INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2008; **18**:165–195
Published online 12 April 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rnc.1192

Decentralized redistribution for cooperative patrol

Brandon J. Moore*,† and Kevin M. Passino

Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, U.S.A.

SUMMARY

This paper addresses the problem of enabling a group of autonomous vehicles to effectively patrol an environment significantly larger than their communication and sensing radii. Our formulation uses an *a priori* spatial decomposition of the environment into smaller areas in order to provide a framework for the allocation of vehicles to different parts of the environment. We develop a distributed cooperative control algorithm that transfers vehicles between these areas based on only local information and prove that it achieves the proper environment-wide distribution of vehicles within a finite time interval. Various applications are discussed and simulations are included to illustrate convergence dynamics as well as to quantify practical performance as a function of problem parameters. Copyright © 2007 John Wiley & Sons, Ltd.

Received 12 April 2006; Revised 20 November 2006; Accepted 21 February 2007

KEY WORDS: cooperative control; resource balancing; discrete algorithms

1. INTRODUCTION

This work is motivated by a mission scenario in which a group of autonomous vehicles is tasked to cooperatively patrol a relatively large environment (e.g. a region of land whose dimensions dramatically exceed the vehicles' maximum communication and sensing radii). The reason for assuming such a small communication radius stems from the proposed deployment of autonomous micro-vehicles whose transmission power may be limited by the capability of their hardware, or is deliberately limited in order to conserve energy. Alternatively, even a full-sized vehicle may voluntarily limit its transmission power in order to avoid broadcasting its position

*Correspondence to: Brandon J. Moore, Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, U.S.A.
† E-mail: mooreb@ece.osu.edu

Contract/grant sponsor: AFRL/VA
Contract/grant sponsor: AFOSR Collaborative Center of Control Science; contract/grant number: F33615-01-2-3154
Contract/grant sponsor: DAGSI Fellowship

to enemy forces trying to find and destroy it (or trying to avoid that vehicle finding and destroying them). The basis for the assumption of a limited sensing radius will depend on what kind of sensing the vehicles perform. If, for example, they are trying to detect enemy radio signals or are using radar, then the same hardware constraints that limited the vehicles' communication will impede their sensing range as well. On the other hand, if the vehicles are taking visible light or infrared images, then the pixel resolution of those images will naturally limit the effective sensing range (not to mention the limitations imposed by physical obstructions in the environment such as trees or hills). In our mission scenario we also assume that the group must maintain a continuous presence in the environment in order to provide effective detection and monitoring of potential targets or other objects of interest. This is in contrast to the type of mission scenarios in which the vehicle would only need to look at each point in the environment once. For example, if vehicles are looking for targets that can move or hide, then a single search of the environment could miss them. Also, the purpose of many patrol missions is to protect against enemy infiltration and so the environment (or at least its borders) must be continually monitored.

A scenario of the type just described imposes two somewhat competing requirements on the vehicle group. The first requirement is one of *dispersion*. In order to avoid large gaps in their coverage of the area, it is necessary for the vehicles to spread themselves throughout the entire environment. If, for instance, the group chose to move in a tight formation through the environment (e.g. a 'mowing the lawn' approach as in [1]), then a mobile target might easily evade this formation by timing its movements correctly or simply by chance. Since the environment is very large, keeping the group in a tight formation increases the distance a vehicle would have to travel to get to any spot in the environment, whereas dispersing the group will mean that at least one vehicle should be able to get to any particular point in a relatively short amount of time and thus no part of the environment will go very long without being observed. Also, a large formation of vehicles generates more noise, a larger thermal or radar signature, etc. and so targets that can move or hide might find it much easier to evade such a formation than a single vehicle.

The second requirement imposed by our scenario is one of *cohesion* and is a product of the range-limited nature of the inter-vehicle communications. In order to effectively patrol the environment it is necessary for the group to cooperate, and that means they must, at least occasionally, meet up in order to communicate. If the group simply disperses without any coordination strategy, then we are faced with two obvious problems. The first is that a vehicle will not have any way to pass on information about targets it has found, thus rendering that information and the entire mission useless. The second is that without some sort of feedback mechanism it will be impossible to maintain the proper dispersion of agents. For instance, if a vehicle is destroyed, malfunctions, or needs to refuel then, without feedback, the gap in coverage that this creates will not be filled. Alternatively, if more vehicles are added to the mission, the group will have to redistribute itself in order to properly accommodate the new arrivals. It may also be that the proper distribution of vehicles in the environment may change from time to time (e.g. if the relative importance of patrolling certain parts of the environment shifts), and so the group may have to respond to that as well.

The approach to this problem that we take in this paper is as follows. We first decompose the environment into a number of smaller *areas* to which individual vehicles are dynamically assigned (see Figure 1). The idea behind this spatial decomposition is that by restricting a vehicle to a small section of the environment, it should be easier to determine how that vehicle should

DOI: 10.1002/rnc

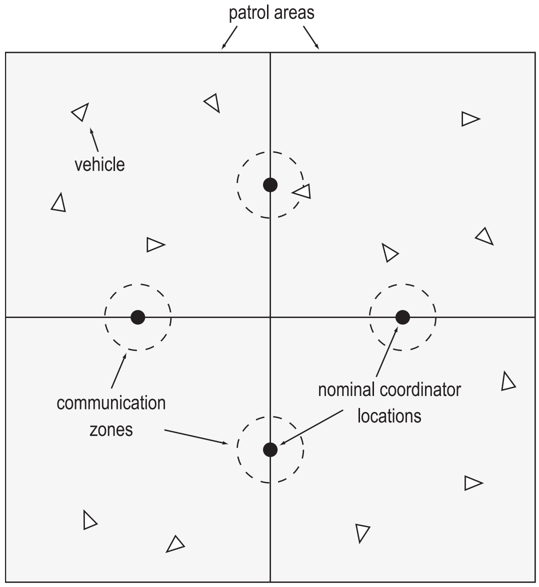

Figure 1. Example environment decomposed into four areas.

behave in order to efficiently patrol that area. More specifically, if we provide one or more locations in an area to which a vehicle can go to in order to communicate with another vehicle, then the small size of the area means that a vehicle can patrol a significant portion of that area in between visits to those locations and still communicate on a fairly regular schedule. That is to say that a vehicle can spend most of its time actually patrolling the area and still remain connected to the group through periodic communication. Of course, in order for a vehicle to communicate by going to one of these special locations, there must be another vehicle there. This brings us to the second part of our approach which is the establishment of a number of *coordinators*. Coordinators are simply a small number of the group's vehicles that loiter in places where two or more areas meet and provide the previously mentioned 'locations' where the other vehicles can come to communicate. Exactly where a coordinator is located depends on the details of the specific scenario. For example, if the vehicles are fixed-wing aircraft, then they obviously cannot stay in exactly one spot. In this case, a coordinator could fly a periodic trajectory in relation to some nominal location and then the other vehicles could fly backwards along this trajectory and eventually make contact with the coordinator. If the coordinator is a ground vehicle or rotary aircraft then it could remain in one spot, but since this increases its chances of being discovered and destroyed by enemy forces, it could periodically relocate to another location. As long as the number of these locations is fixed and the coordinator is guaranteed to visit each of them on a regular basis, then a vehicle seeking to communicate with it need only go to one of those locations and wait for the coordinator to show up. We note that if the number of vehicles in the group is small, we may not want to dedicate any of them to serve in the role of a coordinator. (Even though the coordinator vehicles could conceivably perform other useful tasks in addition to acting as a communication relay, the fact that they are closely tied to a specific location will reduce their effectiveness in doing so.) In this case there is

DOI: 10.1002/rnc

the possibility of using some sort of other device to fulfil the need for coordinators. Depending on how much information we need them to store and process, such a device could potentially be fairly small, and thus easily hid somewhere in the environment. A small device could also be relatively disposable and simply replaced if destroyed (although whatever information it currently held could not).

The preceding formulation of the problem in terms of areas and coordinators divides the group of vehicles into a two-level hierarchy. At the bottom level we have the majority of the vehicles, and from this point on we will simply use ‘vehicle’ and ‘vehicles’ to refer to this portion of the group. Being confined to one area at a time the vehicles’ main job is simply to gather information from that limited region and report it back to the coordinators. Forming the upper level, the coordinators are responsible for directing the vehicles’ efforts within the area (i.e. telling them where to go look if a certain part of the area has something interesting in it or has not been observed for a while), and, most importantly, shifting vehicles between the areas they oversee (i.e. the ones they ‘connect’ by virtue of their nominal location) in order to achieve the proper distribution of vehicles across all the areas in the environment. For now we consider the proper distribution of vehicles to be one in which the number of vehicles in each area is *globally balanced*. By this we mean that the number of vehicles in any area differs from that of any other area by no more than one. The reasoning behind our desire for a globally balanced distribution is that unless we have specific information that leads us to believe it is better to concentrate more vehicles in certain regions, we want to spread them as uniformly as possible across the environment to reduce the gaps in coverage. When the areas are all the same size (as would be the case with a regular gridding of the environment), then this corresponds to wanting the number of vehicles in each area to be as equal as possible (and unless the number of vehicles is evenly divisible by the number of areas, then the best we can do is to have any two areas differ by no more than one vehicle).

Another important responsibility of a coordinator is collecting the information from the vehicles and passing it on to other systems (e.g. to other coordinators via the vehicles) and, since the coordinators will not usually be given the authority to make major decisions without human validation (e.g. attacking targets), they will also have to get information back to (and receive instructions from) a human operator. Because of the limited communication radius of the vehicles, the easiest way to do this would simply be to include the human operator as one of the coordinators of the system. If the system is a long distance from any available operator, however, then extra measures must be taken in order to ensure that that operator receives information on either a periodic or as-needed basis (e.g. sending a vehicle back to the operator’s location or at least close enough to for it to communicate with him).

Other work in cooperative control that shares some similarities with our scenario includes formulations that deal with ‘pop-up’ targets [2, 3] and those that deal with the spatial allocation of resources [4]. However, the most closely related work to ours is on load balancing in distributed computing [5] (with the vehicles viewed as a finite number of discrete load blocks and the areas as the processors) with a few key exceptions. In our mission scenario, decisions to move vehicles from a given area to a neighbouring one are made in a distributed manner at the interconnections between the two areas (i.e. by the coordinators). Essentially, the coordinators for an area must share the vehicles in that area which is in contrast to the traditional load-balancing formulation in which one processor has total control over one (and only one) collection of load blocks. Also, in discrete load-balancing systems, a locally balanced state is all that is guaranteed, so neighbouring processors may differ by as much as the size of the largest

DOI: 10.1002/rnc

load block for both the delay and non-delay cases [6,7] and so the global imbalance grows with the diameter of the network. While such imbalances may be acceptable in a system with a large total load, since the ratio of vehicles to areas may be fairly small in the scenarios proposed above, it is important to achieve a globally balanced state as discussed. With few limited exceptions, the issue of achieving a globally balanced load distribution has received very limited attention in the literature prior to this work. The message passing algorithm presented in [8] is able to achieve a state in which every processor's set of neighbours is balanced within the largest block size, but is still not able to guarantee that that will result in a globally balanced state. It is possible to reduce the expected size of the global imbalance by employing non-deterministic algorithms such as the one presented in [9], although these types of formulations do not preclude pathological cases that keep the system from balancing and also may result in a balanced system becoming unbalanced. The algorithm presented in [10], which shares some commonalities with this work, is able to achieve a globally balanced distribution but is specific to applications involving a regular grid of parallel processors and does not address the delays, asynchronism, and distributed calculation present in our model. A final difference between the usual load-balancing framework and our problem is that in traditional load-balancing systems there is the desire that the occurrence and volume of load transfer events should diminish as load imbalances become small. In the algorithm we present, however, whenever the number of vehicles is not evenly divisible by the number of areas, the 'excess' vehicles will continue to transition between different areas so that some sort of average coverage is achieved on each one (a desirable characteristic for our scenario).

2. DISTRIBUTED COOPERATIVE PATROL ALGORITHM

In this section we discuss the requirements we will impose on the interconnections between the areas and coordinators, how the coordinators can estimate the number of vehicles in an area, and our model and control algorithm.

2.1. Area–coordinator interconnection structure

The *area–coordinator interconnection* is best described by a bipartite graph in which the areas form one set of nodes and the coordinators form the other set. In this graph a coordinator node is connected only to the nodes of the areas for which it is responsible (see Figure 2 for an example). Although we would like to consider the widest class of interconnections possible, in this work we primarily limit ourselves to the specific topology of a unidirectional ring (with line topologies considered as an extension). In this topology there is an equal number of area and coordinator nodes, which alternate in sequence to form a ring comprising all of the nodes in the graph. The ring is unidirectional in that we will only allow the coordinators to transfer vehicles around the ring in one direction (i.e. each coordinator is limited to moving vehicles from one specific area to another specific area). Without loss of generality we will assume that the areas are numbered in sequence around the ring and for convenience we will associate each coordinator with the area from which it may remove vehicles (see Figure 3). Using the terms $i+1$ and $i-1$ to denote the areas after and before area i on the ring (and $i+2$ being the second area after area i, etc.), we have that coordinator i is restricted to moving vehicles from area i to area $i+1$ exclusively.

