

A Self-Organizing Policy for Vehicle Dispatching in Public Transit Systems with Multiple Lines

Rolf N. van Lieshout^{1,2}, Paul C. Bouman¹, Marjan van den Akker³, and Dennis Huisman^{1,4}

¹Econometric Institute and ECOPT, Erasmus University Rotterdam, 3000 DR Rotterdam, The Netherlands

²Corresponding author. Email address: vanlieshout@ese.eur.nl

³Department of Information and Computing Sciences, Utrecht University, 3584 CC Utrecht, The Netherlands

⁴Process quality and Innovation, Netherlands Railways, 3500 HA Utrecht, The Netherlands

Econometric Institute Report Series EI2020-06

In this paper, we propose and analyze an online, decentralized policy for dispatching vehicles in a multi-line public transit system. In the policy, vehicles arriving at a terminal station are assigned to the lines starting at the station in a round-robin fashion. Departure times are selected to minimize deviations from a certain target headway. We prove that this policy is self-organizing: given that there is a sufficient number of available vehicles, a timetable spontaneously emerges that meets the target headway of every line. Moreover, in case one of the vehicles breaks down, the remaining vehicles automatically redistribute over the network to re-establish such a timetable. We present both theoretical and numerical results on the time until a stable state is reached and on how quickly the system recovers after the breakdown of a vehicle. These promising results suggest that our self-organizing policy could be useful in situations where centralized dispatching is impractical or simply impossible due to an abundance of disruptions or the absence of information systems.

1 Introduction

Self-organizing strategies are a promising concept to increase the resilience of urban public transit systems. In such a strategy, the concept of a schedule or timetable is abandoned. Instead, departure times and/or destinations of vehicles are determined locally at stations according to an easy-to-implement policy. In the absence of perturbations, an adequate self-organizing policy causes the system to converge to some preferable state, typically a periodic repetition of services with constant headways (the time between consecutive services). As a result, the impact of disruptions always dies out spontaneously, without intervention by a

central control authority. In this paper, we propose and analyze an easy-to-implement decentralized policy for dispatching vehicles in a multi-line public transit system and prove the resulting system exhibits self-organizing behavior.

There are multiple advantages of self-organizing strategies over centralized approaches to dispatch vehicles. First of all, the information systems that are required for centralized control are complex and expensive. Therefore, it might not be a viable option to all public transit operators, such as smaller operators or operators in developing countries. Secondly, centralized control requires communication between the vehicles and a central control center, where dispatching decisions are made, possibly using a decision support system. This process, involving communication, determining a new schedule and coordination, can be time consuming and prone to errors. Especially if rescheduling needs to be performed frequently, for example because congestion causes travel times to be highly volatile, this can be cumbersome. Moreover, public transit systems and railway systems in particular may suffer from *out-of-control* situations, where extreme events such as power outages or blizzards result in largely disrupted operations (Dekker et al., 2018; Van Lieshout, Bouman, & Huisman, 2020). In such situations, centralized rescheduling approaches are ineffective due to the sheer size of the disruptions and a lack of complete information available at the central control center. In contrast, a self-organizing strategy is typically easy and very fast to apply in all situations, without requiring communication or even the use of a computer. Besides, note that a self-organizing strategy could also serve as a back-up plan: operators may want to apply centralized control as a default and switch to the self-organizing approach when disruptions make the system too difficult to manage centrally.

The self-organizing approach has already been introduced in the context of public transit by Bartholdi and Eisenstein (2012), who developed a simple rule for holding buses at a control point with the goal to reduce headway variation (variation in the time between consecutive services) and prevent bus bunching. The authors show analytically that for the case with a single circular line, as long as there are no perturbations, under their policy any starting position will converge to some fixed point where the headways between all vehicles are equal. Note that this directly implies that whenever there is a perturbation, the headways will automatically self-equalize after some time. Even when one of the vehicles breaks down, a new system headway will naturally emerge. This approach was later extended by Liang, Zhao, Lu, and Ma (2016) and Zhang and Lo (2018), who consider both the backward headway and the forward headway when deciding how long a vehicle should wait at a control point.

In this paper, we extend the literature on self-organizing dispatching strategies by considering more complex public transit networks. Specifically, we consider networks consisting of multiple lines, with the condition that all lines have the same target headway. In such a system, when a vehicle reaches a terminal station of a line one not only needs to decide *when* to depart again, but also *which line* to perform. For

this problem, we propose and theoretically analyze an easy-to-implement decentralized dispatching policy. In our policy, every terminal station maintains a cyclic ordering of its outgoing lines and keeps track of the most recent departure times of these lines. Vehicles arriving at the station are assigned to the outgoing lines in round-robin fashion, according to the cyclic ordering. The departure times of vehicles are chosen such that deviations from the target headway are minimized.

Our main contribution is that we prove that our policy is self-organizing, leading to emergent behavior. Once converged, the decentralized policy matches the performance that can be achieved under centralized control. As long as the number of vehicles is large enough to perform a schedule meeting the target headways, our policy guarantees convergence to such a schedule. This result holds regardless of the initial locations of the vehicles. As a consequence, even when one of the vehicles breaks down or a bus returns to the depot at the end of the driver’s shift, the remaining vehicles spontaneously redistribute over the network to again meet the target headway of all lines. In numerical experiments we illustrate that this happens rapidly, such that the impact of a disruption is quickly absorbed. In case the number of vehicles is not sufficient to meet the target headways using a centralized approach, we prove that our policy keeps the headways, on average, as small as possible given the number of available vehicles. Finally, we also derive upper bounds on the largest headway that can occur and the stabilization time.

The remainder of this paper is structured as follows. In Section 2, we describe the problem setting and explain the policy. In Section 3, we discuss related literature. In Section 4, we theoretically analyze the performance of the policy. In Section 5, we discuss the results of a series of experiments that illustrate the practical performance of the policy. Finally, we conclude the paper in Section 6.

2 The Policy

2.1 Problem Setting and Notation

We represent the public transit system by a directed network $\mathcal{G} = (\mathcal{S}, \mathcal{L})$, where \mathcal{S} is the set of terminal stations and \mathcal{L} the set of lines. The intermediate stations are not relevant for our policy and therefore not included in the network. We assume that the network is symmetric, such that for every line $(s \rightarrow s') \in \mathcal{L}$, the reverse line $(s' \rightarrow s)$ is also an element of \mathcal{L} . Furthermore, we assume that \mathcal{G} is connected (otherwise the connected components can be considered separately). Every line $l \in \mathcal{L}$ is characterized by a travel time denoted by t_l . We allow for asymmetric travel times, so the travel time of a line and its reverse line are not required to be equal. Every line has the same target headway, which we denote as H . In other words, the goal is to operate each line every H time units. We assume that all travel times and H are integer. We let

$\delta^+(s)$ and $\delta^-(s)$ and denote the set of lines originating and terminating at $s \in \mathcal{S}$, respectively.

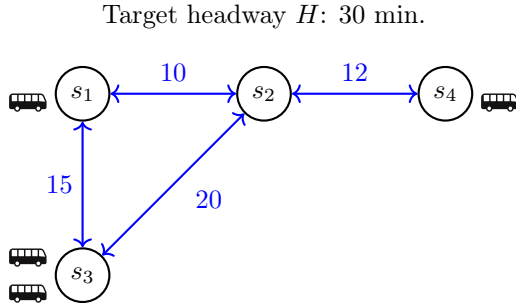


Figure 1: Illustration of the problem setting. Travel times are symmetric and given in minutes.

We assume there is a fixed number of vehicles available in the system, which we denote as n . At the moment of initialization, all vehicles are at stations. Vehicles are allowed to switch between lines at the terminal stations, but are not allowed to deadhead (drive without passengers). Therefore, after a vehicle performs line $(s \rightarrow s')$, the next line the vehicle is assigned to must be an element of $\delta^+(s')$. To meet all the target headways, one needs at least n^* vehicles, with

$$n^* = \frac{\sum_{l \in \mathcal{L}} t_l}{H}.$$

In general, n^* may be fractional, so it can be rounded up to the next integer to obtain a stronger bound. Furthermore, this bound on the number of required vehicles does not depend on whether the system is operated using a centralized or a decentralized approach. Although operators will naturally choose a target headway that is feasible given their fleet size, we consider both the case where $n \geq n^*$ and where $n < n^*$, as the latter may be relevant when there is a breakdown of a vehicle or travel times are longer than anticipated.

