rs} - \sum_{q\in\mathcal{N}} \bar{v}_{qrs})\right] \tag{37}$$

However, equation 33 gives us:

$$\mathbb{E}\left[\sum_{(r,s)\in\mathcal{N}^2} (\bar{d}_{rs} - \sum_{q\in\mathcal{N}} \bar{v}_{qrs})\right] \geq \tau\delta \tag{38}$$

Combining equations (37) and (38) gives us a relationship between the expected value of the state at time τ and the state at time 0.

$$\mathbb{E}[\boldsymbol{w}(\tau)] \geq \mathbb{E}[\boldsymbol{w}(0)] + \tau\delta \tag{39}$$

Taking the limit as $T \to \infty$ yields:

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\boldsymbol{w}(\tau)] \geq \lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} [\mathbb{E}[\boldsymbol{w}(0)] + \tau\delta] = \infty \tag{40}$$

Equation (40) shows that with each additional timestep $w_{rs}(t)$ will increase by δ. As t grows, this will cause the number of waiting passengers to grow to infinity. This violates the constraint in equation (6), meaning the network will be unstable.

■

Appendix E
Proof of Lemma 1

Proof: We utilize the same Lyapunov function defined in Xu et al. [7]:

$$\nu(t) = \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs} w_{rs}(t)}{2}\right) + \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty} \tau x_s^\tau(t)\right) \tag{41}$$

Set $\nu(t) = \nu_1(t) + \nu_2(t)$ where $\nu_1(t)$ and $\nu_2(t)$ are defined as:

$$\nu_1(t) = \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs} w_{rs}(t)}{2}\right) \tag{42}$$

$$\nu_2(t) = \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty} \tau x_s^\tau(t)\right) \tag{43}$$

To examine the difference $\nu(t+1) - \nu(t)$ we begin with $\nu_1(t+1) - \nu_1(t)$. For simplicity, define $\delta_{rs}(t) = w_{rs}(t+1) - w_{rs}(t)$:

$$\nu_1(t+1) - \nu_1(t)$$
$$= \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t+1)\right) \times \left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs} w_{rs}(t+1)}{2}\right) - \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs} w_{rs}(t)}{2}\right) \tag{44}$$

$$= \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) + \delta_{rs}(t)\right) \times \left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs}}{2}(w_{rs}(t) + \delta_{rs}(t))\right) - \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs} w_{rs}(t)}{2}\right) \tag{45}$$

$$= \left(\sum_{(r,s)\in\mathcal{N}^2} \delta_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs}\delta_{rs}(t)}{2}\right) + \left(\sum_{(r,s)\in\mathcal{N}^2} \bar{C}_{rs}\delta(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right) \tag{46}$$

$\left(\sum_{(r,s)\in\mathcal{N}^2} \delta_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs}\delta_{rs}(t)}{2}\right)$ is bounded as shown (47) since the difference in w_{rs} between timesteps is bounded by the incremental increase in demand $\hat{d}_{rs}$. This term will become part of the κ term in (21). The second term will be combined with the $\nu_2(t)$ term.

$$\left(\sum_{(r,s)\in\mathcal{N}^2} \delta_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs}\delta_{rs}(t)}{2}\right) \leq \left(\sum_{(r,s)\in\mathcal{N}^2} \hat{d}_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2} \frac{\bar{C}_{rs}\hat{d}_{rs}(t)}{2}\right) \tag{47}$$

Now we continue working with $\nu_2(t+1) - \nu_2(t)$.

$$\nu_2(t+1)\nu_2(t)$$
$$= \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t+1)\right) \times \left(\sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty} \tau x_s^\tau(t+1)\right) - \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty} \tau x_s^\tau(t)\right) \tag{48}$$

$$= \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) + \delta_{rs}(t)\right) \times \left(\sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty} \tau x_s^\tau(t+1)\right) - \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty} \tau x_s^\tau(t)\right) \tag{49}$$

$$= \delta(t)\left(\sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty}\tau x_s^{\tau}(t+1)\right) + \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right) \times \left(\sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty}\tau\left(x_s^{\tau}(t+1)-x_s^{\tau}(t)\right)\right) \quad (50)$$