DOI: 10.1002/rnc

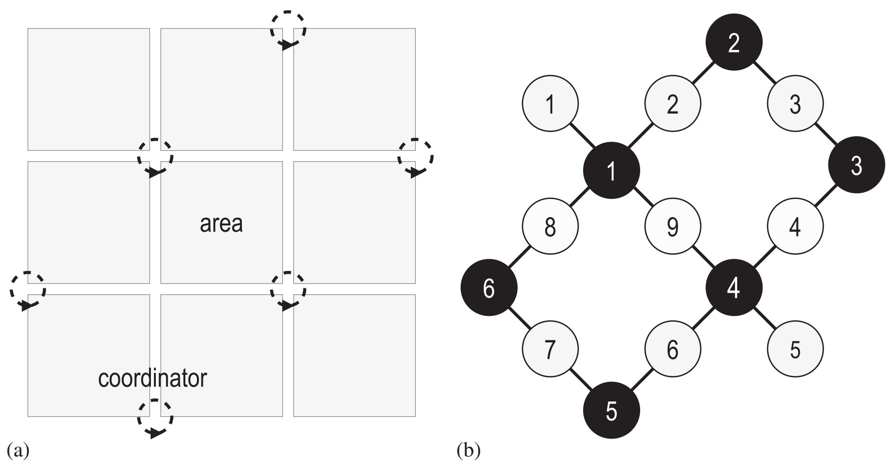

Figure 2. A example of an area–coordinator interconnection (a) and its representation as a bipartite graph (b) with the areas as grey nodes and the coordinators as black nodes.

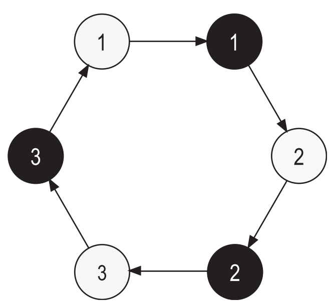

Figure 3. A three area/coordinator unidirectional ring.

Although we have limited our interconnection graph to a unidirectional ring, we need to note that this does not limit us to area–coordinator interconnections with ring-like spatial layouts. For instance, the three by three grid of areas in Figure 2 can be turned into a unidirectional ring (following the numeric labelling of the area nodes in Figure 2(b)) by simply restricting the manner in which the coordinators may move vehicles. For example, we would only let coordinator 4 move resources from area 4 to area 5 and from area 5 to area 6, and force it to ignore its connection to area 9 altogether. This sort of methodology has two clear advantages. First, we use far fewer coordinators than we would if we insisted on having a unique coordinator for every area (a third fewer in the example of Figure 2). Secondly, restricting the way in which a coordinator may transfer vehicles does not in any way interfere with its role as a collector and distributor of information (i.e. in the above example coordinator 4 may still help direct the patrolling of vehicles within area 9 and serve to pass along any interesting information to coordinators 3 and 5 faster than would happen if that information had to follow the same unidirectional ring as the vehicles). From this discussion it should be clear that the general shape

DOI: 10.1002/rnc

of a ring can be 'bent' in order to match the physical layout of the environment. Of course, in the design of area–coordinator connections we need to make sure that we avoid certain tree-like structures (as well as some less obvious topologies) because of the need to have at least one possible ring present in the underlying graph.

2.2. Estimation methods

Due to the range-limited nature of the communication between the vehicles and coordinators and the fact that more than one coordinator may alter the number of vehicles in an area, a coordinator for an area will not immediately be aware that some vehicles have been added or removed from that area and thus will have, in general, an inaccurate perception of the number of vehicles currently present in that area. It is important, therefore, for each coordinator to have an intelligent way of estimating the true number of vehicles in an area, preferably in a way that preserves some sort of useful relationship between the estimate and the true value. In what follows we present a way to do this in light of the restricted way vehicles may be transferred around our unidirectional ring interconnection.

The first requirement of our estimation process is that each vehicle spends no more than a bounded length of time in an area between the times that it 'checks in' with a specific coordinator by moving to that coordinator's communication zone and letting that coordinator know that it (the vehicle) is, or still is, in the area. To be more precise, if a vehicle has checked in with a coordinator at time t, then it has until time $t+D$ to check in again, where $D>0$ is a preset length of time known to both the vehicles and coordinators. Also, when a vehicle is first moved into an area at time t it has until time $t+D$ to check in with all the other coordinators for that area. In general the bound on the maximum time between check ins, D, is determined by the size of the areas and the vehicles' speed. This bound should be generous enough to account for the facts that a vehicle has to check in with all the other coordinators for the area, may encounter unexpected delays in travel (e.g. obstacles for ground vehicles or headwinds for air vehicles), and should spend most of its time patrolling the area (i.e. not simply travelling between coordinators in order to meet an over demanding check-in schedule).

One of the purposes of having the vehicles check in with the coordinators for an area on a regular basis is to allow those coordinators to maintain a fairly accurate list of the vehicles in that area, which is accomplished in the following manner. Whenever a coordinator moves an extra vehicle into an area it gives it a unique identification number that has never been used before. Due to the restriction of the unidirectional ring which limits coordinator i to adding vehicles to area $i+1$, the simplest way to do this is for coordinator i to give each vehicle it adds to area $i+1$ a number with three fields: the designation of the area $(i+1)$, the time it was moved to that area, and one other number to distinguish between vehicles that may have been added at the same time (e.g. the vehicle's serial number). In the event that a vehicle is found in the area without being moved there (i.e. when the system is initially set up or more vehicles are added to system from outside), then that vehicle is identified in the same way. If we want to let coordinator $i+1$ label vehicles found in area $i+1$ (instead of only letting coordinator i do it) then we will have to add another field to the vehicle's identification number for finding the coordinator's designation in order to prevent duplication. Once the vehicles have identification numbers, the coordinators can each keep a list of the vehicles it thinks are in that area. Whenever a coordinator is visited by a vehicle not on its list (or moves a vehicle into the area), it adds it to the list. Whenever a coordinator has not been visited by a vehicle on its list for a time

DOI: 10.1002/rnc

period longer than D (or when it removes a vehicle itself), it assumes that that vehicle has been removed from the area and removes it from its list (and we again see the importance of using a generous bound D). A coordinator's estimation of the number of vehicles in an area is then simply the length of its corresponding list.

In general, we would expect a coordinator's list to be inaccurate for two reasons: first, because there may be vehicles recently added to the area that have not checked in, and second, because there may be vehicles that have been removed from the area for which not enough time has expired for the coordinator to cross them off its list. However, by restricting ourselves to the unidirectional ring interconnection we can do better than this. If we assume that no vehicles enter or leave the system after some initial time (a common assumption in problems of this type [5–7]), then we know that after a time period of length D that the following must be true: first, coordinator i's estimate of the number of vehicles in area i will be less than or equal to the true number because it is the only coordinator that may remove vehicles from that area. Second, coordinator i's estimate of the number of vehicles in area $i+1$ will be greater than or equal to the true number because it is the only coordinator that may add vehicles to that area. Thirdly, if coordinator i does not move any vehicles from area i to area $i+1$ from time t to time $t+D$, then its estimate for area i is greater than or equal to the true number in area i at time t and its estimate for area $i+1$ is less than or equal to the true number in area $i+1$ at time t.

We note that with the unidirectional ring all of the above assertions would still be true if instead of editing their lists when vehicles visited to check in (or failed to do so), the coordinator adding (removing) vehicles to (from) an area simply informed the other coordinator over a communication channel with some delay. Although in the general case physically checking in with a coordinator has the advantage of giving that coordinator a period of exclusive control to move it to another area (i.e. it solves the problem of two coordinators trying to move the same vehicle simultaneously), this is not necessary for the case of the unidirectional ring because only one coordinator per area has the authority to remove vehicles. Thus, we can see that our range-limited communication scenario is almost equivalent to a scenario in which communication is confined to a certain topology (i.e. only coordinators that share an area are directly connected) and subject to substantial delays.

2.3. Model and distributed control strategy

In this section we will model the upper level of our system hierarchy, namely the method by which the coordinators will achieve a globally balanced distribution of vehicles in the areas despite their limited interconnections and the inaccuracies in estimation resulting from delays. To create a mathematical representation of the system we will use a discrete event system model of the same form used in [11]. The state of the system will be updated asynchronously at discrete time instants according to the occurrence of particular events describing the relevant behaviour of the system. The sequence of events is non-deterministic but is constrained by the state of the system at each time step as well as by casual links between pairs of events.

The state of the complete system will be composed of those variables describing the 'plant' (i.e. the number of vehicles in each area) and those describing the distributed controller (i.e. the coordinators' estimates of the number of vehicles in the areas they connect). Let N be the total number of areas in the ring and let $\mathcal{N} = \{1, \ldots, N\}$ be the set of these areas. For the plant, each area $i \in \mathcal{N}$ has an associated variable $x_i \in \mathbb{N}$ whose value is equal to the number of vehicles in that area (where we take the definition of the natural numbers to be $\mathbb{N} = \{0, 1, 2, \ldots\}$,

DOI: 10.1002/rnc

i.e. the non-negative integers as opposed to just the positive integers as is sometimes seen). Let $x_P = [x_1, \ldots, x_N]^\top$ be the entire state of the plant. For the controller, coordinator i (the coordinator that moves vehicles from area i to area $i+1$ has a (possibly outdated) estimate of the true values of x_i and x_{i+1}. Let these estimates be denoted by $\tilde{x}_i^i$ and $\tilde{x}_{i+1}^i$, respectively, and let $x_C = [\tilde{x}_1^1, \tilde{x}_2^1, \ldots, \tilde{x}_N^N, \tilde{x}_1^N]^\top$ denote the entire state of the controller. The entire state of the system is then $x = [x_P^\top, x_C^\top]^\top$ and we let the state space be denoted by $\mathscr{X} = \mathbb{N}^{3N}$. Let $x(k) \in \mathscr{X}$ denote the state of the system at logical time index k (with similar notation for its components).

We first present two subsets of this state space that reflect some of our assumptions about the system. The first,

$$\mathscr{X}_L = \left\{ x \in \mathscr{X} : \sum_{i=1}^{N} x_i = L \right\}$$

with $L = \sum_{i=1}^{N} x_i(0)$, is actually a family of subsets parameterized by the initial condition of the plant $x_P(0)$. As discussed in Section 2.2, one of our primary assumptions will be that no vehicles enter or leave the system during its operation. This assumption is equivalent to the statement that $x(k) \in \mathscr{X}_L$ for all $k \geqslant 0$. The second subset of interest

$$\mathscr{X}_E = \{ x \in \mathscr{X} : \tilde{x}_i^{i-1} \geqslant x_i \geqslant \tilde{x}_i^i \ \forall i \in \mathscr{N} \}$$

defines the condition wherein the coordinator that may add vehicles to area i does not underestimate the true number and the coordinator that may remove vehicles from area i does not overestimate the true number. By assuming that $x(k) \in \mathscr{X}_E$ for all $k \geqslant 0$, we will be able to greatly simplify our model and later analysis. The justification for such an assumption has already been discussed in Section 2.2. For convenience, let $\mathscr{X}_0 = \mathscr{X}_L \cap \mathscr{X}_E$ be the intersection of the these first two subsets, giving us $x(k) \in \mathscr{X}_0$ for all k.

There are two types of events of interest for our system, namely, the transfer of vehicles from one area to the next and a change in one of a coordinator's estimates (i.e. when it recognizes that a vehicle has been added to or removed from a particular area). Let $e_i^{\alpha(i)}$ denote the 'partial' event of coordinator i transferring $\alpha(i) \in \mathbb{N}$ vehicles from area i to area $i+1$. By the logic underlying our assumption that the initial state lies in $\mathscr{X}_0$, we need only consider estimate changes for coordinator i that either increase its estimate $\tilde{x}_i^i$ or decrease its estimate $\tilde{x}_{i+1}^i$. Therefore, let $e_i^{\beta(i)}$ denote that coordinator i has detected $\beta(i) \in \mathbb{N}$ vehicles in area i that it has not seen before. Similarly, let $e_i^{\gamma(i)}$ denote that coordinator i has noted the absence of $\gamma(i) \in \mathbb{N}$ vehicles in area $i+1$ (i.e. has not seen those vehicles for a length of time D and can safely assume they have been previously removed by coordinator $i+1$).

Let the total event space $\mathscr{E}$ be defined as the union of the following sets minus the null set $\emptyset$

$$\mathscr{P}(\{e_i^{\alpha(i)} : \alpha(i) \in \mathbb{N},\ i \in \mathscr{N}\}) \tag{1}$$

$$\mathscr{P}(\{e_i^{\beta(i)} : \beta(i) \in \mathbb{N},\ i \in \mathscr{N}\}) \tag{2}$$

$$\mathscr{P}(\{e_i^{\gamma(i)} : \gamma(i) \in \mathbb{N},\ i \in \mathscr{N}\}) \tag{3}$$

where $\mathscr{P}(\cdot)$ is the power set of its argument. Thus, we can use $e(k) \in \mathscr{E}$ to denote the combination of partial events (i.e. transfers of vehicles and estimate updates) that occur simultaneously at time k.