A visual illustration of the problem setting is provided in Figure 1, depicting a network of four stations, four lines and four vehicles. In this example, $n^* = 3.8$, so at least four vehicles are necessary to meet the target headways.

2.2 Policy Definition

We now propose a policy for dispatching vehicles at a terminal station. The policy determines the next line and the next departure time of a vehicle arriving at a terminal station. In the policy, the lines starting at a terminal station are selected in round-robin fashion, according to a fixed (but arbitrary) cyclic order. Departure times are based on the previous departure times of the lines, which are assumed to be known at the station. The departure time is taken to be the maximum of the target departure time, which is equal to the sum of the previous departure time and the target headway, and the current time (as it is not possible

Table 1: The state at station s_2 at two different time instants.

(a) Current time: $t = 9:10$

Line	Next	Prev. Dep.	Target Dep.
$s_2 \rightarrow s_1$		8:50	9:20
$s_2 \rightarrow s_3$		9:00	9:30
$s_2 \rightarrow s_4$	●	8:35	9:05

(b) Current time: $t = 9:15$

Line	Next	Prev. Dep.	Target Dep.
$s_2 \rightarrow s_1$	●	8:50	9:20
$s_2 \rightarrow s_3$		9:00	9:30
$s_2 \rightarrow s_4$		9:10	9:40

to depart in the past). Note that any minimum required time between services can be incorporated in the definition of the travel times, so we assume without loss of generality that an arriving vehicle can depart immediately.

As an example, suppose a vehicle arrives at station s_2 from Figure 1 at 9:10. Table 1a displays the relevant information at s_2 at this time, indicating which line should be performed next, the previous departure times of the lines starting at s_2 and the target departure times. Our policy assigns the arriving vehicle to line ($s_2 \rightarrow s_4$), as it is indicated that this line should be performed next. Naturally, the previous departure time of this line is also the longest ago. As the target departure time has already passed, the departure time is set at 9:10. Table 1b shows the updated information after the departure. Note that the target departure time of line ($s_2 \rightarrow s_4$) is now equal to 9:40, as it is only based on the most recent departure time. Suppose that the next arrival occurs at time 9:15. The policy assigns the arriving vehicle to line ($s_2 \rightarrow s_1$). As the target departure time of line ($s_2 \rightarrow s_1$) is 9:20, the policy instructs the vehicle to wait for 5 minutes and depart exactly at 9:20.

For a formal definition of the policy, let us (arbitrarily) order the lines starting at station $s \in \mathcal{S}$ as $l_1^s, l_2^s, \dots, l_{|\delta^+(s)|}^s$, representing the cyclic order in which the lines from this station are performed. Let $l_{\text{next}}^s \in \delta^+(s)$ denote the next line to be performed from station s and let τ_l denote the current target departure time of line l (at initialization, all target departure times are 0 and $l_{\text{next}}^s = l_1^s$). Suppose at time t_{now} , a vehicle arrives at station s and $l_{\text{next}}^s = l_i^s$. Our policy assigns the arriving vehicle to line l_i^s and schedules it at time $t' = \max\{\tau_{l_i^s}, t_{\text{now}}\}$. Next, the policy updates the target departure time of the selected line:

$$\tau_{l_i^s} \leftarrow t' + H.$$

Finally, the policy updates l_{next}^s according to the order of the lines:

$$l_{\text{next}}^s \leftarrow l_{(i \bmod |\delta^+(s)|)+1}^s.$$

3 Related Literature

3.1 Self-Organizing Approaches in Public Transit

Bartholdi and Eisenstein (2012) were the first to introduce the concept of self-organization or self-coordination in the field of public transit scheduling. In their approach a vehicle is delayed at a control point by a time proportional to the headway to the trailing vehicle. The authors prove that for the case with a single circular line, this policy ensures that all headways self-equalize over time, regardless of the initial locations of the vehicles. This approach has been extended by Liang et al. (2016) and Zhang and Lo (2018), who consider both the backward headway and the forward headway when computing how long a vehicle should be delayed, resulting in a faster convergence rate. Zhang and Lo (2018) also provide theoretical evidence that the headway variation remains limited under stochastic travel times. However, only single-line systems are considered in these papers.

3.2 Multi-Line Control

For multi-line systems, the approach of Argote-Cabanero, Daganzo, and Lynn (2015) is closest to our work. In this study, the authors propose an adaptive control rule for holding, accelerating and decelerating vehicles with the aim to adhere to the schedule as well as possible. However, the possibility to dynamically switch lines after a vehicle reaches a terminal station is not considered. Furthermore, this approach requires the specification of a target schedule and a number of functions and parameters. In contrast, our approach does explicitly allow vehicles to change lines in order to better spread vehicles over the network and only requires the specification of a target headway, making the policy easier to implement. Other papers focusing on multi-line systems, such as Hernández, Muñoz, Giesen, and Delgado (2015) and Petit, Lei, and Ouyang (2019), consider centralized optimization based approaches to reduce bus bunching, as opposed to applying a local decision rule.

3.3 Rotor-Router Systems

Our policy can be viewed as a generalization of the *rotor-router model*, which was originally introduced in Priezzhev, Dhar, Dhar, and Krishnamurthy (1996) as the deterministic counterpart of a random walk on a graph. In the random walk on a graph, one or more agents move over a graph at discrete and synchronous steps. The next edge to be traversed by an agent is selected randomly from the set of incident edges of the current node where the agent is located (Lovász, 1993). In the rotor-router model, a node does not send agents visiting it to a random neighbour, but instead selects the incident edges in round-robin fashion. That is, every node in the graph maintains a cyclic ordering of its incident edges and has a pointer indicating the next edge to be traversed by an entering agent. Whenever an agent enters a node, the pointer is advanced to the next edge in the cyclic ordering.

As our policy assigns arriving vehicles at a station to lines in a round-robin fashion, similar to the rotor-router system. On the other hand, in the rotor-router model it takes one time step to traverse an edge, whereas in our case a line can have any positive integer valued travel time. Moreover, our policy sometimes instructs to hold a vehicle at a station to meet the target headway, where agents in the rotor-router model move in every time step. However, we will see that some of the results for the rotor-router model also hold for our policy.

For the case where there is only one agent, Priezzhev et al. (1996) proves for the rotor-router mechanism that after a sufficiently long time, the agent gets locked-in in a cycle where every edge is traversed exactly once in both directions. Yanovski, Wagner, and Bruckstein (2003) and Bampas et al. (2009) show that the lock-in time is bounded by $2mD$, where m is the number of edges in the graph and D the diameter of the graph. The ability of the system to recover from, for example, edge deletions (corresponding to the removal of lines in a public transit network) is investigated in Bampas et al. (2017). For the case with multiple agents, Wagner, Lindenbaum, and Bruckstein (1999) prove that the difference in the number of traversals of two edges cannot grow unbounded. Yanovski et al. (2003) present a stronger bound for the maximum difference between the number of traversals of two edges and also prove that a rotor-router system with multiple agents converges to a periodic motion. Chalopin et al. (2015) provide a further analysis of the limit behavior of the multi-agent rotor router system, and show that unlike the case with one agent the duration of the periodic motion (so the time until the system returns to the same state) can be superpolynomial in the number of edges. Finally, Dereniowski, Kosowski, Pająk, and Uznański (2016) prove that the time it takes until all edges are traversed with k agents is at least $\log(k)$ times shorter than with one agent.