The first term is bounded because we know $\sum_{\tau=1}^{\infty} x_s^{\tau}(t+1) \leq F$, so:

$$\delta(t)\left(\sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty}\tau x_s^{\tau}(t+1)\right) \leq \left(\sum_{(r,s)\in\mathcal{N}^2}\hat{d}_{rs}\right)\left(F \times \max_{(q,r,s)\in\mathcal{N}^3}\{C_{qrs}\}\right) \quad (51)$$

Combining the remaining terms from $\nu_1(t)$ and $\nu_2(t)$ (the second terms from equations (46) and (50)). We are now left with the term:

$$\left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right) \times \left(\sum_{(r,s)\in\mathcal{N}^2}\bar{C}_{rs}\delta_{rs}(t) + \sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty}\tau\left(x_s^{\tau}(t+1)-x_s^{\tau}(t)\right)\right) \quad (52)$$

Based on equation (2):

$$\sum_{\tau=1}^{\infty}\tau x_s^{\tau}(t+1) - \sum_{\tau=1}^{\infty}\tau x_s^{\tau}(t) = \sum_{\tau=1}^{\infty}(\tau-1)x_s^{\tau-1}(t+1) - \sum_{\tau=1}^{\infty}\tau x_s^{\tau}(t) \quad (53)$$

$$= \sum_{\tau=1}^{\infty}(\tau-1)x_s^{\tau}(t) - \sum_{\tau=1}^{\infty}\tau x_s^{\tau}(t) + \sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}(t)C_{qs} \quad (54)$$

$$= -\sum_{\tau=0}^{\infty}x_s^{\tau}(t) + \sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}(t)C_{qs} \quad (55)$$

$$= -F + \sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}(t)C_{qs} \quad (56)$$

Based on equation (34):

$$\bar{C}_{rs}\delta_{rs}(t) = \bar{C}_{rs}\left(d_{rs}(t) - \sum_{q\in\mathcal{N}} v_{qrs}(t)\right) \quad (57)$$

From equations (56) and (57), equation (52) becomes:

$$\mathbb{E}\Bigg[\left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2}\bar{C}_{rs}\delta_{rs}(t) + \sum_{s\in\mathcal{N}}\sum_{\tau=1}^{\infty}\tau\left(x_s^{\tau}(t+1)-x_s^{\tau}(t)\right)\right)\Bigg|\boldsymbol{w}(t),\boldsymbol{x}(t)\Bigg] \quad (58)$$

$$= \mathbb{E}\Bigg[\left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{(r,s)\in\mathcal{N}^2}\bar{C}_{rs}\left(d_{rs}(t) - \sum_{q\in\mathcal{N}} v_{qrs}(t)\right) - F + \sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}(t)C_{qs}\right)\Bigg|\boldsymbol{w}(t),\boldsymbol{x}(t)\Bigg] \quad (59)$$

Recall, $v_{qrs}^{\star}(t)$, $e_{qs}^{\star}(t)$, and $a_s(t)$ be chosen by $\pi^{\star}$, where $a_s(t)$ is a function of $e_{qs}^{\star}(t)$ and $v_{qrs}^{\star}(t)$. Then we have:

$$\mathbb{E}\Bigg[\left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}^{\star}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}^{\star}C_{qs} + \sum_{s\in\mathcal{N}} a_s\right)\Bigg|\boldsymbol{w}(t),\boldsymbol{x}(t)\Bigg] \quad (60)$$

We know, $\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) \times \sum_{s\in\mathcal{N}} a_s^{\star}$ is bounded. This is because the term is bounded by at most the fleet size when the number of waiting passengers is smaller than the available vehicles. In addition, the parameter δ will remove a_s unless $\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) \leq \sum_{q\in\mathcal{N}} x_q(t) \leq F$. Together, this causes a_s to be bounded in all scenarios. This term can become part of κ, leaving:

By this logic, we can show that:

$$\mathbb{E}\Bigg[\left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}^{\star}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}^{\star}C_{qs}\right)\Bigg|\boldsymbol{w}(t),\boldsymbol{x}(t)\Bigg] \quad (61)$$

$$\leq \mathbb{E}\Bigg[\left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(\sum_{(q,r,s)\in\mathcal{N}^3} v_{qrs}(t)C_{qrs} + \sum_{(q,s)\in\mathcal{N}^2} e_{qs}C_{qs}\right)\Bigg|\boldsymbol{w}(t),\boldsymbol{x}(t)\Bigg] \quad (62)$$

We note that the terms of equation 62 show up in equation 59. We will show below how these terms simplify out.