Which events may actually occur will depend on the state of the system; specifically, a set-valued enable function $g : \mathcal{X} \rightarrow \mathcal{E}$ is defined as follows: if $e(k) \in g(x)$, then

(a) for each $e_i^{\alpha(i)} \in e(k)$ the following conditions must hold:

(i) $\alpha(i) = 0$ if and only if $\tilde{x}_i^i(k) \leqslant \tilde{x}_{i+1}^i(k)$,
(ii) if $\tilde{x}_i^i(k) > \tilde{x}_{i+1}^i$, then $1 \leqslant \alpha(i) \leqslant \lceil (\tilde{x}_i^i(k) - \tilde{x}_{i+1}^i(k))/2 \rceil$,
(iii) $e_i^{\alpha'(i)} \notin e(k)$ if $\alpha(i) \neq \alpha'(i)$.

Condition (i) prevents coordinator i from transferring vehicles from area i to area $i+1$ unless it thinks that area i has more vehicles than area $i+1$. Condition (ii) uses the ceiling operator $\lceil \cdot \rceil$ (i.e. the smallest integer greater than or equal to its argument) and ensures that coordinator i attempts to balance the number of vehicles in areas i and $i+1$ by passing at least one vehicle from the former to the latter, but not more than would make the number of vehicles in area i less than one below area $i+1$ after the transfer (this is more aggressive than most load-balancing schemes which stipulate that a processor should not pass an amount of load that would make it less lightly loaded than its neighbours [5, 6]). We note that the assumption $x(k) \in \mathcal{X}_0$ and these first two conditions prevent a coordinator from transferring more vehicles than those that actually exist in area i. Condition (iii) simply limits coordinator i to a single transfer of vehicles per composite event $e(k)$.

(b) for each $e_i^{\beta(i)}$ and $e_i^{\gamma(i)}$ in $e(k)$ it must be the case that $\beta(i) \leqslant x_i(k) - x_i^i(k)$ and $\gamma(i) \leqslant x_{i+1}^i(k) - x_{i+1}(k)$ or else the state of the controller might not satisfy the condition of $\mathcal{X}_E$ at the next time index (which is physically impossible under our controlling assumptions). It is also necessary that both $e_i^{\beta'(i)} \notin e(k)$ if $\beta(i) \neq \beta'(i)$ and $e_i^{\gamma'(i)} \notin e(k)$ if $\gamma(i) \neq \gamma'(i)$ for the same reason as condition (iii) in part (a).

Now, at each time index k some event $e(k) \in g(x(k))$ occurs and alters the state of the system according to an update function $x_{k+1} = f(x(k), e(k))$ defined by

$$x_i(k+1) = x_i(k) - \alpha(i) + \alpha(i - i) \tag{4}$$

$$\tilde{x}_i^i(k+1) = \tilde{x}_i^i(k) - \alpha(i) + \beta(i) \tag{5}$$

$$\tilde{x}_{i+1}^i(k+1) = \tilde{x}_{i+1}^i(k) + \alpha(i) - \gamma(i) \tag{6}$$

where the indexed α, β, and γ values here on are taken from the partial events in $e(k)$ (and we let $\alpha(i)$, $\beta(i)$, or $\gamma(i)$ equal zero if coordinator i does not have a corresponding partial event). Simply put, the function f updates each $x_i \in x_{\mathrm{P}}$ by subtracting the vehicles transferred out of area i by coordinator i and adding those transferred into it by coordinator $i-1$. It also updates a coordinator's estimates according to that coordinator's actions and new information it has received.

Let $\mathcal{E}^{\mathbb{N}}$ be the set of all event trajectories. The set of valid event trajectories $E_V \subset \mathcal{E}^{\mathbb{N}}$ contains all event trajectories $E = [e(0), e(1), \ldots]$ such that there exists a state trajectory $X = [x(0), x(1), \ldots] \in \mathcal{X}^{\mathbb{N}}$ satisfying $e(k) \in g(x(k))$ for all $k \in \mathbb{N}$. Whereas the enable function captures the dynamics of this system from one time step to the next, we will need to define a set of

DOI: 10.1002/rnc

allowed trajectories in order to fully describe its behaviour over time. This subset of E_V, denoted E_B, consists of all event trajectories that meet the following conditions:

(1) There exists a positive integer B such that for every event trajectory $E \in E_B$ and for any time index k, the series of events $e(k), e(k+1), \ldots, e(k+B-1)$ contains at least one occurrence of the partial event $e_i^{\alpha(i)}$ for all $i \in \mathcal{N}$ if $\tilde{x}_i^i(k') > 0$ for all $k' \in \{k, \ldots, k+B-1\}$. This is equivalent to saying that whenever coordinator i thinks that there are vehicles in area i (which implies that there actually are vehicles in that area) then it must try to balance the number of vehicles in area i and area $i+1$ within B events. Because coordinators can only communicate with a vehicle when they check in, that is also the only time they can decide to transfer that vehicle. Therefore, the parameter B must be the maximum number of events that can occur within the maximum length of time a vehicle can go without checking in with a coordinator.

(2) For the same constant B as above, it must be the case for every event trajectory $E \in E_B$, any time index k, and every $i \in \mathcal{N}$, that if the series of events $e(k), e(k+1), \ldots, e(k+B-1)$ does not contain a partial event $e_i^{\alpha(i)}$ with $\alpha(i) > 0$, then we must have $\tilde{x}_i^i(k+B) \geqslant x_i(k)$ and $\tilde{x}_{i+1}^i(k+B) \leqslant x_{i+1}(k)$. This is simply a mathematical restatement of the last estimation property discussed in Section 2.2, and B is again the maximum number of events that can occur in the maximum length of time D that a vehicle can go without having checked in with a particular coordinator.

Based on the conditions laid out above for the set of allowed trajectories, there is a need for us to show that the number of events that may occur in a finite period of time is indeed finite. To do this we note that each event index k corresponds to a vehicle checking in with a coordinator. Since the physical act of checking in involves both communication and computation, it must take a certain amount of time which can be bounded from below by a positive constant. In reality we will not want the vehicles to repeatedly check in with the coordinators as they have other work to do (i.e. patrolling the area they are in), and so this lower bound is apt to be relatively large. In any case, the number of events a single vehicle may trigger in any time period of length D has an upper bound, and since there are a finite number of vehicles in the system, the total number of events that can be triggered in that same time period is also bounded.

3. CONVERGENCE RESULTS

3.1. Distribution dynamics and convergence time bound

In this section we analyse the system model of Section 2.3. We will show that the set

$$\mathcal{X}_I = \{x \in \mathcal{X}_0 : |x_i - x_j| \leqslant 1 \text{ for all } i, j \in \mathcal{N}\}$$

(representing the subset of $\mathcal{X}_0$ for which the number of vehicles in any area differs from that of any other area by no more than one, i.e. the desired globally balanced distribution) is invariant and that for any initial condition $x(0) \in \mathcal{X}_0$ the state of the system converges to $\mathcal{X}_I$ in a finite length of time that can be bounded from above by a function of the model parameters (specifically, the number of areas N, the number of vehicles in the system L, and the magnitude of delays B). We include here the important lemmas and the final convergence theorem and present their proofs in the Appendix.

DOI: 10.1002/rnc

Lemma 1
The set $\mathscr{X}_I$ is invariant with respect to the system model in Section 2.3.

The proof of the above lemma shows that trajectories within the invariant set $\mathscr{X}_I$ are not, in general, static. When the total number of vehicles L, is not evenly divisible by the number of areas N, the excess vehicles are continually transferred around the ring so that the number of vehicles at each node alternates between two consecutive integers $\lfloor L/N \rfloor$ and $\lfloor L/N \rfloor + 1$ (where $\lfloor \cdot \rfloor$ is the floor operator, i.e. the greater integer less than or equal to its argument). When L is divisible by N, however, it is easily shown that the only possible state within $\mathscr{X}_I$ is the perfectly balanced state $x_i = x_j$ for all $i, j \in \mathscr{N}$.

Lemma 2
For the system model described in Section 2.3, the minimum number of vehicles in any area $m(k) = \min_{i \in \mathscr{N}} x_i(k)$ is non-decreasing in time. Similarly, the maximum number of vehicles in any area $M(k) = \max_{i \in \mathscr{N}} x_i(k)$ is non-increasing in time.

The above lemma shows that our system model is 'well-behaved' in that the global imbalance in the system, as measured by $M(k) - m(k)$, is non-increasing with time. The essence of the following lemma and theorem is to show that this imbalance must eventually decrease to a value of one or zero.

Lemma 3
Let $n(k) \triangleq |\{i \in \mathscr{N} : x_i(k) = m(k)\}|$ be the number of areas with only $m(k)$ vehicles at time k. For the system model described in Section 2.3, for any time index k and any $x(k) \in \mathscr{X}_0 - \mathscr{X}_I$, there exists a finite number $T \leqslant 2B\lceil N/2 \rceil$ such that either $m(k+T) > m(k)$ or $n(k+T) < n(k)$. In other words, whenever the state of the system is not in the invariant set $\mathscr{X}_I$, it is eventually the case that the minimum number of vehicles in any area increases or the subset of areas that have only that minimum number decreases in size.

Theorem 1
For the system model in Section 2.3, for any initial condition $x(0) \in \mathscr{X}_0$ there exists a finite number $T \leqslant 2B\lceil N/2 \rceil (N-1)(\lfloor L/N \rfloor + 1)$ such that $x(k) \in \mathscr{X}_I$ for all $k \geqslant T$.

A couple of remarks should be made about the above bound on the convergence time of our system model. The first thing to note is that this bound is generally conservative with respect to its dependence on the size of the ring N (a fact illustrated by the simulation results in Section 4 below). This conservativeness stems from the 'worst-case scenario' nature of our analysis which, while making such analysis tractable, ignores the fact that much of the system's progress towards a globally balanced state will usually be done in parallel instead of serially. It is also interesting to note that the bound in Theorem 1 is cast in terms of the *average* number of vehicles per area L/N as opposed to the total number L. The reason for this is that the coordinators are implicitly trying to lift the minimum number of vehicles in an area as close as they can to the average L/N (because accomplishing that feat means the system is very close to the globally balanced state). So, in effect, if we increase the length of the ring while leaving the total number of vehicles constant, it may take longer to raise the minimum number of vehicles

DOI: 10.1002/rnc

per area by one, but that minimum has to be raised fewer times overall in order for the system to reach the globally balanced state.

3.2. Extension to line topologies

A simple modification of the system model presented in Section 2.3 will allow us to achieve global balancing on line topologies. A line topology is an area–coordinator interconnection of N areas and $N-1$ coordinators in which coordinator i connects areas i and $i+1$ but no coordinator connects areas 1 and N (i.e. like a ring with one coordinator removed). See Figure 4(a) for the graph associated with a five-area line. In order to achieve global balancing on a line we will make a transformation of the interconnection as follows. We first rename the areas of the line interconnection as *zones* and then we assign two areas to each zone where both of these areas cover the entire physical area of their zone (i.e. they overlap completely). We then connect these areas into a unidirectional ring as shown in Figure 4(b). More specifically, a coordinator in the system is allowed to move vehicles from area i to area $i+1$ if and only if both those areas are in one of the two zones to which the coordinator is connected in the original topology.

Note that this set-up by itself almost allows us to globally balance the number of vehicles in the zones because globally balancing the ring of areas will mean that the number of vehicles in any two zones cannot differ by more than two. In order to achieve true global balancing of the zones, however, we must make a small modification to the system modelled in Section 2.3. Let us assume that areas 1 and N lie in one or the other of the zones with only one coordinator (i.e. one of the zones on either end of the line). Now, vehicle transfers from any area $i \neq N$ to area $i+1$ will obey the rules laid out in the enable function g of Section 2.3. Vehicle transfers from area N to area 1, however, will be handled a little differently. For this case, conditions (i) and (ii) of the enable function will be replaced by the following rules:

$$\text{if } \tilde{x}_N^N(k) \leqslant \tilde{x}_1^N(k) + 1 \quad \text{then } \alpha(k) = 0 \tag{7}$$

$$\text{if } \tilde{x}_N^N(k) \geqslant \tilde{x}_1^N(k) + 2 \quad \text{then } 1 \leqslant \alpha(k) \leqslant \left\lfloor \frac{\tilde{x}_i^i(k) - \tilde{x}_{i+1}^i(k)}{2} \right\rfloor \tag{8}$$

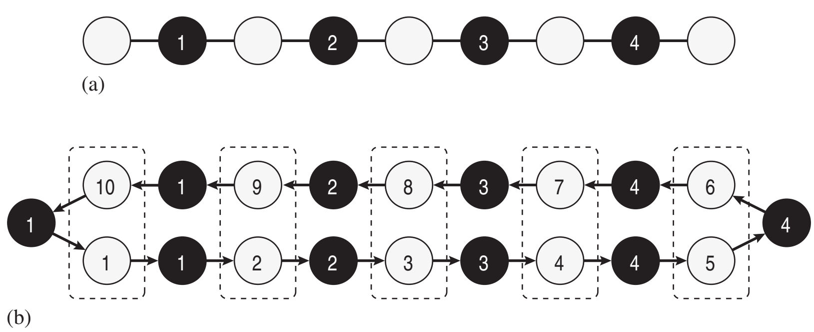

Figure 4. Original line topology shown in (a) and transformation into a ring shown in (b). Dashed rectangles in (b) denote zones (i.e. the original areas in (a)). Co-ordinator labels are repeated in (b) to clearly show which coordinator is responsible for each part of the ring.