4 Theoretical Analysis

In this section, we analyze the emerging behavior of the system in case all vehicles are scheduled according to the proposed policy. First, we investigate whether the policy services all lines in a fair or balanced manner, as preferably each line should have approximately the same number of departures. Secondly, we analyze the long run behavior of the system and investigate whether the target headway of every line is met. Thirdly, we provide worst case results on the maximum headway that can occur under the policy and the time it can take before the system reaches a stable state. We conclude this section with a discussion regarding the performance of the policy in case there is no common target headway of all lines.

4.1 Balanced Services

We first analyze the extent to which our policy leads to a balanced service of all lines. Ideally, at any point in time, every line should have approximately the same number of departures. Formally, let $f(s \rightarrow s')$ denote the number of departures of line $(s \rightarrow s') \in \mathcal{L}$ up to time t . The lemmas and theorems that we prove hold for any t . Hence, for readability, we omit the index t . In this section, the goal is to show that the difference between $f(s_1 \rightarrow s'_1)$ and $f(s_2 \rightarrow s'_2)$ is bounded for two arbitrary lines $(s_1 \rightarrow s'_1), (s_2 \rightarrow s'_2) \in \mathcal{L}$

As the policy serves lines in a round-robin fashion, for two lines $(s \rightarrow s')$ and $(s \rightarrow s'')$ originating at the same station, it holds that $|f(s \rightarrow s') - f(s \rightarrow s'')| \leq 1$. The first part of our analysis only depends on this property of our policy. Because this property is shared with the rotor-router system, we apply a similar analysis as presented by Yanovski et al. (2003).

Let $\mathcal{S}_1, \mathcal{S}_2$ be a partition of the set of all stations. Then, we define $f(\mathcal{S}_1 \rightarrow \mathcal{S}_2)$ as the number of times lines starting in \mathcal{S}_1 and ending in \mathcal{S}_2 have been performed up to some time t (again we omit the index t). We also refer to $f(\mathcal{S}_1 \rightarrow \mathcal{S}_2)$ as the flow from \mathcal{S}_1 to \mathcal{S}_2 . As it holds for every of the n vehicles that it is impossible to cross from \mathcal{S}_1 to \mathcal{S}_2 twice, without crossing back from \mathcal{S}_2 to \mathcal{S}_1 , we can make the following observation (first made by Wagner et al. (1999)).

Observation 1. *For a partition $\mathcal{S}_1, \mathcal{S}_2$ of the set of stations, it holds that $f(\mathcal{S}_1 \rightarrow \mathcal{S}_2) - f(\mathcal{S}_2 \rightarrow \mathcal{S}_1) \leq n$.*

Using Observation 1, it is possible to prove Lemma 2, which gives an upper bound on the difference between the number of times a line and its reverse line are performed.

Lemma 2. *For every line $(s \rightarrow s') \in \mathcal{L}$, it holds that $|f(s \rightarrow s') - f(s' \rightarrow s)| \leq n$.*

Proof. We define $f(s) := \min_{(s \rightarrow s') \in \delta^+(s)} f(s \rightarrow s')$ for a station $s \in \mathcal{S}$, denoting the minimum number of departures for any line leaving s . As the lines are always served in a round-robin fashion, we have that $0 \leq f(s \rightarrow s') - f(s) \leq 1$.

Now, suppose that the lemma is not true, such that there exists a pair of opposite lines $(s \rightarrow s')$ and $(s' \rightarrow s)$ with $f(s \rightarrow s') = j$ and $f(s' \rightarrow s) \leq j - n - 1$. By definition, it holds that $f(s) \geq j - 1$ and $f(s') \leq j - n - 1$. Consider the partition $\mathcal{S}_1, \mathcal{S}_2$ where $\mathcal{S}_1 = \{s \in \mathcal{S} : f(s) \geq j - n\}$ and $\mathcal{S}_2 = \{s \in \mathcal{S} : f(s) \leq j - n - 1\}$. We have that $s \in \mathcal{S}_1$ and $s' \in \mathcal{S}_2$. Let the number of lines crossing from \mathcal{S}_1 to \mathcal{S}_2 be m . As the flow from s to s' is j and the flow over all other lines from \mathcal{S}_1 to \mathcal{S}_2 is at least $j - n$, we find that

$$f(\mathcal{S}_1 \rightarrow \mathcal{S}_2) \geq (m - 1)(j - n) + j.$$

Similarly, the flow from \mathcal{S}_2 to \mathcal{S}_1 is at most

$$f(\mathcal{S}_2 \rightarrow \mathcal{S}_1) \leq (m - 1)(j - n) + j - n - 1.$$

Therefore, it follows that

$$f(\mathcal{S}_1 \rightarrow \mathcal{S}_2) - f(\mathcal{S}_2 \rightarrow \mathcal{S}_1) \geq n + 1.$$

As this contradicts Observation 1, the assumption that the lemma is not true must be wrong. \square

We now present a theorem that bounds the difference between $f(s_1 \rightarrow s'_1)$ and $f(s_2 \rightarrow s'_2)$ for two arbitrary lines, which implies that the number of times two lines are performed cannot differ too much, at any point in time. To do so, let $\text{dist}(s_1, s_2)$ denote the number of lines in the shortest path from s_1 to s_2 (shortest in terms of number of lines).

Theorem 3. *For two lines $(s_1 \rightarrow s'_1), (s_2 \rightarrow s'_2) \in \mathcal{L}$, it holds at any time that $|f(s_1 \rightarrow s'_1) - f(s_2 \rightarrow s'_2)| \leq (\text{dist}(s'_1, s_2) + 1)(n + 1)$.*

Proof. First, we prove a bound on the difference between the number of times two consecutive lines are performed. Thereafter, we consider a shortest path between s'_1 and s_2 and iteratively apply this bound to prove the theorem.

By Lemma 2 it holds that $f(s \rightarrow s') \leq f(s' \rightarrow s) + n$ and by definition of the policy it holds that $f(s' \rightarrow s) \leq f(s' \rightarrow s'') + 1$. Hence, for two consecutive lines we find that $f(s \rightarrow s') \leq f(s' \rightarrow s'') + n + 1$.

Next, let $s'_1 = s_a, s_b, \dots, s_p = s_2$ denote a shortest path from s'_1 to s_2 . It holds that

$$\begin{aligned}
f(s_1 \rightarrow s'_1) &\leq f(s_a \rightarrow s_b) + n + 1 \\
&\leq f(s_b \rightarrow s_c) + 2(n + 1) \\
&\vdots \\
&\leq f(s_0 \rightarrow s_p) + \text{dist}(s'_1, s_2)(n + 1) \\
&\leq f(s_2 \rightarrow s'_2) + (\text{dist}(s'_1, s_2) + 1)(n + 1).
\end{aligned}$$

Hence, $f(s_1 \rightarrow s'_1) - f(s_2 \rightarrow s'_2) \leq (\text{dist}(s'_1, s_2) + 1)(n + 1)$. The theorem follows by symmetry. \square

The desirable property of the bound proven in Theorem 3 is that it does not depend on t . Therefore, it holds that the difference between the number of times two lines are performed cannot grow unbounded. In what follows, we use this observation to characterize the long run behavior of the system.

4.2 Limit Behavior

In this section, we analyze the emerging properties of the system in the long run. The first result states that after some time it is guaranteed that the system enters a periodic motion (i.e. starts to cycle) and that every line is performed the same number of times in one cycle. The proof is an adaption of the proof by Yanovski et al. (2003) of the same property for the rotor-router system.

Lemma 4. *After a certain finite time, the system enters a periodic motion. In every cycle, every line is performed the same number of times.*

Proof. As travel times and the target headway H are all integers, vehicles always depart at integer time points. Hence, it suffices to consider the system only at integer time points. The state of the system at an integer time point can be represented by all locations of the vehicles and the time since the latest departure of all lines. According to Theorem 3, the number of times two lines are performed cannot differ too much. This implies that the time since the latest departure of a line cannot be unbounded. It then follows that the number of possible states is finite. Furthermore, since the policy is deterministic, it must be that the system returns to the same state, at which point the system starts to cycle. Therefore, the system enters a periodic motion.