Since policy $\pi^{\star}$ is a minimization, $\sum_{(q,r,s)\in\mathcal{N}^2}\bar{C}_{rs}v_{qrs}(t) \geq \sum_{(q,r,s)\in\mathcal{N}^3} C_{qrs}v_{qrs}(t)$. This means these two terms are bounded and drop out of equation (59). The term $\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) \times \sum_{(q,s)\in\mathcal{N}^2} e_{qs}(t)C_{qs}$ is also bounded by Proposition 2. This leaves:

$$\mathbb{E}\left[\left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t)\right)\left(-F + \sum_{(r,s)\in\mathcal{N}^2}\bar{C}_{rs}d_{rs}(t)\right)\Bigg|\boldsymbol{w}(t),\boldsymbol{x}(t)\right] \quad (63)$$

$$= \left(\sum_{(r,s)\in\mathcal{N}^2} w_{rs}(t) \right) \left(-F + \sum_{(r,s)\in\mathcal{N}^2} \bar{C}_{rs}\bar{d}_{rs}(t) \right) \tag{64}$$

$$\leq -\epsilon |\boldsymbol{w}(t)| \tag{65}$$

since $F > \sum_{(r,s)\in\mathcal{N}^2} \bar{C}_{rs}\bar{d}_{rs}$ by equation (19).

Equation (21) follows from equations (59), (62) and (65). ■

APPENDIX F
PROOF OF PROPOSITION 4

Proof: Based on Lemma 1 summed from $t = 1$ to $t = T$:

$$\mathbb{E}\left[\, \nu(\boldsymbol{w}(T+1)) - \nu(\boldsymbol{w}(1)) \,|\, \boldsymbol{w}(t), \boldsymbol{x}(t)\right] \leq \kappa T - \epsilon \sum_{(t=1)}^{T} |\boldsymbol{w}(t)| \tag{66}$$

Equation (66) can be simplified to:

$$\epsilon \frac{1}{T} \sum_{(t=1)}^{T} \mathbb{E}\left[\boldsymbol{w}(t)\right] \leq \kappa - \frac{1}{T}\mathbb{E}\left[\nu(\boldsymbol{w}(T+1))\right] + \frac{1}{T}\mathbb{E}\left[\nu(\boldsymbol{w}(1))\right] \leq \kappa + \frac{1}{T}\mathbb{E}\left[\nu(\boldsymbol{w}(1))\right] \tag{67}$$

Equation (67) is equivalent to the definition of stability in Definition 1. ■

REFERENCES

[1] D. Kang and M. W. Levin, "Maximum-stability dispatch policy for shared autonomous vehicles," *Transp. Res. B, Methodol.*, vol. 148, pp. 132–151, Jun. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261521000746

[2] K. M. Gurumurthy, K. M. Kockelman, and M. D. Simoni, "Benefits and costs of ride-sharing in shared automated vehicles across Austin, Texas: Opportunities for congestion pricing," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2673, no. 6, pp. 548–556, Jun. 2019, doi: 10.1177/0361198119850785.

[3] M. W. Levin, K. M. Kockelman, S. D. Boyles, and T. Li, "A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application," *Comput., Environ. Urban Syst.*, vol. 64, pp. 373–383, Jul. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S019897151630237X

[4] D. J. Fagnant and K. M. Kockelman, "The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios," *Transp. Res. C, Emerg. Technol.*, vol. 40, pp. 1–13, Mar. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X13002581

[5] S. Narayanan, E. Chaniotakis, and C. Antoniou, "Shared autonomous vehicle services: A comprehensive review," *Transp. Res. C, Emerg. Technol.*, vol. 111, pp. 255–293, Feb. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X19303493

[6] L. Li, T. Pantelidis, J. Y. J. Chow, and S. E. Jabari, "A real-time dispatching strategy for shared automated electric vehicles with performance guarantees," 2020, *arXiv:2006.15615*.