DOI: 10.1002/rnc

where $\lfloor \cdot \rfloor$ is the floor operator, i.e. the largest integer less than or equal to its argument. The effect of this modified enable function is that the number of vehicles allowed to be transferred from area N to area 1 is limited so that area N has at least as many vehicles as area 1 after the transfer (and vehicles must be transferred when doing so will not violate this rule). Under this modification we have the following results.

Theorem 2

For the modification of the system model of Section 2.3 described by (7) and (8), for any initial condition $x(0) \in \mathscr{X}_0$ there exists a finite number $T \leqslant 2B(N-1)^2(\lfloor L/N \rfloor + 1)$ such that $x(k) \in \mathscr{X}_I$ for all $k \geqslant T$.

Theorem 3

For any $k \in \mathbb{N}$ such that $x(k) \in \mathscr{X}_I$, the line topology formed from the modified ring of Section 3.2 with N areas is guaranteed to have $\frac{1}{2}N$ globally balanced zones for all $k' \geqslant k + T$, where $T = \frac{1}{2}N(N-1)$.

Theorem 2 is almost identical to Theorem 1 in both its analysis and final result. The only difference is the modified rules (7) and (8) effectively slow down the (worst case) dynamics of the system by a factor of approximately two. The result of Theorem 3 is possible only because of the way we have set up the modified ring and where we located the 'special' coordinator that uses the modified rules. This modification forces the distribution of the vehicles in the globally balanced ring into a specific configuration where the zones are also globally balanced (see proofs in the Appendix).

3.3. Extension to prioritized balancing

While uniformly distributing the patrol vehicles across the environment may be the smart thing to do if we have no prior information about where the targets might appear, in many practical cases we will want to concentrate more vehicles in parts of the environment that have a higher priority than other parts. In this section we will discuss a few methods that help us to achieve such a goal.

There are three simple solutions to this problem that simply manipulate the way the system is set up in order to concentrate more vehicles in certain parts of the environment. First, if we know ahead of time what we want the distribution of vehicles to look like, we can partition the environment into areas in such a manner that the parts of the environment with higher priority are covered by areas of smaller size. Then, by using the same algorithm as in Section 2.3, when the system achieves a globally balanced distribution of vehicles across the areas, the areas of smaller size will have a higher density of vehicles (see Figure 5). The main problem with this method is that it is not very flexible; if the desired distribution changes, then the areas would have to change size and shape and many coordinators would likely have to adjust their position, creating a host of problems for vehicles that were trying to check in with them.

The second simple solution is to start with physical areas that are all the same size and then attach a number of 'virtual areas' to each physical area in proportion to its priority. Vehicles assigned to a virtual area would patrol the part of the environment covered by the physical area to which that virtual area is attached (similar to the way the two areas covered a zone in the line topology above). For example, if the priority of area i was twice that of area j, then two virtual

DOI: 10.1002/rnc

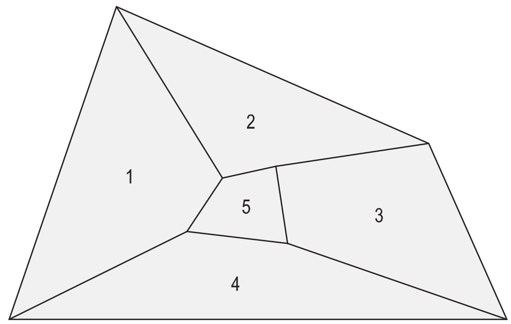

Figure 5. Example of surveillance area layout with areas of unequal size. The density of AAVs in area 5 will be higher than the other areas if the number of AAVs per area is the same.

areas could be added to area i and one to area j so that i would have twice as many vehicles as j once all the virtual areas were balanced. The advantage of this solution is that virtual areas can be added or deleted in response to changes in the priorities of the real areas. The disadvantages to this solution are that (1) the priorities of the real areas must be integer valued, and (2) since (in the balanced state) the number of vehicles in each virtual area may oscillate between two consecutive integers, the number of vehicles in a physical area with many virtual areas can vary considerably. Specifically, the number of vehicles in a physical area with v virtual areas can vary from $v\lfloor L/N \rfloor$ to $v\lfloor L/N \rfloor + v$ (where $\lfloor L/N \rfloor$ is the smallest number of vehicles in any virtual area when the system is balanced).

The third simple solution would simply be to give each coordinator the option to 'steal' vehicles from the system (i.e. keep vehicles in its area but report them as removed) whenever it discovers that there are high priority tasks to be done in its area and then return those vehicles after those tasks were accomplished. Although we have no analytical results for to prove this, so long as frequency and magnitude of these vehicle 'thefts' is limited, it is reasonable to assume that the redistribution dynamics of the system will keep the areas fairly well balanced.

We now present a fourth (and not so simple) option, which is to return to the formulation in which the areas are synonymous with the regions of the environment and generalize the algorithm of Section 2.3 in order to accommodate the possibility that some areas have a higher priority than others. Simply put, we would like each area to have a certain portion of the total number of vehicles (referred to as balancing 'virtual load' in [6, 7]). In Section 2.3 it was assumed that this proportion was simply $1/N$ for each of the N areas. For the more general case, each area $i \in \mathcal{N}$ is given a priority value $p_i \in (0, \infty)$ and its ideal proportion of the total load is given by p_i/P, where $P = \sum_{j=1}^{N} p_j$. In keeping with the notation established in Section 2.3, the same area's ideal number of vehicles is given by $(p_i/P)L$ and represents that area's 'fair share' of the L vehicles. Of course, this value will not typically be an integer for all areas, so any feasible division of the vehicles will result in some areas having more than their fair share and some having less. Determining the best distribution in this situation is what is known as either an *allocation problem* or an *integer sharing problem* and it has received a lot of attention in the area of political science because of its relevance to proportional representation in legislative bodies (e.g. if $L = 435$ and $p_1, \ldots, p_{50}$ are the populations of the U.S. states, then solving this allocation problem determines the number of seats each state receives in the House of Representatives).

DOI: 10.1002/rnc

Of course, the optimality of the allocation can be defined in many different ways, each requiring a different solution method and possibly producing a different result. A comprehensive review of this problem can be found in both [12, 13].

The optimality criterion in which we are primarily interested is reducing the maximum imbalance between the quantities x_i/p_i and x_j/p_j for any two areas i and j. This is a direct extension of our desire in Section 2.3 to reach a goal set where the number of vehicles in two areas differ by no more than one. Strictly speaking, we want to minimize the following cost function:

$$\max_{i,j\in\mathcal{N}} \left|\frac{x_i}{p_i} - \frac{x_j}{p_i}\right| = \max_{i\in\mathcal{N}} \frac{x_i}{p_i} - \min_{j\in\mathcal{N}} \frac{x_j}{p_j} \tag{9}$$

where it should be clear that this is equivalent to our previous goal of $x \in \mathcal{X}_I$ when $p_i = p_j$ for all $i,j \in \mathcal{N}$. Since multiplication of each patrol area's priority by a common number does not change their desired proportions but does reduce the value of this cost function, the following relative cost function is often used instead

$$\frac{\max_{i\in\mathcal{N}} x_i/p_i}{\min_{j\in\mathcal{N}} x_j/p_j} = \max_{i,j\in\mathcal{N}} \frac{p_j x_i}{p_i x_j} \tag{10}$$

and note that the same allocation minimizes both (9) and (10).

This optimality criterion is not covered in the previous references, but is addressed in [14]. That work proposes using dynamic programming methods to determine the optimal allocation. Given the complexity of finding the optimal allocation we will not attempt to solve that problem in a distributed manner, but rather develop an algorithm that forces the distribution of vehicles, $x(k)$, into a region of the state space where the value of the cost function (10) is within a some known bound of its optimal value.

The model we will use for our proportional balancing system will be identical to that in Section 2.3 with the exception of the enable function $g(x)$ and the goal set. To reflect the change from balancing vehicle numbers to balancing weighted versions thereof, conditions (i) and (ii) for part (a) of $g(x)$ are redefined as

$$\text{if } \frac{\tilde{x}_i^i(k)}{p_i} \leqslant \frac{\tilde{x}_{i+1}^i(k)}{p_{i+1}} \quad \text{then } \alpha(i) = 0 \tag{11}$$

$$\text{if } \frac{\tilde{x}_i^i(k)}{p_i} > \frac{\tilde{x}_{i+1}^i(k)}{p_{i+1}} \quad \text{then } 1 \leqslant \alpha(i) < \frac{p_i(1 - \tilde{x}_{i+1}^i(k)) + p_{i+1}(1 + \tilde{x}_i^i(k))}{p_i + p_{i+1}} \tag{12}$$

Rule (11) and the lower limit on $\alpha(i)$ in (12) ensure that coordinator i transfers at least one vehicle if area i has a higher number of vehicles in proportion to its priority than does area $i+1$ (and none if it has less). The upper limit on $\alpha(i)$ in (12) has been recalculated to ensure that coordinator i does not pass more than the minimum number of vehicles necessary to reverse the sign of the imbalance between x_i/p_i and x_{i+1}/p_{i+1}. It is easily verified that if $p_i = p_{i+1}$ then (11) and (12) are equivalent to conditions (i) and (ii) of the original enable function $g(x)$ in Section 2.3.

Our goal set for prioritized balancing will be the following:

$$\mathcal{X}_{\mathrm{P}} = \left\{ x \in \mathcal{X}_0 : \frac{x_{i+1}}{p_{i+1}} - \frac{x_i}{p_i} < \frac{1}{p_i} + \frac{1}{p_{i+1}} \ \forall i \in \mathcal{N} \right\}$$

DOI: 10.1002/rnc

Unfortunately, this set is defined as a collection of local conditions rather than a single global one as $\mathcal{X}_I$ is. Indeed, even when $p_i = p_{i+1}$ for all $i,j \in \mathcal{N}$, $\mathcal{X}_I$ is a proper subset of $\mathcal{X}_\mathrm{P}$ for all but a few trivial situations (i.e. when L and/or N are less than two). The fact that we can achieve better results for the non-proportional case justified its separate analysis in Section 3.1. The following analysis of the proportional balancing case will show that the set $\mathcal{X}_\mathrm{P}$ is invariant and the system model converges to this set in a finite time of known bound.

Lemma 4
The set $\mathcal{X}_\mathrm{P}$ is invariant with respect to the system model of Section 2.3 and the modifications to the enable function defined by (11) and (12).

As with $\mathcal{X}_I$ in Section 3.1, trajectories within $\mathcal{X}_\mathrm{P}$ are hardly ever static. Unfortunately, because of the strictly local results that we are able to achieve with prioritized balancing, we are not guaranteed that the system will ever achieve the minimum global imbalance. In fact, due to the 'churning' nature of trajectories in $\mathcal{X}_\mathrm{P}$, the system will not even stay at the state with the minimum global imbalance unless p_i, $i = 1, \ldots, N$ and L are such that it is possible to have zero global imbalance (otherwise there must be at least one area i such that $x_{i+1}(k)/p_{i+1} - x_i(k)/p_i$ is greater than zero, which will eventually result in coordinator i transferring a vehicle from area i to area $i+1$).

Theorem 4
For the system model of Section 2.3 with the modifications to the enable function defined by (11) and (12), for any initial condition $x(0) \in \mathcal{X}_0$ there exists a finite number $T \in \mathbb{N}$ such that $x(k) \in \mathcal{X}_\mathrm{P}$ for all $k \geqslant T$, where

$$T < \frac{2B(G^N - G)}{(G-1)^2} \left\lceil \frac{L}{\min_i(p_i p_{i+1})} \right\rceil$$

and $G = \max_i \lceil 1 + p_i/p_{i+1} \rceil$.

Because of the many worst-case assumptions made in its derivation, the bound in Theorem 4 is typically extremely conservative. Even though the bound on T is not of much use, its derivation does provide insight into the dynamics of the system (see Appendix).

At this point it is instructive to explain why prioritized balancing fails to achieve the minimum global imbalance as the non-proportional algorithm does. The main reason for this shortcoming is the ambiguity that the prioritized algorithm introduces at the local level of the coordinators. In both the prioritized and non-prioritized algorithms, a coordinator must make decisions based on only the number of vehicles in two areas. In the non-prioritized case a coordinator i makes an estimate of the average number of vehicles per area for the whole system by taking the average of just areas i and $i+1$, and then, if it can, transfers vehicles from area i to area $i+1$ to achieve this average (giving area $i+1$ the extra vehicle is there is one). For example, take a system that has reached a globally balanced state where each area has either 3 or 4 vehicles. Assuming perfect estimates for simplicity, if we have either $x_i = 3, x_{i+1} = 3$, or $x_i = 4, x_{i+1} = 3$, or $x_i = 3, x_{i+1} = 4$ then coordinator i will correctly estimate that proper minimum number of vehicles per area (i.e. $\tilde{m} = 3$) and can act accordingly to move the excess vehicles forward on the ring if it is in a position to do so. In the case where $x_i = 4, x_{i+1} = 4$ coordinator i will improperly estimate $\tilde{m}$ as being 4, but since it will not take any action in this case, the ambiguity here does not hurt.