To prove the second part of the lemma, note that if the number of times two lines are performed during a cycle would be different, over time the difference would grow without bound, contradicting Theorem 3. \square

It follows from the proof of Lemma 4 that we can represent the state of the system at time t using some state vector \mathcal{V}_t . Following Chalopin et al. (2015), we call a state \mathcal{V}_t *stable* if there exists $t' > t$ such that $\mathcal{V}_{t'} = \mathcal{V}_t$. Equivalently, we say that the system has stabilized once it has entered the periodic motion. By Lemma 4, \mathcal{V}_t will always be stable for large enough t . The *stabilization time*, denoted as T_{stable} , is the smallest value such that $\mathcal{V}_{T_{\text{stable}}}$ is stable. Furthermore, the *periodicity*, denoted as T_{period} , is the smallest value such that $\mathcal{V}_{T_{\text{stable}}+T_{\text{period}}} = \mathcal{V}_{T_{\text{stable}}}$. In the remainder of this subsection, we analyze the properties of the system once it reaches a stable state in more detail. In Section 4.3, we analyze how large the stabilization time can be in the worst case.

To provide more insight into the emerging behavior of the system, we first consider the number of idle vehicles at stations. Let $\text{arr}_s(a, b)$ and $\text{dep}_s(a, b)$ denote the number of arrivals and departures at s in the half-open interval $[a, b)$ respectively. Then, the number of idle vehicles at station t , denoted by $i_s(t)$, satisfies

$$i_s(t) = i_s(0) + \text{arr}_s(0, t) - \text{dep}_s(0, t).$$

This brings us to the following lemma:

Lemma 5. *For the number of idle vehicles at a station, it holds that $i_s(t) \geq i_s(t + H)$.*

Proof. Assume $i_s(t) < i_s(t + H)$. Then, it must hold that $\text{dep}_s(t, t + H) < \text{arr}_s(t, t + H)$. As every line has at most one departure per H time units, the number of arrivals at s during H time units, is at most $|\delta^-(s)| = |\delta^+(s)|$ (the number of lines that terminates at s). Hence, we find that $\text{dep}_s(t, t + H) < |\delta^+(s)|$. As such, there exists a line l without a departure in the interval $[t, t + H)$. This implies that $i_s(t + H) = 0$, as otherwise line l would have had a departure in this interval. As $0 = i_s(t + H) > i_s(t) \geq 0$, we reach a contradiction. The conclusion is that $i_s(t) \geq i_s(t + H)$. \square

Next, we analyze a global performance indicator, the *utilization*. Let $\gamma_s(a, b)$ denote the total idle time of vehicles at station s in the interval $[a, b)$. The utilization, denoted by $\text{util}(a, b)$, represents the average proportion of time that the vehicles are driving in the interval $[a, b)$:

$$\text{util}(a, b) := 1 - \frac{\sum_{s \in \mathcal{S}} \gamma_s(a, b)}{(b - a)n}.$$

By Lemma 5, the number of idle vehicles cannot increase over time. Therefore, it holds that the utilization cannot decrease over time. The next lemma formalizes this statement.

Lemma 6. *For the utilization, it holds that $\text{util}(t, t + H) \leq \text{util}(t + H, t + 2H)$. Moreover, for $t' \geq T_{\text{stable}}$*

it holds that

$$\text{util}(t', t' + H) = \text{util}(t' + iH, t' + (i + 1)H) \text{ for any } i \in \mathbb{Z}^+.$$

Proof. Observe that $\gamma_s(a, b) = \int_a^b i_s(t)dt$. By applying Lemma 5 we find

$$\begin{aligned} \gamma_s(t, t + H) &= \int_t^{t+H} i_s(x)dx \\ &\geq \int_t^{t+H} i_s(x + H)dx \\ &= \int_{t+H}^{t+2H} i_s(x)dx \\ &= \gamma_s(t + H, t + 2H). \end{aligned}$$

Therefore, it holds that

$$\begin{aligned} \text{util}(t, t + H) &= 1 - \frac{\sum_{s \in \mathcal{S}} \gamma_s(t, t + H)}{Hn} \\ &\leq 1 - \frac{\sum_{s \in \mathcal{S}} \gamma_s(t + H, t + 2H)}{Hn} \\ &= \text{util}(t + H, t + 2H). \end{aligned}$$

For the second part of the lemma, note that by the periodicity of the system, we have that for $t' > T_{\text{stable}}$ $\text{util}(t', t' + H) = \text{util}(t' + iT_{\text{period}}, t' + iT_{\text{period}} + H)$ for any $i \in \mathbb{Z}^+$. Since $\text{util}(t, t + H) \leq \text{util}(t + H, t + 2H)$, it follows for $t' > T_{\text{stable}}$ that $\text{util}(t', t' + H) = \text{util}(t' + iH, t' + (i + 1)H)$ for any $i \in \mathbb{Z}^+$. □

From the above lemma, it follows that once the system reaches a stable state, the utilization is constant over consecutive intervals of duration H , even though T_{period} , the duration of the periodic motion, can in general be strictly larger than H . We formally define this limit value of the utilization as $u := \text{util}(T_{\text{stable}}, T_{\text{stable}} + H)$, to which we refer as the *stable utilization*. Note that the policy ensures that at all lines are performed at least once in one cycle of the periodic motion, such that $u > 0$.

Before we state the main theorem, recall that n^* is a lower bound on the number of vehicles required to meet the target headways. Theorem 7 shows that the behavior of the system depends on whether $n < n^*$ or $n \geq n^*$.

Theorem 7. *If $n < n^*$, then the stable utilization u equals 1 and the average headway of all lines during the periodic motion equals $\frac{n^*}{n}H$. Otherwise, the headways of all lines during the periodic motion equal H for all lines and the stable utilization equals $\frac{n^*}{n}$.*

Proof. As we have shown that the utilization converges to a certain stable utilization $0 < u \leq 1$, we can distinguish the following two cases:

Case I: $0 < u < 1$. This implies that there exists at least one station where there is strictly positive idle time during the periodic motion. It must be that the lines originating from this station are performed once per H time units, as a vehicle only waits if it is in time to meet the target headway of its next line. As all lines are performed the same number of times in the periodic motion by Lemma 4, it follows that $T_{\text{period}} = H$ and every line is performed exactly once per H time units in both directions. By definition, this implies that $n \geq n^*$. As every line is operated once per H time units, the utilization converges to

$$u = \frac{\sum_{l \in \mathcal{L}} t_l}{nH} = \frac{n^*}{n}.$$

Case II: $u = 1$. As every line can be operated at most once every H time units, this implies that $n \leq n^*$. Moreover, it follows from Lemma 4 that the system enters a periodic motion in which all vehicles have no idle time and all lines are performed the same number of times. Let g denote the number of times each line is performed during a cycle. As all vehicles are running all the time, it holds that

$$nT_{\text{period}} = g \sum_{l \in \mathcal{L}} t_l = gn^*H.$$

Consequently, the average headway of all lines, which we denote as \bar{H} , equals

$$\bar{H} = \frac{T_{\text{period}}}{g} = \frac{n^*}{n}H.$$

□

The above theorem provides a concise characterization of the behavior of the system under the proposed policy. The result can be seen to be optimal in some sense. If the number of vehicles is large enough to meet the target headways using centralized control, our decentralized policy is also able to meet the target headways. In case there are not sufficient vehicles to meet the target headways, every vehicle is used all the time and every line has the same average headway, equal to the smallest headway possible under centralized scheduling. Moreover, if $n \geq n^* + 1$, there is some slack in the system, such that if a vehicle breaks down, the headways of all lines again converge to H . In the other case, there is no slack in the system and every breakdown of a vehicle leads to an increase in headways, and therefore to a reduction in passenger service.

4.3 Worst Case Analysis

In this part, we provide worst case results of the headway deviation in case $n < n^*$ and on the time it takes to reach a stable state.