[7] T. Xu, M. W. Levin, and M. Cieniawski, "A zone-based dynamic queueing model and maximum-stability dispatch policy for shared autonomous vehicles," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 3827–3832.

[8] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Controll.*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[9] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 177–195, Nov. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X13001782

[10] T. Wongpiromsarn, T. Uthaicharoenpong, Y. Wang, E. Frazzoli, and D. Wang, "Distributed traffic signal control for maximum network throughput," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 588–595.

[11] J. Wu, D. Ghosal, M. Zhang, and C.-N. Chuah, "Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 896–909, Feb. 2018. [Online]. Available: https://escholarship.org/uc/item/1qm8390f

[12] P. Mercader, W. Uwayid, and J. Haddad, "Max-pressure traffic controller based on travel times: An experimental analysis," *Transp. Res. C, Emerg. Technol.*, vol. 110, pp. 275–290, Jan. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X19307442

[13] N. Xiao, E. Frazzoli, Y. Li, Y. Wang, and D. Wang, "Pressure releasing policy in traffic signal control with finite queue capacities," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 6492–6497.

[14] J. Gregoire, X. Qian, E. Frazzoli, A. D. L. Fortelle, and T. Wongpiromsarn, "Capacity-aware backpressure traffic signal control," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 164–173, Jun. 2015.

[15] A. Kouvelas, J. Lioris, S. Fayazi, and P. Varaiya, "Maximum pressure controller for stabilizing queues in signalized arterial networks," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2421, no. 1, pp. 133–141, 2014. [Online]. Available: http://journals.sagepub.com/doi/10.3141/2421-15

[16] T. Le, P. Kovács, N. Walton, H. Vu, L. H. Andrew, and S. P. Hoogendoorn, "Decentralized signal control for urban road networks," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 431–450, Sep. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X14003325

[17] M. W. Levin, J. Hu, and M. Odell, "Max-pressure signal control with cyclical phase structure," *Transp. Res. C, Emerg. Technol.*, vol. 120, Nov. 2020, Art. no. 102828. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20307324

[18] X. Sun and Y. Yin, "A simulation study on max pressure control of signalized intersections," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 18, pp. 117–127, 2018. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0361198118786840

[19] J. Robbennolt, R. Chen, and M. Levin, "Microsimulation study evaluating the benefits of cyclic and non-cyclic max-pressure control of signalized intersections," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2676, no. 12, pp. 303–317, Dec. 2022, doi: 10.1177/03611981221095520.

[20] H. Taale, J. van Kampen, and S. Hoogendoorn, "Integrated signal control and route guidance based on back-pressure principles," *Transp. Res. Proc.*, vol. 10, pp. 226–235, Jan. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352146515002598

[21] A. A. Zaidi, B. Kulcsár, and H. Wymeersch, "Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2134–2143, Aug. 2016.

[22] T. Le, H. L. Vu, N. Walton, S. P. Hoogendoorn, P. Kovács, and R. N. Queija, "Utility optimization framework for a distributed traffic control of urban road networks," *Transp. Res. B, Methodol.*, vol. 105, pp. 539–558, Nov. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261516303022

[23] H. Chai, H. M. Zhang, D. Ghosal, and C.-N. Chuah, "Dynamic traffic routing in a network with adaptive signal control," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 64–85, Dec. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X17302279

[24] R. Chen, J. Hu, M. W. Levin, and D. Rey, "Stability-based analysis of autonomous intersection management with pedestrians," *Transp. Res. C, Emerg. Technol.*, vol. 114, pp. 463–483, May 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X19309623

[25] M. W. Levin, D. Rey, and A. Schwartz, "Max-pressure control of dynamic lane reversal and autonomous intersection management," *Transportmetrica B, Transp. Dyn.*, vol. 7, no. 1, pp. 1693–1718, Dec. 2019. [Online]. Available: https://par.nsf.gov/servlets/purl/10158013

[26] T. D. Chen, K. M. Kockelman, and J. P. Hanna, "Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions," *Transp. Res. A, Policy Pract.*, vol. 94, pp. 243–254, Dec. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S096585641630756X

[27] M. Hyland and H. S. Mahmassani, "Operational benefits and challenges of shared-ride automated mobility-on-demand services," *Transp. Res. A, Policy Pract.*, vol. 134, pp. 251–270, Apr. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0965856419307888

[28] D. J. Fagnant and K. M. Kockelman, "Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas," *Transportation*, vol. 45, no. 1, pp. 143–158, 2018, doi: 10.1007/s11116-016-9729-z.