DOI: 10.1002/rnc

Now, consider a similar situation in the prioritized balancing case. The closest analogy to the globally balanced state of the non-prioritized case is if each area i has either $\tilde{m}_i = \lfloor (p_i/P)L \rfloor$ or $\tilde{m}_i + 1$ vehicles (i.e. the integer part of its fair share of the total vehicles or that number plus one). For our example, let $p_i = 2$ and $p_{i+1} = 3$ and let L be such that $\tilde{m}_i = 2$ and $\tilde{m}_{i+1} = 3$. If coordinator i arrives at a situation in which $x_i = 3$ and $x_{i+1} = 4$, then it does not have enough information to act correctly in order to keep the overall system balanced. Since the only way coordinator i can estimate $\tilde{m}_i$ and $\tilde{m}_{i+1}$ is to make a weighted average of those two areas, it will wind up estimating these minimum vehicle numbers as $\lfloor \frac{2}{5} \cdot 7 \rfloor = 2$ and $\lfloor \frac{3}{5} \cdot 7 \rfloor = 4$, respectively, instead of their correct values of 2 and 3. Thus, the coordinator will not know it is acting incorrectly when it transfers a vehicle from area i to area $i+1$.

4. SIMULATIONS

4.1. Redistribution dynamics

The purpose of this section is to provide the illustrative example of the system dynamics. To do this we simulated a system with 10 areas and 127 vehicles which is initially globally balanced but loses all the vehicles in areas 7–10 shortly into the simulation. That is to say that the system experience a step disturbance which, if we redefine our initial time, is equivalent to starting the system in an unbalanced state. We also note that the simulation probably has an unrealistically high number of vehicles for our patrol scenario, but we do this because it helps highlight the system dynamics. Check in times of the vehicles were staggered at random intervals and coordinators made a transfer decision every time a vehicle checked in with it. In our simulation we used real-time delays (as opposed to the event time used in the model and analysis) so all state values will appear as functions of t as opposed to k. Since we gain no additional insight by scaling the length of the delays, in all our simulations one unit of time is taken to be equal to the maximum delay between successive check ins by an individual vehicle. Figure 6 shows the resulting trajectory of $x_i(t)$ for each area $i = 1, \ldots, 10$ and shows how the system recovers a globally balanced state within 30 time units of the disturbance (albeit at a lower number of vehicles per area). In this plot we can see how the system responds fairly gracefully to the disturbance, noting how the minimum and maximum number of vehicles in an area are non-decreasing and non-increasing, respectively. The same data are also shown in Figure 7 with the value of $x_i(t)$ plotted in shades of grey in order to show how the holes created by the disturbance at time $t = 5$ get filled in from the other areas of the ring. This figure also shows the dynamic nature of the globally balanced state (i.e. from time $t = 35$ onward we can see how the 'excess' vehicles shuffle along the ring in a staggered fashion).

4.2. Average convergence time

Next, since the bound on convergence time expressed in Theorem 1 is generally conservative, we are motivated to perform some statistical analysis of the system's practical performance via Monte Carlo simulations to determine the average convergence time for a typical scenario. In order to do this we started each simulation in an initial condition that created the maximum initial imbalance (i.e. putting all the vehicles into a single area) while initializing all the coordinator estimates to zero since this should produce a good estimate of the system's actual worst-case performance and takes into account the initial time period before our assumption of

DOI: 10.1002/rnc

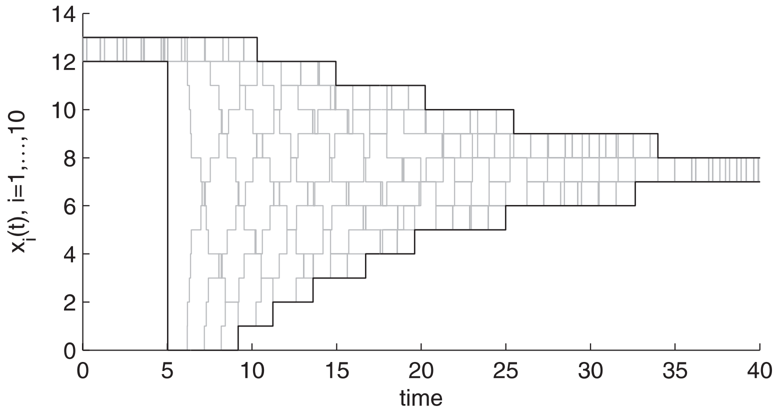

Figure 6. Number of vehicles in each area of a ring when a number of vehicles are removed at time $t = 5$. Trajectories $x_i(t)$, $i = 1, \ldots, 10$ are drawn in grey with $\max_{i \in \mathcal{N}} x_i(t)$ and $\min_{i \in \mathcal{N}} x_i(t)$ highlighted with bold black lines. One time unit is equal to the maximum delay between successive check ins by an individual vehicle.

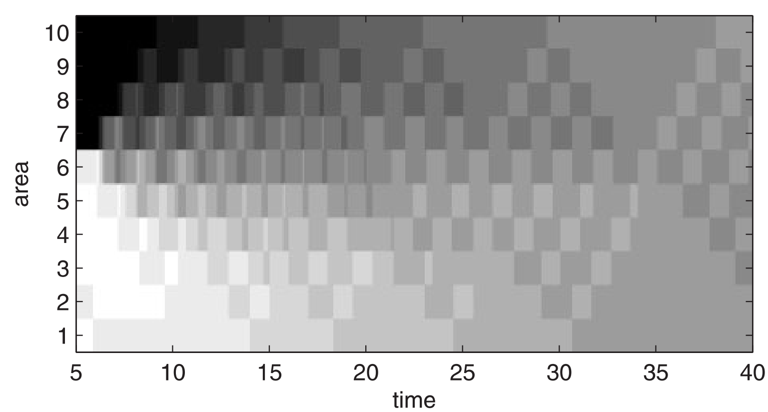

Figure 7. Same data as Figure 6 plotted spatially with the value of $x_i(t)$ denoted by the grey value of the block at (t, i) (black equals zero and white equals 13).

$x \in \mathscr{X}_E$ is guaranteed to be valid. The travel time between coordinators was fixed and constant for all areas with the check in times of the vehicles randomly staggered. Co-ordinators again made a decision to transfer a vehicle every time one checked in. The convergence time of the system was investigated for different combinations of the number of areas N and the average number of vehicles per area L/N. The length of the travel time between coordinators was not varied as the results for this simulation set-up would simply be scaled in proportion to this time. The data points that appear in Figures 8 and 9 are the average of 50 simulation runs each. While this may seem like a small number of runs to use, it is justified by the extremely small sample variance.

DOI: 10.1002/rnc

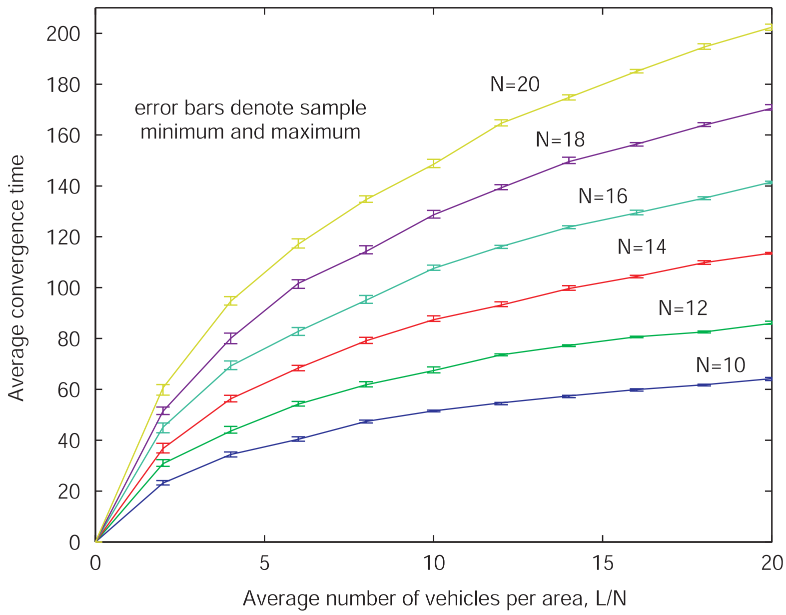

Figure 8. Average convergence time *versus* average number of vehicles per area for different numbers of areas. One time unit is equal to the maximum transit time of a vehicle between the coordinators of its assigned area.

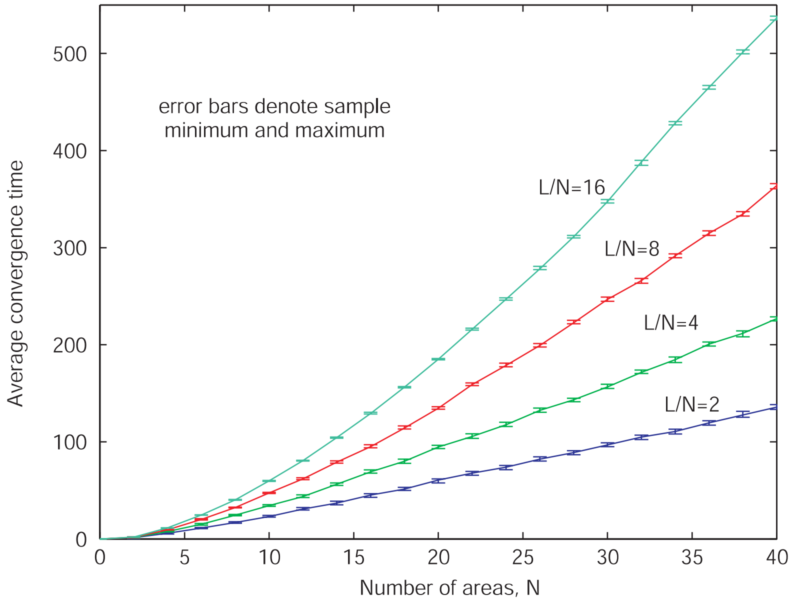

Figure 9. Average convergence time *versus* number of areas for different values of the average number of vehicles per area. One time unit is equal to the maximum transit time of a vehicle between the coordinators of its assigned area.

DOI: 10.1002/rnc

When the number of areas in the ring is kept constant and the average number of vehicles per area was increased, we found a roughly logarithmic relationship between the average number of vehicles and the convergence time of the system (Figure 8) rather than the linear one implied by the bound of Theorem 1. This discrepancy is due to the nature of our analysis in the proof of that theorem. In that analysis we always assume that the number of vehicles transferred is the minimum of one per event and that these events happen at the slowest possible rate. What happens in our simulation set-up, however, is that over a period of two time units (i.e. the time between two successive visits of a vehicle to the same coordinator) a coordinator transfers roughly the maximum number of vehicles that it can. Since this upper limit is proportional to the relative imbalance of the two areas a coordinator connects, the global imbalance is reduced at a rate roughly proportional to its size. This phenomenon is observed somewhat in Figure 6 where the rate at which the global imbalance decreases slows down as the system nears the balanced state. Since the settling time of the dynamical system $\dot{z} = -az$ increases in proportion to the logarithm of its initial condition z_0, it is not too surprising for us to find a similar relationship in our simulations results.

When the average number of vehicles per area is held constant our simulation results (Figure 9) suggest that the convergence time has an (asymptotically) linear relationship to the number of areas in the ring (as opposed to the quadratic bound in Theorem 1). To see why we kept the average number of vehicles per area L/N constant instead of just the total number L let us instead consider what we could expect in the latter case as the number of areas increased. Going between one and L areas the convergence time would increase because it would naturally take longer for the vehicles to spread out from their original area. For any number of areas larger than L, however, once the vehicles get spread out across the first L areas, the system is already as balanced as it can it, and so the convergence time could not increase any further.

5. CONCLUSIONS

In this paper we have presented a novel formulation of the resource/load-balancing problem in which a distributed set of coordinators must share responsibility for multiple groups of resources (as opposed to the traditional formulation in which each coordinator has sole responsibility for a single resource group). In addition, we have developed a distributed cooperative control strategy that enables those coordinators to achieve a distribution of vehicles that is globally balanced across the areas (instead of just locally balanced between connected areas). The globally balanced state is achieved 'gracefully' in that the global imbalance is non-increasing and it is also achieved within a finite time of known bound. To our knowledge, this is the first cooperative control strategy capable of achieving global balancing in a distributed and asynchronous setting. Although presented in the context of an autonomous vehicle patrol mission, the algorithm developed here has significant application to generic resource or load-balancing problems due to its ability to achieve this globally balanced state.