4.3.1 Worst Case Headway Deviation

In contrast with the results Bartholdi and Eisenstein (2012) and Zhang and Lo (2018) obtained for the single-line case, in case $n < n^*$ (so there are not enough vehicles available), our policy leads to convergence of the *average headways* of all lines, but not necessarily to convergence of the headways themselves. A natural question to ask is how large the maximum headway can become in the worst case. Theorem 8 shows that the headway cannot be larger than $H + (n^* - n)H$, such that the excess headway is never larger than $(n^* - n)H$.

Theorem 8. *If $n < n^*$, once the system is in a stable state, all headways are at most $H + (n^* - n)H$.*

Proof. For this proof it is convenient to think of every line l as having length t_l and think of every vehicle as a snake having length H and moving 1 unit distance per unit time (such that it takes t_l time units to traverse a line). As $n < n^*$, it holds according to Theorem 7 that the system converges to a periodic motion where the snakes are constantly moving. Furthermore, the policy ensures that consecutive departures of the same line are always separated by at least H time units, such that two snakes, despite having length H , cannot occupy the same part of a line. Therefore, once the system has stabilized, the snakes cover a part of the network of length nH . The part of the network that is not covered by any of the snakes then has length, $\sum_{l \in \mathcal{L}} t_l - nH = n^*H - nH = (n^* - n)H$. Thus, whenever a snake starts traversing a line, the distance between the front of the snake and the tail of the preceding snake on the line is at most $(n^* - n)H$. As the length of every snake is H , it follows that the distance between the fronts of two vehicles is, at any time, at most $H + (n^* - n)H$. Hence, the time between two consecutive departures of the same line is at most $H + (n^* - n)H$. \square

This theorem has a nice interpretation, as it shows that if there is only a small shortage of vehicles, the headways cannot become very large. As long as the discrepancy between n and n^* is not too big, the target headways are met reasonably well. For example, if $n = n^* - 1$, the maximum headway that can occur is $2H$.

4.3.2 Stabilization Time

In this part, we derive worst case bounds on the stabilization time T_{stable} . We investigate the time until stabilization for two special cases. In both cases, the stabilization time depends on the unweighted diameter

of the network, which is denoted by D and represents the maximum number of lines one is required to traverse when traveling from one station to another. First, we analyze the case where $n = n^* = 1$, so a single vehicle suffices for meeting the target headway.

Theorem 9. *If $n = n^* = 1$, it holds that $T_{\text{stable}} \leq DH$.*

Proof. As there is only one vehicle, the system is stabilized if the vehicle continuously performs an Euler tour every H time units. Bampas et al. (2009) shows that for the rotor-router model with a single agent, an Euler tour is established in "phases" and that in the worst case, D phases are required. This result directly extends to our setting. In every phase, the vehicle performs a tour starting and ending at s_0 , the initial location of the vehicle. A phase ends when all the lines originating at s_0 have been performed during that phase. Furthermore, in the worst case, the cyclic order of the outgoing lines at every s is such that in phase i , station s is visited if and only if $\text{dist}(s_0, s) \leq i$. Therefore, after round D the vehicle will have entered the periodic motion and continuously perform an Euler tour. Furthermore, since $n^* = 1$ every closed tour over the network takes at most H time units, which implies that the duration of every round is H . It follows that the system stabilizes in the worst case at time DH . \square

Next, we analyze the case where $t_l = H$ for all $l \in \mathcal{L}$ and $n = n^* = |\mathcal{L}|$. Since all travel times are equal to the target headway, we can analyze the system in iterations of duration H and define $i_s(m)$ as the number of vehicles located at station s at the end of iteration m . The system has stabilized if and only if $i_s(m) = \text{deg}(s)$ for every $s \in \mathcal{S}$, where $\text{deg}(s)$ denotes the degree of station s in the network (the number of lines in the set $\delta^+(s)$). This motivates the following definition:

Definition. We define $C_s(m) = i_s(m) - \text{deg}(s)$ as the *charge* of station s after iteration m . The station s is called *positively charged* if $C_s(m) > 0$ and *negatively charged* if $C_s(m) < 0$. Otherwise, the station is called *neutral*.

We can observe that the total charge over the network equals zero:

$$\sum_{s \in \mathcal{S}} C_s(m) = \sum_{s \in \mathcal{S}} (i_s(m) - \text{deg}(s)) = n - \sum_{s \in \mathcal{S}} \text{deg}(s) = n - n^* = 0.$$

According to the policy, the number of vehicles leaving station s in iteration $m+1$ equals $\min\{i_s(m), \text{deg}(s)\}$. As the number of vehicles entering station s in an iteration is at most $\text{deg}(s)$, it follows that the charge of a neutral or positively charged station can never increase. Hence, a station can change from being positively charged to neutral, but not vice versa. On the other hand, until the system stabilizes, it is possible that negatively charged stations become neutral and vice versa.

We define the potential function $\Phi(m) = \sum_{s \in \mathcal{S}: C_s(m) > 0} C_s(m)$, equal to the sum of the positive charges. As the charge of positively stations can only decrease and neutral stations cannot become positively charged, it follows directly $\Phi(m) \geq \Phi(m+1)$. If $m \geq T_{\text{stable}}/H$, it holds that $\Phi(m) = 0$. In order to bound T_{stable} , we use the following result from the rotor-router system

Lemma 10. *For a rotor-router system with $k > 1$ agents, the cover time (the time until all edges have been visited at least once) on a graph with diameter D and m edges is at most $O\left(\frac{mD}{\log k}\right)$. If there is only 1 agent, the cover time is $O(mD)$.*

Proof. See Dereniowski et al. (2016). □

Theorem 11. *If $t_l = H$ for all $l \in \mathcal{L}$ and $n = n^* = |\mathcal{L}|$, it holds that $T_{\text{stable}} = O(|\mathcal{L}|^2 DH)$.*

Proof. Clearly, $\Phi(m)$ is integer and $0 \leq \Phi(m) \leq n - 1$. To bound the number of iterations the potential function can stay constant, we use the concept of *anti-vehicles*: whenever a line is **not** traversed by a vehicle in some iteration, it is traversed by an anti-vehicle. Thus, in case station s is negatively charged after iteration m , there are $i_s(m)$ regular vehicles and $-C_s(m)$ anti-vehicles leaving s in iteration $m+1$. Moreover, if $\Phi(m) = \Phi(m+1)$, it holds that in case s is negatively charged after iteration $m+1$, the absolute value of the charge equals the number of anti-vehicles entering s in iteration m . Hence, the anti-vehicles can be seen as carriers of the negative charge over the network.

Suppose that $\Phi(m) = f > 0$. Then, there are f anti-vehicles in the network after iteration m . We are interested in how long it takes until one of the anti-vehicles arrives at a positively charged station, as such an event reduces the potential function. As in any iteration the anti-vehicles traverse the lines that are not traversed by regular vehicles, it can be seen that that the anti-vehicles move according to the same policy as the regular vehicles, *but with the cyclic order of the lines reversed*. Moreover, since anti-vehicles move in every iteration (otherwise the number of anti-vehicles at a station would be larger than the degree), this system is equivalent to a rotor-router system. Therefore, the number of iterations until one of the anti-vehicles hits a positively charged station is at most the cover time of a rotor-router system with f agents. Applying Lemma 10 and using that the potential can decrease at most $n - 1 = |\mathcal{L}| - 1$ times and that every iteration takes H time units, it follows that

$$T_{\text{stable}} = O(|\mathcal{L}|DH) + \sum_{f=2}^{|\mathcal{L}|-1} O\left(\frac{|\mathcal{L}|D}{\log f} H\right) = O(|\mathcal{L}|^2 DH).$$

□

The results in Theorem 9 and 11 illustrate that the stabilization time depends on the diameter and number

of lines of the network and the target headway. For high-frequency networks that are highly connected, for example urban transit systems, stabilization occurs rapidly. For large elongated networks operated at lower frequencies, for example inter-regional transit systems, stabilization is established more slowly.