[29] J. Liu, K. M. Kockelman, P. M. Boesch, and F. Ciari, "Tracking a system of shared autonomous vehicles across the Austin, Texas network using agent-based simulation," *Transportation*, vol. 44, no. 6, pp. 1261–1278, Nov. 2017, doi: 10.1007/s11116-017-9811-1.

[30] W. Zhang, S. Guhathakurta, J. Fang, and G. Zhang, "Exploring the impact of shared autonomous vehicles on urban parking demand: An agent-based simulation approach," *Sustain. Cities Soc.*, vol. 19, pp. 34–45, Dec. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221067071530010X

[31] K. T. Seow, N. H. Dang, and D.-H. Lee, "A collaborative multiagent taxi-dispatch system," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 607–616, Jul. 2010.

[32] M. Maciejewski and K. Nagel, "Simulation and dynamic optimization of taxi services in MATSim," *Transp. Sci.*, 2013.

[33] Q. Ge, K. Han, and X. Liu, "Matching and routing for shared autonomous vehicles in congestible network," *Transp. Res. E, Logistics Transp. Rev.*, vol. 156, Dec. 2021, Art. no. 102513. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S136655452100274X

[34] M. Hyland and H. S. Mahmassani, "Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests," *Transp. Res. C, Emerg. Technol.*, vol. 92, pp. 278–297, Jul. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X18306028

[35] K. M. Gurumurthy, K. M. Kockelman, and B. J. Loeb, "Chapter four—Sharing vehicles and sharing rides in real-time: Opportunities for self-driving fleets," in *Advances in Transport Policy and Planning*, vol. 4, E. Fishman, Ed. Cambridge, MA, USA: Academic Press, Jan. 2019, pp. 59–85. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2543000919300344

[36] Z. Xu et al., "Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, pp. 905–913, doi: 10.1145/3219819.3219824.

[37] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for mobility-on-demand systems," *Int. J. Robot. Res.*, vol. 31, no. 7, pp. 839–854, 2012.

[38] R. Zhang and M. Pavone, "Control of robotic mobility-on-demand systems: A queueing-theoretical perspective," *Int. J. Robot. Res.*, vol. 35, nos. 1–3, pp. 186–203, Jan. 2016, doi: 10.1177/0278364915581863.

[39] J. Ma, X. Li, F. Zhou, and W. Hao, "Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach," *Transp. Res. C, Emerg. Technol.*, vol. 84, pp. 124–141, Nov. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X17302322

[40] J. Cramer and A. B. Krueger, "Disruptive change in the taxi business: The case of Uber," *Amer. Econ. Rev.*, vol. 106, no. 5, pp. 177–182, 2016, doi: 10.1257/aer.p20161002.

[41] A. Komanduri, Z. Wafa, K. Proussaloglou, and S. Jacobs, "Assessing the impact of app-based ride share systems in an urban context: Findings from Austin," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 7, pp. 34–46, Dec. 2018, doi: 10.1177/0361198118796025.

[42] M. W. Levin, H. Smith, and S. D. Boyles, "Dynamic four-step planning model of empty repositioning trips for personal autonomous vehicles," *J. Transp. Eng., A, Syst.*, vol. 145, no. 5, May 2019, Art. no. 04019015. [Online]. Available: https://ascelibrary.org/doi/full/10.1061/JTEPBS.0000235

[43] S. Hörl, C. Ruch, F. Becker, E. Frazzoli, and K. W. Axhausen, "Fleet operational policies for automated mobility: A simulation assessment for Zurich," *Transp. Res. C, Emerg. Technol.*, vol. 102, pp. 20–31, May 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X18304248

[44] F. Rossi, R. Zhang, Y. Hindy, and M. Pavone, "Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms," *Auton. Robots*, vol. 42, no. 7, pp. 1427–1442, 2018, doi: 10.1007/s10514-018-9750-5.