Our current research efforts lie in developing a cooperative control strategy for a larger class of area–coordinator interconnections (i.e. not just a unidirectional ring). Achieving such a generalization has the potential to decrease the convergence time of the algorithm by allowing much more highly interconnected area set-ups. Potential future research directions include the analysis of situations in which vehicles may enter or leave the system at a constrained rate or in which the areas' priorities vary with time.

APPENDIX A

A.1. Proof of Lemma 1

Throughout this proof and the rest of this section we use the function $m(k) \triangleq \min\{x_i(k) : i \in \mathcal{N}\}$ to denote the smallest number of vehicles possessed by any area in the ring at time index k. It should be clear from the definition of $\mathcal{X}_I$ that $x(k) \in \mathcal{X}_I$ if and only if $x_i(k) \in \{m(k), m(k)+1\}$. Recall that $x(k) \in \mathcal{X}_I$ implies $x(k) \in \mathcal{X}_0$ as well and so $\tilde{x}_i^i(k) \leqslant x_i(k)$ and $\tilde{x}_{i+1}^i(k) \geqslant x_{i+1}(k)$ for all $i \in \mathcal{N}$. For any enabled event $e(k) \in g(x(k))$, its effect on the number of vehicles in area i will be determined solely by $x_i(k)$ and the values $\alpha(i)$ and $\alpha(i-1)$ as described by the update function $f(x(k), e(k))$. For clarity, we break down the argument into two cases as follows:

Case 1: $x_i(k) = m(k)$. We have that $\alpha(i) = 0$ due to condition (*i*) of the enable function because

$$\tilde{x}_i^i(k) \leqslant x_i(k) = m(k) \leqslant x_{i+1}(k) \leqslant \tilde{x}_{i+1}^i(k)$$

and because $x_{i-1}(k) \leqslant m(k) + 1$ it follows that

$$\tilde{x}_{i-1}^{i-1}(k) \leqslant x_{i-1}(k) \leqslant m(k) + 1 = x_i(k) + 1 \leqslant \tilde{x}_i^{i-1}(k) + 1$$

which implies $\tilde{x}_{i-1}^{i-1}(k) - \tilde{x}_i^{i-1}(k) \leqslant 1$ and so $\alpha(i-1) \in \{0, 1\}$. Thus,

$$x_i(k+1) = x_i(k) - \alpha(i) + \alpha(i-1) = x_i(k) + \alpha(i-1)$$

and hence we have $x_i(k+1) \in \{m(k), m(k)+1\}$.

Case 2: $x_i(k) = m(k) + 1$. This time $\alpha(i-1) = 0$ because

$$\tilde{x}_{i-1}^{i-1}(k) \leqslant x_{i-1}(k) \leqslant m(k) + 1 = x_i(k) \leqslant \tilde{x}_i^{i-1}(k)$$

and $\alpha(i) \in \{0, 1\}$ because

$$\tilde{x}_i^i(k) \leqslant x_i(k) = m(k) + 1 \leqslant x_{i+1}(k) + 1 \leqslant \tilde{x}_{i+1}^i(k) + 1$$

Hence, we have

$$x_i(k+1) = x_i(k) - \alpha(i) + \alpha(i-1) = x_i(k) - \alpha(i)$$

and so $x_i(k+1) \in \{m(k), m(k)+1\}$ again.

Thus, these two cases show that if $x(0) \in \mathcal{X}_I$, it must be the case that $x(k) \in \mathcal{X}_I$ for all k. □

A.2. Proof of Lemma 2

To prove the first part of the hypothesis we show that $x_i(k+1) \geqslant m(k)$ for all $i \in \mathcal{N}$ and for all k. For any time index k consider the possible change in each $x_i(k)$. Since the number of vehicles in an area i can only decrease when vehicles are transferred to area $i+1$, if coordinator i does not have a corresponding partial event $e_i^{\alpha(i)}$ in $e(k)$, then we automatically have that $x_i \times (k+1) \geqslant x_i(k) \geqslant m(k)$. Consider now the value of $x_i(k+1)$ when coordinator i does have

DOI: 10.1002/rnc

a corresponding partial event $e_i^{\alpha(i)} \in e(k)$. Assume $\tilde{x}_i^i(k) > \tilde{x}_{i+1}^i(k)$ (or else $\alpha(i) = 0$ and the coordinator will still not transfer any vehicles)

$$
\begin{aligned}
x_i(k+1) &= x_i(k) - \alpha(i) + \alpha(i-1) \geqslant x_i(k) - \alpha(i) \\
&\geqslant x_i(k) - \left\lceil \frac{\tilde{x}_i^i(k) - \tilde{x}_{i+1}^i(k)}{2} \right\rceil \\
&> x_i(k) - \left(\frac{1}{2} \tilde{x}_i^i(k) - \frac{1}{2} \tilde{x}_{i+1}^i(k) + 1 \right) \\
&\geqslant x_i(k) - \left(\frac{1}{2} x_i(k) - \frac{1}{2} x_{i+1}(k) + 1 \right) \\
&= \frac{1}{2} x_i(k) + \frac{1}{2} x_{i+1}(k) - 1 \\
&\geqslant \frac{1}{2} m(k) + \frac{1}{2} m(k) - 1 = m(k) - 1
\end{aligned}
$$

where the second inequality comes from condition (ii) of the enable function. The inequality used in the third line is $\lceil b \rceil < b + 1$. Due to the integer nature of the states, the result that $x_i(k+1) > m(k) - 1$ implies $x_i(k+1) \geqslant m(k)$.

To prove the second part of the hypothesis, we similarly have that $x_i(k+1) \leqslant x_i(k) \leqslant M(k)$ unless coordinator $i-1$ transfers vehicles into area i. Take a coordinator $i-1$ which does have a corresponding partial event $e_{i-1}^{\alpha(i-1)} \in e(k)$ and consider the value of $x_i(k+1)$ assuming $\tilde{x}_{i-1}^{i-1} \times (k) > \tilde{x}_i^{i-1}(k)$

$$
\begin{aligned}
x_i(k+1) &= x_i(k) - \alpha(i) + \alpha(i-1) \leqslant x_i(k) + \alpha(i-1) \\
&\leqslant x_i(k) + \left\lceil \frac{\tilde{x}_{i-1}^{i-1}(k) - \tilde{x}_i^{i-1}(k)}{2} \right\rceil \\
&< x_i(k) + \left(\frac{1}{2} \tilde{x}_{i-1}^{i-1}(k) - \frac{1}{2} \tilde{x}_i^{i-1}(k) + 1 \right) \\
&\leqslant x_i(k) + \left(\frac{1}{2} x_{i-1}(k) - \frac{1}{2} x_i(k) + 1 \right) \\
&= \frac{1}{2} x_i(k) + \frac{1}{2} x_{i-1}(k) + 1 \\
&\leqslant \frac{1}{2} M(k) + \frac{1}{2} M(k) = M(k) + 1
\end{aligned}
$$

and $x_i(k+1) < M(k) + 1$ implies $x_i(k+1) \leqslant M(k)$. Thus, the hypothesis has been proven. □

A.3. Proof of Lemma 3

Throughout the following argument, assume that $m(k)$ does not increase. If it does, then the hypothesis is proven trivially. Given the result of Lemma 2, this is equivalent to assuming that $m(k)$ remains constant.

In order to better illustrate the dynamics of the system model, we adopt the following conventions for the remainder of this proof. Since the area–coordinator interconnection

DOI: 10.1002/rnc

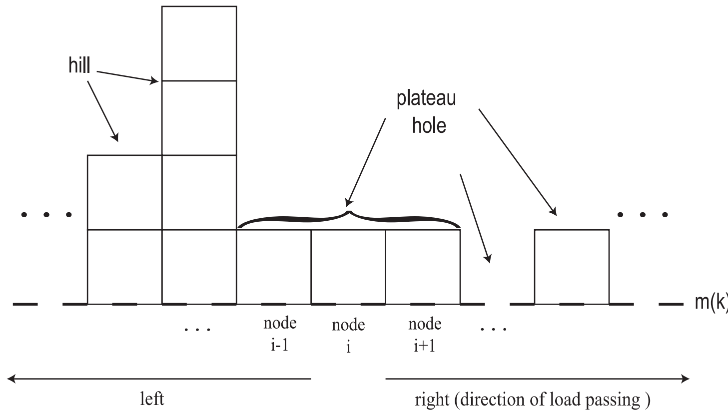

Figure A1. Illustration of terms.

topology is a unidirectional ring, we will call the direction that vehicles are transferred along the ring *right*, and the opposite direction *left*. Nodes with $x_i(k)=m(k)$ will be referred to as *holes* and those with $x_i(k)\geqslant m(k)+2$ as *hills*. A group of nodes $\{i,i+1,\ldots,j\}\subset\mathcal{N}$ (where i may equal j) such that $x_n(k)=m(k)+1$ for all $i\leqslant n\leqslant j$ and such that neither $x_{i-1}(k)$ nor $x_{j+1}(k)$ equal $m(k)+1$ will be called a *plateau* with *length* given by $(j-i+1) \bmod N$. The motivation for these terms comes from visualizing the vehicles as unit sized blocks which are stacked according their area (see Figure A1).

We first show that $n(k)$, the number of areas with only $m(k)$ vehicles, is non-increasing. If $n(k)$ did increase, then at some time k there would be at least one coordinator i transferring vehicles such that $x_i(k)\geqslant m(k)+1$ and $x_i(k+1)=m(k)$ (i.e. it would have to transfer so many vehicles out of area i that that area went from being a hill or part of a plateau to being a hole). It would also have to be the case that both $x_{i+1}(k)$ and $x_{i+1}(k+1)$ were greater than $m(k)$ because (a) if $x_{i+1}(k)=m(k)$ then, as shown in the proof of Lemma 1, we would have $\alpha(i+1)=0$ and so any transfer of vehicles from area i to area $i+1$ could at most change which areas have only $m(k)$ vehicles, but not the number of such areas, and (b) Lemma 2 tells us that $x_{i+1}(k)-\alpha(i+1)\geqslant m(k)$ so a positive value for $\alpha(i)$ would ensure that $x_{i+1}(k+1)\geqslant m(k)+1$. We can then borrow the intermediate inequality $x_i(k+1)>\frac{1}{2}x_i(k)+\frac{1}{2}x_{i+1}(k)-1$ from the proof of Lemma 2 to see that $x_i(k+1)>m(k)$, which is a contradiction.

To begin the proof that $n(k)$ is eventually decreasing take any time index k and any state $x(k)\notin\mathcal{X}_I$. Now, since $x(k)\notin\mathcal{X}_I$, there exists areas $\bar{i}$ and $\underline{i}$ such that $\bar{i}$ is a hill, $\underline{i}$ is a hole, and $\{\bar{i}+1,\bar{i}+2,\ldots,\underline{i}-1\}$ is a plateau with length not greater than $N-2$. Assume for now that the plateau is at least two areas long. Note that until coordinator $\bar{i}$ or $\underline{i}-1$ transfers some vehicles (to areas $\bar{i}+1$ or $\underline{i}$, respectively), none of the other coordinators for the areas in $\{\bar{i},\bar{i}+1,\ldots,\underline{i}\}$ will transfer any vehicles. This has already been shown for coordinator $\underline{i}$ since $x_{\underline{i}}(k)=m(k)$, but it also holds for the coordinators for the areas in the plateau (except coordinator $\underline{i}-1$) because $\tilde{x}_i^i(k)\leqslant x_i(k)=x_{i+1}(k)\leqslant\tilde{x}_{i+1}^i(k)$ for $i\in\{\bar{i}+1,\ldots,\underline{i}-2\}$. Also note that until coordinator $\bar{i}$ transfers some vehicles out of area $\bar{i}$ that area is guaranteed to remain a hill.

According to the second condition imposed by the set of allowed trajectories, E_B, if coordinator i does not transfer any vehicles from time index k to time index $k+B$, then its

DOI: 10.1002/rnc

estimate $\tilde{x}^i_i(k+B)$ is at least the value of $x_i(k)$ and its estimate $\tilde{x}^i_{i+1}(k+B)$ is no more than $x_{i+1}(k-B)$. This means unless node $\bar{i}$ transfers at least one vehicle to area $\bar{i}+1$ by time $k+B$, then

$$\tilde{x}^{\bar{i}}_{\bar{i}}(k+B) \geqslant x_{\bar{i}}(k) \geqslant m(k)+2 > m(k)+1 = x_{\bar{i}+1}(k) \geqslant \tilde{x}^{\bar{i}}_{\bar{i}+1}(k+B)$$

and then in accordance with the first condition of E_B and the second condition of the enable function g there must occur a partial event $e^{\alpha(\bar{i})}_{\bar{i}}$ with positive $\alpha(\bar{i})$ within the next B time steps (i.e. for some k' such that $k+B \leqslant k' \leqslant k+2B-1$). Therefore, by time $k+2B$ we will have $x_{\bar{i}+1} \times (k') \geqslant m(k)+2$ and $x_{\bar{i}}(k') \geqslant m(k)+1$ which means that area $\bar{i}+1$ becomes a hill without area $\bar{i}$ becoming a hole. In effect, a hill is forced to move right along the ring at least once every $2B$ time steps.