4.4 Different Target Headways

We conclude the theoretical analysis with the observation that it is unlikely that the presented results can be extended to settings where there is no common target headway, but lines may have different target headways. In particular, note that the behavior of the system is different depending on whether there are enough vehicles or not. As such, by using simulation, it is possible to find out how many vehicles are required to meet the target headway for every line. However, Van Lieshout (2019) proves that the problem of deciding whether a line plan with arbitrary frequencies can be operated with a certain number of vehicles (i.e. whether there exists a timetable that meets all target headways) is NP-complete. Hence, if a policy would have the property that the headways of all lines converge to the target headways if and only if the number of available vehicles is at least the minimum required number under centralized scheduling, this policy would solve an NP-complete problem. In the least, either the policy would prescribe decisions that cannot be computed in polynomial time, or the worst case time until convergence would have to be superpolynomial (unless $P=NP$).

On the other hand, this insight does not mean that our policy cannot be applied in systems where the target headway of lines is different. A first possibility is to decompose the network into sub-networks where there is a common target headway. Secondly, one could still choose to apply the policy and accept that there are no theoretical performance guarantees. To do so, the policy needs to be slightly generalized. Whenever there is a departure of line l' at time t' , the target departure time should now be updated according to the formula $\tau_{l'} \leftarrow t' + h_{l'}$, where h_l denotes the target headway of line l . We assess the performance of this policy numerically in the next section.

5 Numerical Experiments

In this section, we describe the results of a series of experiments that illustrate the practical performance of the proposed policy. First, we analyze how the time it takes to reach a stable state grows if the size of the network increases. Next, we investigate how long it takes to re-stabilize after one of the vehicles breaks down. In the two final experiments we test the performance of the policy in situations where the assumptions of the theoretical analysis of the previous section are not met. Specifically, we analyze the behavior of the system in case the lines do not have a common target headway and in case the travel times are not fixed but stochastic.

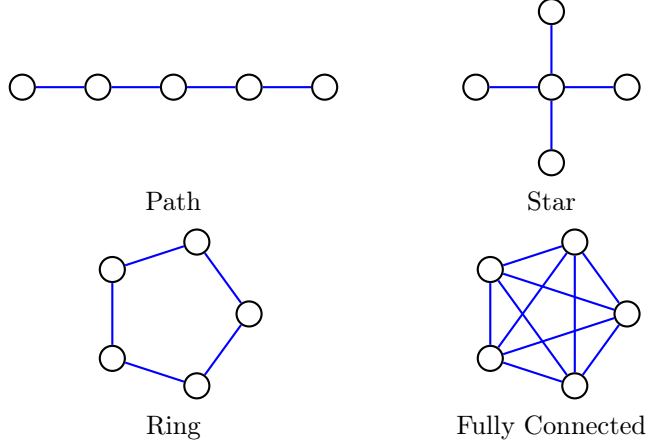


Figure 2: Different network topologies used in the numerical experiments.

5.1 Stabilization Time

We assess the time to reach a stable state for four types of network topologies: path, ring, star and fully connected. The differences between these types of networks are illustrated in Figure 2. In the first two experiments, we set the travel time of each edge equal to one time unit and set $H = 1$ and $n = n^*$.

In Figures 3a-3b, it is shown how the stabilization time grows with the size of the network, if all vehicles start from an unbalanced starting position. That is, we start with all vehicles at a single station. For the path network and the star network, we start with all vehicles at one of the outer stations. To minimize the rate at which the vehicles are spread out over the networks, for each station s , l_{next}^s is initialized such that the first time a vehicle enters s , the vehicle is sent back over the reverse line it came from. When comparing the stabilization time for a fixed number of stations, we find that the network topologies with the largest diameter also have the largest stabilization time, which we also expected based on the worst case results in Section 4.3.2. Moreover, the stabilization time grows at a faster rate for the path and ring network compared to the star and fully connected network, which is likely caused by the fact that the diameters of the latter two networks is constant in the number of stations, whereas the diameter of the former two networks increases linearly in the number of stations.

In Figures 4a-4b, it is shown how the time until stabilization grows with the size of the network, if the starting configuration is randomly generated. Here, we take the average over 2,500 runs. The required time is much shorter compared to the unbalanced starting situation. Interestingly, the fully connected network takes more time on average to stabilize than the star network, whereas in the previous experiment the star network required more time. Most likely, this is caused by the fact that if all vehicles often attend the same station, their departure times are more quickly coordinated. However, in the previous experiment all vehicles started at the same outer station of the star network, such that it took a long time before all vehicles had

departed from the starting station.

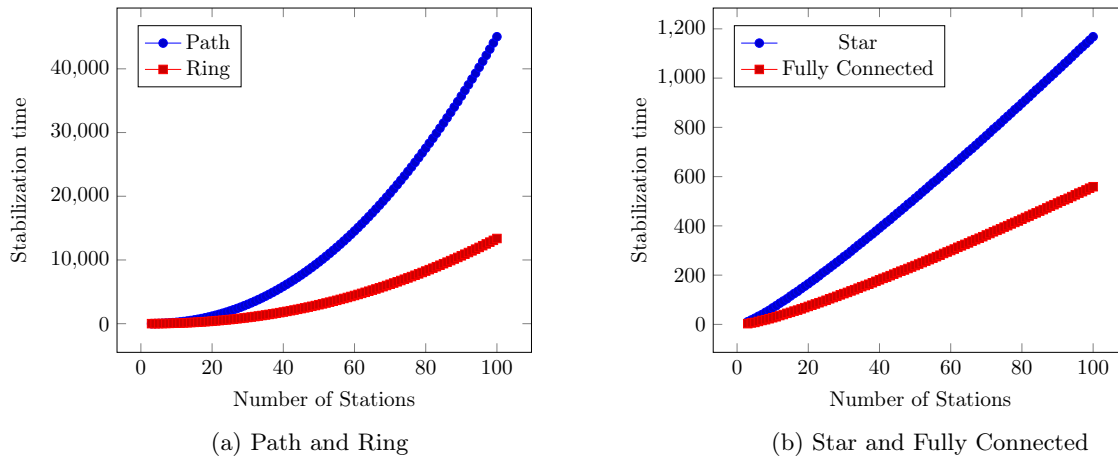


Figure 3: Stabilization time, starting from an unbalanced starting configuration.

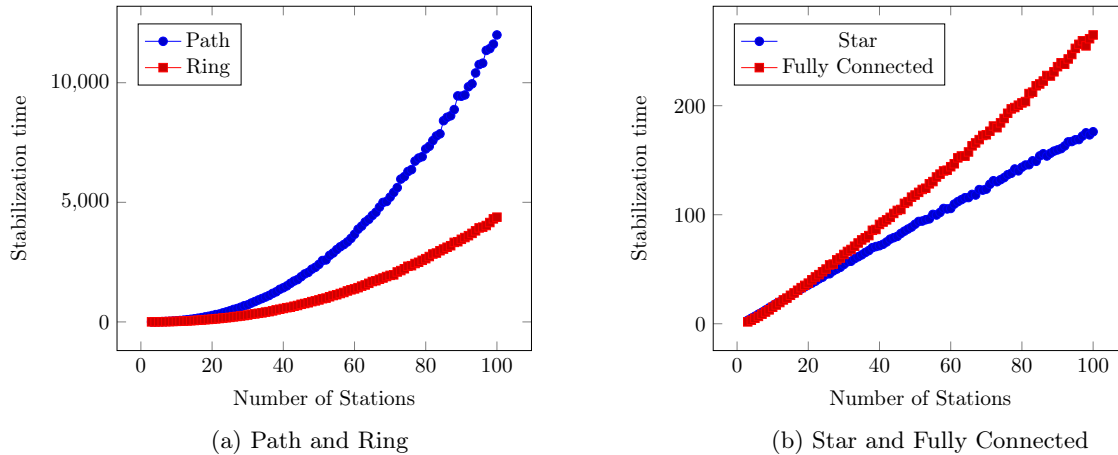


Figure 4: Stabilization time, starting from a random configuration, averaged over 2500 samples.