[45] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in Singapore," in *Road Vehicle Automation* (Lecture Notes in Mobility), G. Meyer and S. Beiker, Eds. Cham, Switzerland: Springer, 2014, pp. 229–245, doi: 10.1007/978-3-319-05990-7_20.

[46] S. Illgen and M. Hock, "Literature review of the vehicle relocation problem in one-way car sharing networks," *Transp. Res. B, Methodol.*, vol. 120, pp. 193–204, Feb. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0191261518300547

[47] E. Skordilis, Y. Hou, C. Tripp, M. Moniot, P. Graf, and D. Biagioni, "A modular and transferable reinforcement learning framework for the fleet rebalancing problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11903–11916, Aug. 2022.

[48] Z. Qin, H. Zhu, and J. Ye, "Reinforcement learning for ridesharing: An extended survey," 2021, *arXiv:2105.01099*.

[49] A. Castagna, M. Guériau, G. Vizzari, and I. Dusparic, "Demand-responsive rebalancing zone generation for reinforcement learning-based on-demand mobility," *AI Commun.*, vol. 34, pp. 73–88, Jan. 2021.

[50] Y. Gao, D. Jiang, and Y. Xu, "Optimize taxi driving strategies based on reinforcement learning," *Int. J. Geograph. Inf. Sci.*, vol. 32, no. 8, pp. 1677–1696, Aug. 2018, doi: 10.1080/13658816.2018.1458984.

[51] C. Fluri, C. Ruch, J. Zilly, J. Hakenberg, and E. Frazzoli, "Learning to operate a fleet of cars," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2292–2298.

[52] A. O. Al-Abbasi, A. Ghosh, and V. Aggarwal, "DeepPool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4714–4727, Dec. 2019.

[53] K. Lin, R. Zhao, Z. Xu, and J. Zhou, "Efficient large-scale fleet management via multi-agent deep reinforcement learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Jul. 2018, pp. 1774–1783, doi: 10.1145/3219819.3219993.

[54] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan, "Bounds on average delays and queue size averages and variances in input-queued cell-based switches," in *Proc. IEEE INFOCOM. Conf. Comput. Commun., 20th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Apr. 2001, pp. 1095–1103.

[55] (Jun. 2022). *Transportation Networks for Research Core Team*. Accessed: Jun. 20, 2022. [Online]. Available: https://github.com/bstabler/TransportationNetworks

[56] R. Vosooghi, J. Puchinger, M. Jankovic, and A. Vouillon, "Shared autonomous vehicle simulation and service design," *Transp. Res. C, Emerg. Technol.*, vol. 107, pp. 15–33, Oct. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X19304449

[57] J. Bischoff and M. Maciejewski, "Simulation of city-wide replacement of private cars with autonomous taxis in Berlin," *Proc. Comput. Sci.*, vol. 83, pp. 237–244, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050916301442

[58] C. Ruch, S. Horl, and E. Frazzoli, "AMoDeus, a simulation-based testbed for autonomous mobility-on-demand systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3639–3644.

Jake Robbennolt received the B.S. degree in civil engineering from the University of Minnesota in 2022. He is currently pursuing the master's degree with the Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin. His research interests include network modeling and intelligent transportation systems. He was a recipient of the 2023 Best TRB Freeway Operations Committee Young Professional/Student Paper Award.

Michael W. Levin received the B.S. degree in computer science and the Ph.D. degree in civil engineering from The University of Texas at Austin, Austin, TX, USA, in 2013 and 2017, respectively. He is currently an Assistant Professor with the Department of Civil, Environmental, and Geo-Engineering, University of Minnesota. His research interests include traffic flow and network modeling of connected autonomous vehicles and intelligent transportation systems. He is a member of the Network Modeling Committee (ADB30) of the Transportation Research Board. His work has received several awards, including the 2019 Ryuichi Kitamura Award and the 2016 Milton Pikarsky Award from the Council of University Transportation Centers. He is on the editorial board of *Transportation Research Part B: Methodological*.