By an argument parallel to the one above, unless node $\underline{i}-1$ transfers a vehicle to area $\underline{i}$ by time $k+B$, then

$$\tilde{x}^{\underline{i}-1}_{\underline{i}-1}(k+B) \geqslant x_{\underline{i}-1}(k) \geqslant m(k)+1 > m(k) = x_{\underline{i}}(k) \geqslant \tilde{x}^{\underline{i}-1}_{\underline{i}}(k+B)$$

and so a partial event $e^{\alpha(\underline{i}-1)}_{\underline{i}-1}$ with positive $\alpha(\underline{i}-1)$ must occur for some k'' such that $k+B \leqslant k'' \leqslant k+2B-1$. We will then have $x_{\underline{i}-1}(k'') = m(k)$ and $x_{\underline{i}}(k'') = m(k)+1$ which means that area $\underline{i}-1$ has become a hole and area $\underline{i}$ has been elevated to the level of a plateau. Here, we have shown that a hole is forced to migrate to the left along the ring at least once every $2B$ time steps.

Given the forced motion of the hill and hole of interest we can see that the length of the plateau must decrease by at least 2 by time index $k+2B$. We can now redefine $\bar{i}$ and $\underline{i}$ appropriately (i.e. $\bar{i} := \bar{i}+1$ and $\underline{i} := \underline{i}-1$) and repeat the same argument as many times as necessary in order to show that by no later than time index $k+2B\lceil (N-2)/2 \rceil$ we have either one or no areas in the plateau between $\bar{i}$ and $\underline{i}$. In the case where the plateau is of zero length, since coordinator $\underline{i}$ will not transfer any vehicles, by some time k' less than $k+2B \times \lceil (N-2)/2 \rceil + B - 1$ we will have that $\tilde{x}^{\bar{i}}_{\bar{i}}(k') \geqslant x_{\bar{i}}(k'-B) \geqslant m(k)+2$ and $\tilde{x}^{\bar{i}}_{\underline{i}}(k'-B) = x_{\underline{i}}(k') = m(k)$ and thus by some time k'' less than $k+2B\lceil (N-2)/2 \rceil + 2B - 1$ coordinator $\bar{i}$ will have a corresponding partial event $e^{\alpha(\bar{i})}_{\bar{i}} \in e(k'')$ with $\alpha(\bar{i})$ such that

$$1 \leqslant \alpha(\bar{i}) < \tfrac{1}{2}(\tilde{x}^{\bar{i}}_{\bar{i}}(k'') - \tilde{x}^{\bar{i}}_{\underline{i}}(k'')) + 1 \leqslant \tfrac{1}{2}(x_{\bar{i}}(k'') - m(k)) + 1$$

With this bound on $\alpha(\bar{i})$, it is easily shown that $x_{\bar{i}}(k''+1) \geqslant m(k)+1$ and $x_{\underline{i}}(k''+1) \geqslant m(k)+1$ using the same logic as in Lemma 2. Hence, the hole is 'filled in' with vehicles from the hill, but not so many that the hill becomes a hole itself. The same process occurs if the plateau has length one because both coordinator $\bar{i}$ and $\bar{i}+1 = \underline{i}-1$ will be forced to transfer at least one vehicle to areas $\bar{i}+1$ and $\underline{i}$, respectively, within the next $2B$ time steps. Either the hole first moves to area $\underline{i}-1$ and then gets filled in from the hill in area $\bar{i}$ (possibly simultaneously) or the hill first moves to area $\underline{i}-1$ and then fills in the hole in area $\underline{i}$ (with the same guarantee that no new hole is created when the old one is filled in). Every time that a hole gets filled in in this manner, the number of holes obviously decreases by one. Since the entire process described above is guaranteed to occur within $\lceil (N-2)/2 \rceil + 1 = \lceil N/2 \rceil$ iterations of length no greater than $2B$, the hypothesis is shown to be true for $T = 2B\lceil N/2 \rceil$. □

A.4. Proof of Theorem 1

Define $m(k)$ and $n(k)$ as in the previous proofs. Starting from any initial state $x(0) \in \mathcal{X}$ and given any event sequence $E \in E_B$ defining a trajectory $X = [x(0), x(1), \ldots]$, it follows from Lemma 3

DOI: 10.1002/rnc

that when $x(k) \notin \mathcal{X}_I$ and the minimum resource level $m(k)$ remains constant, a process occurs whereby a hole is eliminated within $2B\lceil N/2 \rceil$ time steps. Since there can be no more than $N-1$ holes, it follows that all the holes at the current minimum resource level must be eliminated within $2B\lceil N/2 \rceil (N-1)$ time steps. Since the total number of vehicles in the system is equal to some fixed finite value L, there exists a finite value $\tilde{m}$ which $m(k)$ cannot exceed. Specifically, $\tilde{m} = \lfloor L/N \rfloor$. Since $m(0) \geqslant 0$, it must be the case that $m(k)$ has increased to $\tilde{m}$ by time $2B\tilde{m}\lceil N/2 \rceil (N-1)$ and remains constant from that point on.

Again by Lemma 3, $n(k)$ must keep decreasing. This time, however, once $n(k)$ is reduced to $N(\tilde{m}+1) - L$, those areas with more than $\tilde{m}$ vehicles must have $\tilde{m}+1$ vehicles (or else $\sum_i^N x_i(k)$ would not equal L). If there are more holes than $N(\tilde{m}+1) - L$, it will take at most another $2B\lceil N/2 \rceil (N-1)$ time steps for the excess holes to be eliminated. Thus, $x(k)$ will enter the invariant set $\mathcal{X}_I$ no later than time $T = 2B\lceil N/2 \rceil (N-1)(\lfloor L/N \rfloor + 1)$. □

A.5. Proof of Theorem 2

Proof

With the modification described by (7) and (8), we are no longer guaranteed that both a hill will move right and a hole left every $2B$ time steps and thus decrease the length of the plateau between them. This is because under (7) a hill at area N will not move to area 1 if area 1 is a plateau, nor will a hole at area 1 move to area N if area N is a plateau. However, we are guaranteed that if the plateau consists of at least one area, then either the hill *or* the hole will move in every $2B$ time steps (because it is not possible to have both the hill at area N and the hole at area 1 in this case). Also, when the hill and hole are at consecutive areas then even if they are at areas N and 1 the hole will be filled in with $2B$ time steps. Thus, the holes are still guaranteed to eventually get filled in, only it could take not just $2B\lceil N/2 \rceil$ time steps to fill in a hole as in the unmodified system, but up to $2B(N-1)$ time steps (i.e. the plateau must decrease in length by just one area every $2B$ time steps, the maximum length of a plateau is $N-2$ areas, and it could take another $2B$ time steps to fill in the hole after the plateau has been eliminated). The maximum value for T is then taken from Theorem 1 by substituting $2B(N-1)$ for $2B\lceil N/2 \rceil$. □

A.6. Proof of Theorem 3

Take any k such that the modified ring associated with the line topology has $x(k) \in \mathcal{X}_I$. Let H be the number of areas which are holes in the globally balanced state. These H holes must eventually move left on the ring until areas 1 through H are holes and areas $H+1$ through N form a plateau, at which point all vehicle transfers stop because of rule (7). Because of the way the modified ring is set up (with areas 1 and N in one of the end zones of the line topology), when this state is reached either all the zones will have *at most* one area which is a hole (i.e. if $H \leqslant N/2$) or *at least* one area which is a hole (i.e. if $H \geqslant N/2$). In this situation the number of vehicles in any two zones can differ by no more than one, so they are globally balanced. To form a worst-case bound on the number of time steps it may take to reach this state, note when $x(k) \in \mathcal{X}_I$, the hole that will eventually be in area 1 must be in some area $i \leqslant N-H+1$ at time k and thus must be in area 1 no later than $k + 2B(N-H)$. Similarly the hole that eventually winds up in area 2 must get there no

DOI: 10.1002/rnc

later than $k+2B(N-H)+2B(N-H+1)$, etc. so the hole that is eventually in area H gets there no later than (and thus the zones are globally balanced by) $k+2B\sum_{i=0}^{H-1}(N-H+i)=k+2B(HN-\frac{1}{2}H(H+1))$. This quadratic bound is maximized at both $H=N-1$ and $H=N$ (although when $H=N$ the zones are actually balanced as soon as the areas area globally balanced). Thus, when the areas are globally balanced at time step k, the zones will be globally balanced no later than time step $k+\frac{1}{2}N(N-1)$. □

A.7. *Proof of Lemma 4*

To prove the set $\mathscr{X}_{\mathrm{P}}$ is invariant it suffices to show that if $x_{i+1}(k)/p_{i+1}-x_i(k)/p_i$ is less than $1/p_i+1/p_{i+1}$ (true for all $i\in\mathscr{N}$ if and only if $x(k)\in\mathscr{X}_{\mathrm{P}}$) then $x_{i+1}(k+1)/p_{i+1}-x_i(k+1)/p_i$ is as well. Take any $i\in\mathscr{N}$ and consider the value of $x_{i+1}(k+1)/p_{i+1}-x_i(k+1)/p_i$.

Case 1: $x_{i+1}(k)/p_{i+1}-x_i(k)/p_i\geqslant 0$. Here, it must be the case that $\tilde{x}^i_{i+1}(k)/p_{i+1}-\tilde{x}^i_i(k)/p_i\geqslant 0$ because of the conditions imposed on the estimates by our assumption that $x\in\mathscr{X}_0\subset\mathscr{X}_E$. This implies that $\alpha(i)=0$ by the modified enable function and consequently we have both $x_i(k+1)\geqslant x_i(k)$ and $x_{i+1}(k+1)\leqslant x_{i+1}(k)$. Therefore, $x_{i+1}(k+1)/p_{i+1}-x_i(k+1)/p_i$ will be no greater than $x_{i+1}(k)/p_{i+1}-x_i(k)/p_i$, so it must still satisfy the local condition of $\mathscr{X}_{\mathrm{P}}$.

Case 2: $x_{i+1}(k)/p_{i+1}-x_i(k)/p_i<0$. In this case coordinator i may transfer vehicles from area i to area $i+1$. In this case we make use of the upper limit on $\alpha(i)$ from (12) to show that the same bound holds,

$$
\begin{aligned}
\frac{x_{i+1}(k+1)}{p_{i+1}}-\frac{x_i(k+1)}{p_i} &= \frac{x_{i+1}(k)+\alpha(i)}{p_{i+1}}-\frac{x_i(k)-\alpha(i)}{p_i}\\
&= \frac{p_i x_{i+1}(k)-p_{i+1}x_i(k)+(p_i+p_{i+1})\alpha(i)}{p_i p_{i+1}}\\
&< \frac{p_i x_{i+1}(k)-p_{i+1}x_i(k)+p_i(1-\tilde{x}^i_{i+1}(k))+p_{i+1}(1+\tilde{x}^i_i(k))}{p_i p_{i+1}}\\
&= \frac{p_i+p_{i+1}+p_i(x_{i+1}(k)-\tilde{x}^i_{i+1}(k))+p_{i+1}(\tilde{x}^i_i(k)-x_i(k))}{p_i p_{i+1}}\\
&\leqslant \frac{p_i+p_{i+1}}{p_i p_{i+1}}=\frac{1}{p_i}+\frac{1}{p_{i+1}}
\end{aligned}
$$

where the last inequality follows from the fact that both $x_{i+1}(k)-\tilde{x}^{i+1}_i(k)$ and $\tilde{x}^i_i(k)-x_i(k)$ are non-positive because $x\in\mathscr{X}_E$. □

A.8. *Proof of Theorem 4*

To prove the existence of a finite convergence time T, assume that for all $k\in\mathbb{N}$, $x(k)\notin\mathscr{X}_{\mathrm{P}}$ and prove the hypothesis by contradiction. Lemma 4 showed us that if $x_{i+1}(k)/p_{i+1}-x_i(k)/p_i$ is less than $1/p_i+1/p_{i+1}$ then so is $x_{i+1}(k+T)/p_{i+1}-x_i(k+T)/p_i$ for all $T\in\mathbb{N}$ (i.e. once a node meets the local criterion of $\mathscr{X}_{\mathrm{P}}$, it continues to do so from that point on). Thus, for our assumption to be true, there must be at least one node i such that $x_{i+1}(k)/p_{i+1}-x_i\times(k)/p_i\geqslant 1/p_i+1/p_{i+1}$ for all k. This implies $\tilde{x}^i_i(k)/p_i<\tilde{x}^i_{i+1}(k)/p_{i+1}$ for all k and this in turn implies that $\alpha(i)=0$ for all k (i.e. coordinator i will never transfer a vehicle from area i to area

$i+1$ because it always thinks (correctly) that the weighted number of vehicles in area $i+1$ exceeds that of area i).