5.2 Re-Stabilizing after a Vehicle Breakdown

To get a better sense of the performance in practice, we perform a third experiment, where we start in a stable state (i.e. a feasible timetable). Then, we let one of the vehicles break down and analyze how long it takes to re-stabilize. We refer to the number of vehicles in the system above the minimum number of required vehicles to reach a stable state as the *buffer*. Provided that there is a buffer of at least one vehicle, we know that the system will always bounce back to a stable state after the breakdown. We perform this experiment on a star network with five lines with a target headway of 15 minutes and travel times uniformly drawn between 10 and 30 minutes. We start this experiment from a random stable state, which is achieved by having the system converge to a stable state from a random starting configuration.

In Figures 5a-d, the results of this experiment are visualized, for different sizes of the buffer and with ten randomly generated networks for each buffer size. The horizontal axis depicts the time since the breakdown and the vertical axis the current maximum headway in the network. As expected, the maximum headway in the system can be quite large right after the vehicle breakdown. However, the impact of the breakdown dies out rather quickly. Even with only a single vehicle as a buffer, the maximum headway in the system reduces to less than 20 minutes within the first hour. On the other hand, there can be a long tail off effect, as for some of the replications we observe it takes a long time before all headways really have converged to 15 minutes. However, in practice passengers will hardly notice the difference between a headway of 15 minutes and a slightly larger headway. We also observe that the maximum headway converges to 15 minutes much faster when there is a larger buffer in the number of vehicles.

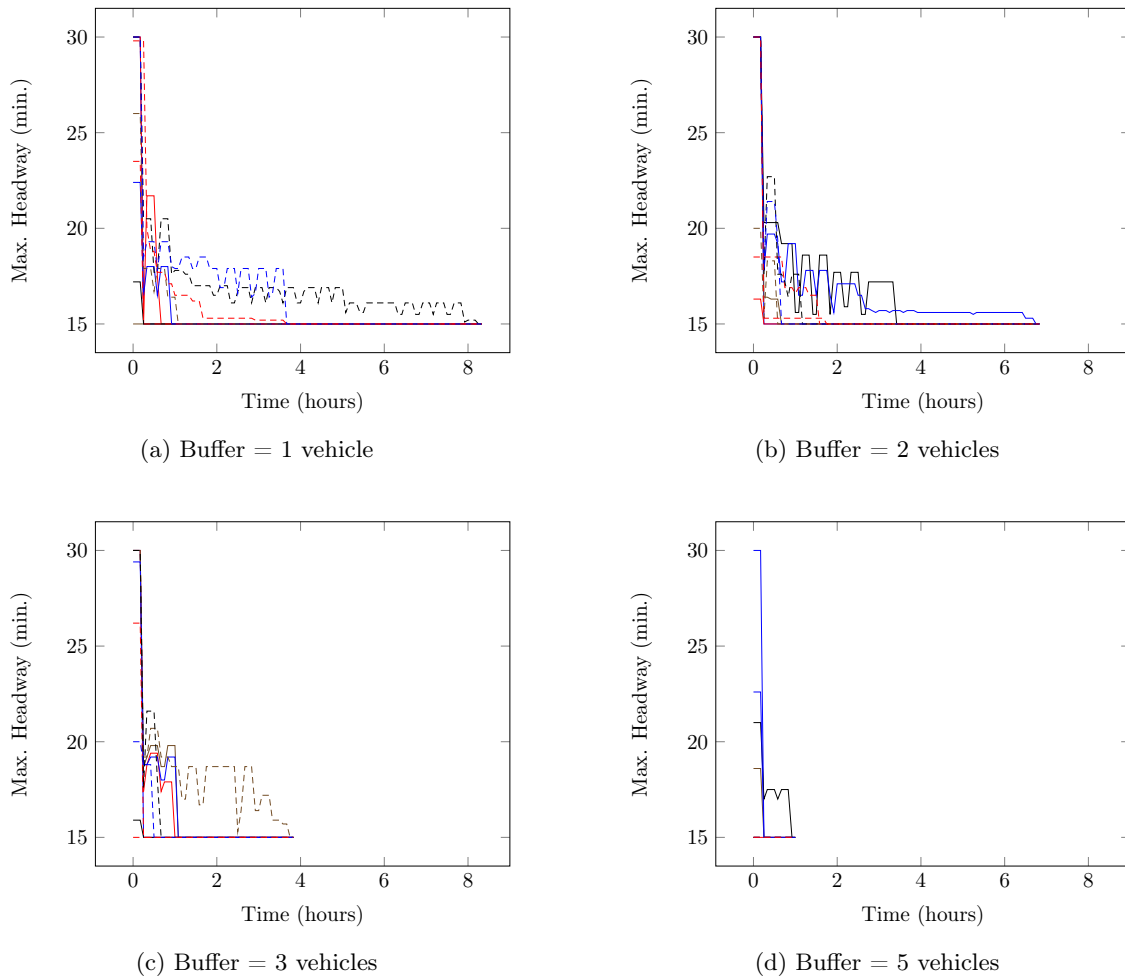


Figure 5: Maximum headway after a breakdown of a vehicle plotted over time, with a different number of vehicles on the network. The maximum headway is plotted for 10 replications. The target headway equals 15 minutes.

5.3 Different Target Headways

In the next experiment, we test the performance of the policy in case the lines in the system do not all have the same target headway. We conduct this experiment on a star network with three lines, with target headways of 10, 15 and 20 minutes, respectively. The travel time for each line is uniformly drawn between 10 and 30 minutes.

In Figures 6a-6b, the headways of the three lines are plotted over time for a randomly generated instance. Only the headways of the lines from the central station of the star network to the outer stations are included here. If the number of vehicles is equal to the minimum number required to meet the target headways, we observe that the system does not converge to a stable state where the target headways are always met. Instead, the system converges to a periodic motion where there are slight deviations from the target headways. In this periodic motion, the headways of the line with a target headway of 10 minutes vary between 10 and 13 minutes and for the line with a target headway of 20 minutes they vary between 20 and 23 minutes. On the other hand, if there is one more vehicle in the system, we can observe that the headways do all converge to the target headways. Hence, this indicates that despite the absence of theoretical guarantees, the policy still performs well, but that a larger number of vehicles may be required to ensure that the target headways are met at all times.

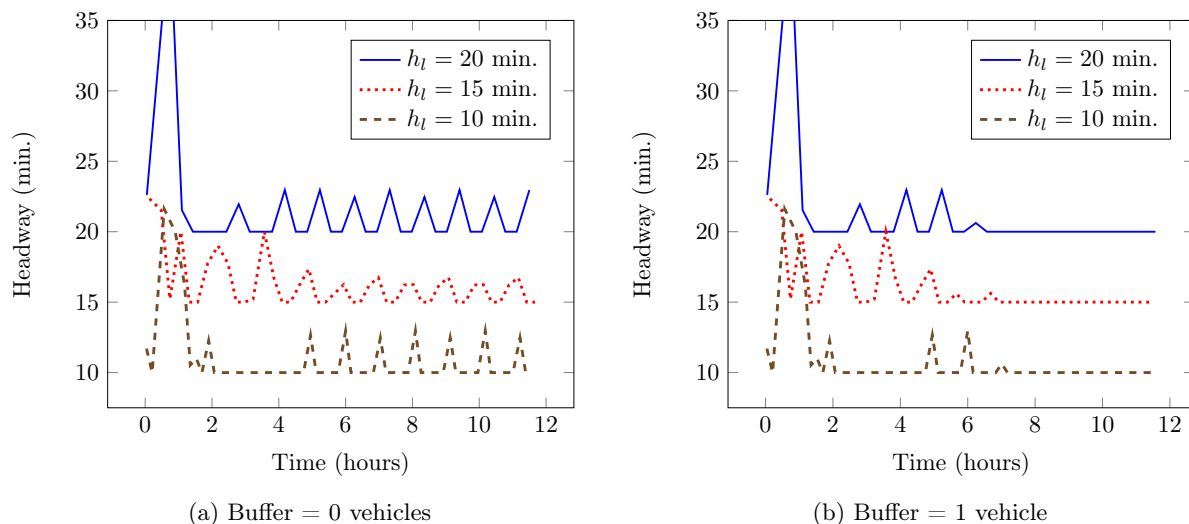


Figure 6: Headways of lines with different target headways, plotted over time, with and without a buffer.