Now since $x(k) \in \mathscr{X}_L$, we have an upper bound of L for $x(k)$, which means that there exists $T_{i-1} < \infty$ such that $\alpha(i-1) = 0$ for all $k > T_{i-1}$. If not, then $x_i(k)$ would go to infinity as vehicles continued to be added to area i without being removed by coordinator i. Similarly, there must exist some time $T_{i-2} < \infty$ such that $\alpha(i-2) = 0$ for all $k > T_{i-2}$ if $x_{i-1}(k)$ is to remain bounded after time T_{i-1}. Indeed, there must be some time $T_0 < \infty$ such that $\alpha(i) = 0$ for all $i \in \mathscr{N}$ and all $k > T_0$. In other words, since coordinator i never transfers any vehicles around the ring, there must be some finite time after which all coordinators cease to transfer vehicles.

Using the conditions that define the allowed trajectories E_B, if coordinator $i+1$ does not transfer any vehicles of area i for B time steps and coordinator $i-1$ does not transfer any vehicles into area i during the same period, then coordinator i's estimate of the number of vehicles in area i and area $i+1$ will be correct. Therefore, after time $T_0 + B$ all the estimated values in $x_C(k)$ will be correct and the only way for $\alpha(i)$ to be zero for all i and all $k > T_0 + B$ is to have $x_i(k)/p_i \leqslant x_{i+1}(k)/p_{i+1}$ for all i. However, it is a fact that if no element of a set of real numbers is greater than the others, then they must all be equal. This implies that $x_{i+1}(k)/p_{i+1} - x_i(k)/p_i = 0$ for all $i \in \mathscr{N}$ and all $k > T_0 + B$ which is less than $1/p_i + 1/p_{i+1}$. This means $x(k) \in \mathscr{X}_P$ for all $k > T_0 + B$, which contradicts our assumption. Hence, there must exist some $T < \infty$ such that $x(T) \in \mathscr{X}_P$ (and $x(k) \in \mathscr{X}_P$ for all $k \geqslant T$ follows from Lemma 1). This completes the proof that a finite convergence time exists.

In order to derive a bound on the value of T we will require a few more definitions. Let a group of nodes $I \subset \mathscr{N}$ that satisfies both the properties $x_{i+1}/p_{i+1} - x_i/p_i < 1/p_i + 1/p_{i+1}$ for all $i \in I$ and the property that there exists a unique node $i \in I$ such that $i+1 \notin I$ be referred to as a *block*. Simply speaking, a block is a group of consecutive areas that satisfy the local condition used in the definition of $\mathscr{X}_P$ and which is bounded on the left and right by an area which does not (using the same convention for direction as in Figure 1). We will refer to the blocks that contain the areas $i \in \mathscr{N}$ satisfying $x_i/p_i > x_{i+1}/p_{i+1}$ as *active* blocks, which is to say that there is at least one coordinator for an area in those blocks that will eventually transfer a vehicle to the next area in the ring. Blocks without such an area are referred to as *inactive*. Let us also refer to the area to the immediate right of a block as its *leading edge*.

We have already shown that nodes satisfying $x_{i+1}/p_{i+1} - x_i/p_i < 1/p_i + 1/p_{i+1}$ at a particular time continue to do so from then on, and thus blocks do not decrease in size. Active blocks can expand to the right by raising the number of vehicles in their leading edge area until it becomes part of the block, but active blocks may also become inactive before this happens. Inactive blocks cannot become active again by themselves but may merge with an active block on their left. So long as $x(k) \notin \mathscr{X}_P$, there must be one or more active blocks (because $x(k) \notin \mathscr{X}_P$ implies $x_{i+1}/p_{i+1} < x_i/p_i$ for at least one area i) and these blocks must grow and merge until $x(k)$ enters $\mathscr{X}_P$ (because they must keep transferring vehicles to the right and onto their leading edges).

Let $I(k)$ denote the evolution of a block that remains active for all k such that $x(k) \notin \mathscr{X}_P$. In order to get a bound on the convergence time of the system model with proportional balancing, we need to know how long it can take before this block has transferred enough vehicles to its leading edge to turn that area into part of the block. Let us define a value $G = \max_i \lceil 1 + p_i/p_{i+1} \rceil$ to be a worst-case *storage limit* for nodes within a block.

DOI: 10.1002/rnc

To illustrate the importance of this value, let us take a node $i \in I(k)$ and show what results when G vehicles are added to it,

$$\begin{aligned}
&\frac{x_{i+1}(k)}{p_{i+1}} - \frac{x_i(k)}{p_i} < \frac{1}{p_i} + \frac{1}{p_{i+1}} \\
&\quad \Rightarrow \frac{x_i(k)}{p_i} > \frac{x_{i+1}(k)}{p_{i+1}} - \frac{1}{p_i} - \frac{1}{p_{i+1}} \\
&\quad \Rightarrow \frac{x_i(k)+G}{p_i} > \frac{x_{i+1}(k)}{p_{i+1}} - \frac{1}{p_i} - \frac{1}{p_{i+1}} + \frac{G}{p_i} \\
&\quad = \frac{x_{i+1}(k)}{p_{i+1}} - \frac{1}{p_i} - \frac{1}{p_{i+1}} + \frac{\max_i \left\lceil 1 + \dfrac{p_i}{p_{i+1}} \right\rceil}{p_i} \\
&\quad \geqslant \frac{x_{i+1}(k)}{p_{i+1}} - \frac{1}{p_i} - \frac{1}{p_{i+1}} + \frac{1 + \max_i \dfrac{p_i}{p_{i+1}}}{p_i} \\
&\quad \geqslant \frac{x_{i+1}(k)}{p_{i+1}}
\end{aligned}$$

And so we see that G is the minimum number of vehicles that must be transferred to an area i in a block before we are guaranteed that it will (eventually) have to pass a vehicle on to area $i+1$. Since this holds for all areas in a block, we can see that it may take the transfer of G^2 vehicles to area i in order to eventually get coordinator $i+1$ to transfer one vehicle to area $i+2$. Similarly, in order to add one vehicle to area $i+n$, it might be the case that G^n vehicles must be added to node i.

The above information will help us bound the maximum length of time that may pass before the size of $I(k)$ must increase by at least one node. Let i^* be the leading edge of block $I(k)$ and let $n(k) = |I(k)|$ be its length. Because of the constant load condition imposed by $x(k) \in \mathcal{X}_L$, the imbalance $x_{i^*+1}/p_{i^*+1} - x_{i^*}/p_{i^*}$ can be no more than L/p_{i^*+1} (i.e. if all the vehicles were on node i^*+1). Therefore, no more than

$$\left\lceil \frac{\dfrac{L}{p_{i^*+1}} - \dfrac{1}{p_{i^*}} - \dfrac{1}{p_{i^*+1}}}{p_{i^*}} \right\rceil$$

vehicles must be transferred to area i^* in order for it to become part of the block. For simplicity we over bound this value as follows:

$$\left\lceil \frac{\dfrac{L}{p_{i^*+1}} - \dfrac{1}{p_{i^*}} - \dfrac{1}{p_{i^*+1}}}{p_{i^*}} \right\rceil \leqslant \left\lceil \frac{L-1}{p_{i^*}p_{i^*+1}} \right\rceil < \left\lceil \frac{L}{\min_i(p_i p_{i+1})} \right\rceil$$

Now assume that all of these vehicles have to come from the far left side of the block, i.e. from area $i^* - n(k)$. In accordance with the previous discussion, coordinator $i^* - n(k)$ may have to transfer as many as $G^{n(k)-1}$ vehicles to node $i^* - n(k) + 1$ in order to eventually add one vehicle to area i^*, and so it may have to transfer $G^{n(k)-1}\lceil L/\min_i(p_i p_{i+1})\rceil$ vehicles in order to eventually

DOI: 10.1002/rnc

make area i^* part of the block. If node $i-n(k)$ transfers vehicles at the minimum rate of one per $2B$ time steps (i.e. B time steps to guarantee its estimates are correct (and thus permit the transfer of vehicles) and another B time steps to actually transfer a vehicle), than it could take $2BG^{n(k)-1}\lceil L/\min_i(p_i p_{i+1})\rceil$ time steps to get all the necessary vehicles from node $i^*-n(k)$ to node $i^*-n(k)+1$. Making the simplifying assumption that all of these vehicles must arrive at area $i^*-n(k)+1$ before coordinator $i^*-n(k)+1$ starts transferring any of the $G^{n(k)-2}\lceil L/\min_i \times (p_i p_{i+1})\rceil$ vehicles it must transfer into area $i^*-n(k)+2$, we can see that the time it takes before node i^* joins the block can be bounded from above by

$$\begin{aligned}
&2B\left\lceil\frac{L}{\min_i(p_i p_{i+1})}\right\rceil G^{n(k)-1}+2B\left\lceil\frac{L}{\min_i(p_i p_{i+1})}\right\rceil G^{n(k)-2}+\cdots+2B\left\lceil\frac{L}{\min_i(p_i p_{i+1})}\right\rceil\\
&=2B\left\lceil\frac{L}{\min_i(p_i p_{i+1})}\right\rceil(G^{n(k)-1}+G^{n(k)-2}+\cdots+1)\\
&=2B\left\lceil\frac{L}{\min_i(p_i p_{i+1})}\right\rceil\frac{G^{n(k)-1}}{G-1}
\end{aligned}$$

If we now make another extremely conservative assumption that it takes this long for a single block to grow from one area to two, two areas to three, and so on until it goes from $N-1$ area to N (and ceases to be a block because $x(k)$ is then in $\mathscr{X}_{\mathrm{P}}$), then the convergence time of the algorithm can be bounded from above as follows:

$$\begin{aligned}
T&\leqslant 2B\left\lceil\frac{L}{\min_i(p_i p_{i+1})}\right\rceil\left(\frac{G-1}{G-1}+\frac{G^2-1}{G-1}+\cdots+\frac{G^{N-1}-1}{G-1}\right)\\
&=\frac{2B}{G-1}\left\lceil\frac{L}{\min_i(p_i p_{i+1})}\right\rceil(G+G^2+\cdots+G^{N-1}-(N-1))\\
&=\frac{2B}{G-1}\left\lceil\frac{L}{\min_i(p_i p_{i+1})}\right\rceil\left(\frac{G(G^{N-1}-1)}{G-1}-(N-1)\right)\\
&<\frac{2B(G^N-G)}{(G-1)^2}\left\lceil\frac{L}{\min_i(p_i p_{i+1})}\right\rceil
\end{aligned}$$

Where the $N-1$ term in the second to last line is dropped for simplicity because it is relatively small in comparison to $(G^N-G)/(G-1)$ for even modest values of N (since $G\geqslant 2$). This completes the derivation of a specific bound on the convergence time of the system. □

ACKNOWLEDGEMENTS

This work was supported by the AFRL/VA and AFOSR Collaborative Center of Control Science (Grant F33615-01-2-3154). In addition, the work was supported by a DAGSI Fellowship to B. Moore. The authors would like to thank Corey Schumacher of AFRL for his input.

REFERENCES

1. Schumacher C, Chandler PR, Rasmussen SJ. Task allocation for wide area search munitions via iterative network flow. *AIAA Guidance, Navigation, and Control Conference*, no. 2002–4586, Monterey, CA, 2002.

2. Liu Y, Cruz JB, Sparks AG. Coordinating networked uninhabited air vehicles for persistent area denial. *43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004; 3351–3356.
3. Frazzoli E, Bullo F. Decentralized algorithms for vehicle routing in a stochastic time-varying environment. *43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004, 3357–3363.
4. Cortés J, Martinez S, Bullo F. Coordinated deployment of mobile sensing networks with limited-range interactions. *43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, December 2004; 1944–1949.
5. Bertsekas DP, Tsitsiklis JN. *Parallel and Distributed Computation*: *Numerical Methods*. Athena Scientific: Bellmont, MA, 1997.
6. Burgess KL, Passino KM. Stability analysis of load balancing systems. *International Journal of Control* 1995; **61**:357–393.
7. Finke J, Passino KM, Sparks A. Cooperative control via task load balancing for networked uninhabited autonomous vehicles. *42nd IEEE Conference on Decision and Control*, Maui, HI, December 2003; 31–36.
8. Cortés A, Ripoll A, Cedó F, Senar M, Luque E. An asynchronous and iterative load balancing algorithm for discrete load model. *Journal of Parallel and Distributed Computing* 2002; **62**:1729–1746.
9. Els R, Monien B. Load balancing of unit size tokens and expansion properties of graphs. *SPAA '03*: *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures*. ACM Press: New York, 2003; 266–273.
10. Henrich D. The liquid model load balancing method. *Journal of Parallel Algorithms and Applications, Special Issue on Algorithms for Enhanced Mesh Architectures* 1996; **8**:285–307.
11. Burgess KL, Passino KM. *Stability Analysis of Discrete Event Systems*. Wiley: New York, NY, 1998.
12. Balinski ML, Young HP. *Fair Representation: Meeting the Ideal of One Man*, *One Vote*. Yale University Press: New Haven, CT, 1997.
13. Huntington EV. A new method of apportionment of representatives. *Quarterly Publications of the American Statistical Association* 1921; **17**:859–870.
14. Burt OR, Harris CC. Apportionment of the U.S. House of Representatives: a minimum range, integer solution, allocation problem. *Operations Research* 1963; **11**:648–652.

DOI: 10.1002/rnc