5.4 Stochastic Travel Times

In a final experiment, we test the performance of the policy in case the travel times are not fixed, as we assumed in the theoretical analysis, but stochastic. We perform this experiment on a star network with five lines with a target headway of 15 minutes. The nominal travel time for each line is uniformly drawn between 10 and 30 minutes. The realized travel time is equal to the sum of the nominal travel time and a disturbance term. As it is likely that there is correlation in the duration of subsequent trips, we generate the disturbances ε^l for each line according to an autoregressive model:

$$\varepsilon_i^l = \rho \varepsilon_{i-1}^l + \eta_i, \quad \eta_i^l \sim N(0, \sigma^l).$$

We set $\rho = 0.8$ and $\sigma^l = \frac{1}{4}t_l$.

In Figure 7, the empirical cumulative probability density function of the headway is visualized for a randomly generated network. This function is presented for different number of buffers, which are computed based on the nominal travel times. As expected, the headways are not always equal to the target headway of 15 minutes as there are constant disturbances keeping the system away from a stable state. However, it can be observed that the headways are reasonably close to the target headway. Even without any buffer, over 50 percent of the headways are equal to the target headway and 80 percent of the headways are shorter than 20 minutes. When the buffer is larger, these numbers rapidly increase. Therefore, this suggests that the policy performs reasonably well if the travel times are stochastic, but that a larger number of vehicles is required to obtain a (very) high service level.

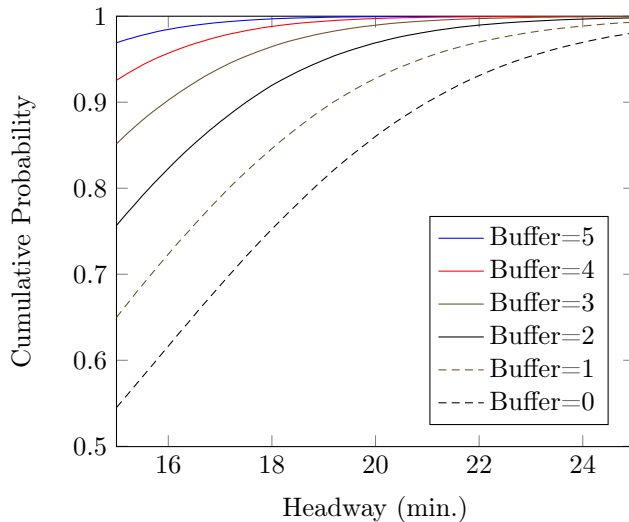


Figure 7: Empirical cumulative probability density function of the headway under stochastic travel times, with a different number of vehicles on the network. The target headway equals 15 minutes.

6 Conclusion

We proposed a self-organizing policy for dispatching vehicles in multi-line public transit systems. Theoretical and numerical analyses illustrate that our policy performs well. In idealized conditions and provided that a sufficient number of vehicles is available, it is guaranteed that the system converges to a stable state where the target headway of each line is met. In case travel times are not fixed but stochastic, or in case lines have different target headways, the deviations from the target headways are small, especially if there is some reserve capacity in the number of vehicles. Furthermore, our policy causes the system to quickly recover after disruptions, such as the breakdown of a vehicle.

Our promising theoretical and numerical results show that the potential of self-organizing strategies extends to multi-line public transit networks. Specifically urban high-frequency networks seem suited for our approach, as convergence is more rapidly established if the target headway and size of the network is smaller. Compared to schedule-based approaches, the self-organizing approach is much easier to implement, as it does not require constructing a schedule, monitoring adherence to the schedule and rescheduling after disruptions. Only the target headway needs to be set, which should be feasible with respect to the travel times of all lines and the number of available vehicles.

For further research, it would be interesting to investigate if the policy can be generalized to ensure that headways are always self-equalizing, even if $n < n^*$. Likely, this would require that the target headway is no longer exogenous, but emerges spontaneously due to the dynamics of the system. It is an open question whether this can be achieved by a simple decentralized policy, without coordination or communication between different parts of the network.

Acknowledgements This research was funded by NWO, the Netherlands Organisation for Scientific Research, as part of the research programme Complexity in Transport & Logistics (project number 439.16.111).

References

- Argote-Cabanero, J., Daganzo, C. F., & Lynn, J. W. (2015). Dynamic control of complex transit systems. *Transportation Research Part B: Methodological*, 81, 146–160.
- Bampas, E., Gašieniec, L., Hanusse, N., Ilcinkas, D., Klasing, R., & Kosowski, A. (2009). Euler tour lock-in problem in the rotor-router model. In *International Symposium on Distributed Computing* (pp. 423–435).
- Bampas, E., Gašieniec, L., Hanusse, N., Ilcinkas, D., Klasing, R., Kosowski, A., & Radzik, T. (2017). Robustness of the rotor–router mechanism. *Algorithmica*, 78(3), 869–895.

- Bartholdi, J. J., & Eisenstein, D. D. (2012). A self-coordinating bus route to resist bus bunching. *Transportation Research Part B: Methodological*, 46(4), 481–491.
- Chalopin, J., Das, S., Gawrychowski, P., Kosowski, A., Labourel, A., & Uznański, P. (2015). Limit behavior of the multi-agent rotor-router system. In *International Symposium on Distributed Computing* (pp. 123–139).
- Dekker, M. M., Van Lieshout, R. N., Ball, R. C., Bouman, P. C., Dekker, S. C., Dijkstra, H. A., . . . Van den Akker, J. M. (2018). A next step in disruption management: Combining operations research and complexity science. In *Proceedings of the Conference on Advanced Systems in Public Transport*.
- Dereniowski, D., Kosowski, A., Pająk, D., & Uznański, P. (2016). Bounds on the cover time of parallel rotor walks. *Journal of Computer and System Sciences*, 82(5), 802–816.
- Hernández, D., Muñoz, J. C., Giesen, R., & Delgado, F. (2015). Analysis of real-time control strategies in a corridor with multiple bus services. *Transportation Research Part B: Methodological*, 78, 83–105.
- Liang, S., Zhao, S., Lu, C., & Ma, M. (2016). A self-adaptive method to equalize headways: Numerical analysis and comparison. *Transportation Research Part B: Methodological*, 87, 33–43.
- Lovász, L. (1993). Random walks on graphs: A survey. *Combinatorics, Paul Erdős is Eighty*, 2(1), 1–46.
- Petit, A., Lei, C., & Ouyang, Y. (2019). Multiline bus bunching control via vehicle substitution. *Transportation Research Part B: Methodological*, 126, 68–86.
- Priezzhev, V. B., Dhar, D., Dhar, A., & Krishnamurthy, S. (1996). Eulerian walkers as a model of self-organized criticality. *Physical Review Letters*, 77(25), 5079.
- Van Lieshout, R. N. (2019). *Integrated periodic timetabling and vehicle circulation scheduling* (Tech. Rep.). Econometric Institute Report Series EI2019-27.
- Van Lieshout, R. N., Bouman, P. C., & Huisman, D. (2020). Determining and evaluating alternative line plans in out-of-control situations. *Transportation Science*, 54(3), 740–761.
- Wagner, I. A., Lindenbaum, M., & Bruckstein, A. M. (1999). Distributed covering by ant-robots using evaporating traces. *IEEE Transactions on Robotics and Automation*, 15(5), 918–933.
- Yanovski, V., Wagner, I. A., & Bruckstein, A. M. (2003). A distributed ant algorithm for efficiently patrolling a network. *Algorithmica*, 37(3), 165–186.
- Zhang, S., & Lo, H. K. (2018). Two-way-looking self-equalizing headway control for bus operations. *Transportation Research Part B: Methodological*, 110, 280–301.