Optimizing the Transportation System's Response Capabilities

February 7, 2011

Christos G. Cassandras
Related research supported in part by NSF and DOE
Department of Electrical and Computer Engineering
and Division of Systems Engineering
Boston University
Head, Division of Systems Engineering, and
Professor of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
617-353-7154
cgc@bu.edu

Ioannis Ch. Paschalidis
Related research supported in part by NSF and DOE
Department of Electrical and Computer Engineering
and Division of Systems Engineering
Boston University
Professor of Electrical, Computer, and Systems Engineering
Co-Director, Center for Information and Systems Engineering
8 Saint Mary's Street
Boston, MA 02215
617-353-0434
yannisp@bu.edu

Abstract

For the purposes of post-disaster response and recovery we view the transportation system as a network whose nodes are transportation hubs and whose links correspond to various transportation modes. We focus on: (a) transport resources (e.g., first responders, supplies) from "depots" to nodes in need, and (b) evacuating people. Regarding (a), we present a Cooperative Receding Horizon (CRH) methodology whereby, rather than trying to plan out vehicle trajectories over a long time horizon, we only consider states reached at the end of a limited time window. The controller can hedge against uncertainty and react to emerging events. Regarding (b), we present techniques suitable for networks operating in regimes where nodes are accumulating entities (e.g., people) and the objective is to "empty" the network by transporting them to specific destinations. Our policies provide guarantees on worst case delay and enable cooperation among evacuation vehicles.

Keywords: Post-disaster response, controlled evacuation, transportation, resource provisioning.

Optimizing the Transportation System's Response Capabilities

1 Introduction

For the purposes of a post-disaster response and recovery scenario we view the transportation system as a network whose nodes are major transportation hubs (e.g., train stations, ports, airports, depots, major intersections) or destination points (e.g., places where a disaster has occurred) which need to be "visited" in order to perform some function (e.g., deliver resources or remove undesirable material.) The links of the network correspond to the various modes of transportation between the nodes (e.g., highways, railroads, airplanes). The links are traversed by various types of vehicles depending on the "assets" the transportation system has at its disposal. Whereas the normal operation of this system is in "steady-state," the aftermath of a disaster introduces a variety of "transient phenomena" to which the system must respond following policies that are markedly different from its normal operating mode. We will focus on two main post-disaster response functions: (a) transport resources (e.g., first responders, supplies) from "depots" to nodes where they are needed, and (b) evacuating people from the affected areas.

Regarding (a), these tasks must occur in a way that minimizes time of delivery, while maximizing the benefit of a particular task over another. This gives rise to optimization problems with two characteristics distinguishing them from standard transportation problems: (i) the need for cooperation among the vehicles so as to maximize the service provided by the group (instead of an individual member), and (ii) the dynamic nature of the problem, in the sense that new requests arise during the execution of present tasks. These are all routing and scheduling problems with explosive combinatorial complexity, yet controllers are required to calculate commands in real time. To that end, we have developed a **Cooperative Receding Horizon** (CRH) methodology (first introduced in [Li and Cassandras, 2006]) which will be further described in this paper. Rather than attempting to devise an optimal plan for all vehicle trajectories over a long time horizon, the basic CRH idea is to consider states the system can reach at the end of a limited time window called the "planning horizon." A CRH controller selects a state at the end of the planning horizon which optimizes a given objective function in an expected value sense, thus hedging against uncertainty and reacting to future events as they arise. This is why it is referred to as a "hedge-and-react" approach, as opposed to the more traditional "estimate-and-plan" methods.

Regarding (b), we will present techniques we have developed for the study of networks operating in regimes where nodes have accumulated large amounts of entities (e.g., people) and the objective is to "empty" the network by transporting these entities to specific destinations. We will consider two distinct scenarios: a "static" and a "dynamic" one depending on the urgency for the evacuation. The static scenario corresponds to an orderly process while the dynamic scenario considers a very volatile situation where evacuees need to leave the affected area as soon as possible. Our static evacuation policies provide guarantees on worst case delay and allow for cooperation among vehicles, essentially assigning each congested node to a particular vehicle. Our dynamic policies also leverage vehicle cooperation and solve a very hard dynamic optimization problem suffering from the well known curse-of-dimensionality using recently developed approximate dynamic programming techniques.

2 The Cooperative Receding Horizon Response Approach

We view the transportation system as a network whose nodes are either **bases** or **target points** interconnected by links traversed by different vehicle types referred to as **vehicles**. A base may be a transportation hub where resources can be acquired or deposited or where vehicles are stored when not required or when needing maintenance and refueling. A target point is any destination of interest for one or more vehicles. We assume that there are B bases, M vehicles, and N target points with the understanding that these numbers may change over time. Associated with a target point i is a **reward** $R_i(t)$ which represents the importance of this point relative to others at any given time t. A similar reward is associated with any base, but it is a function of both the base b and a vehicle j and denoted by $R_{bj}(t)$; this captures, for instance, the importance of a vehicle reaching a base for refueling purposes. It is often convenient to write $R_i(t)$ as $R_i\phi_i(t)$ where R_i is an initially assigned reward and $\phi_i(t) \in [0,1]$ is a discounting function which describes the reward change over time. Thus, by appropriately selecting $\phi_i(t)$, it is possible to capture timing constraints such as assigning a deadline to a target point beyond which the reward is lost, as well as precedence constraints. Finally, by allowing $R_i < 0$ for some i and properly selecting $\phi_i(t)$, we may also model an obstacle or threat in the mission space (see also [Li and Cassandras, 2004]).

A **mission** is defined as the process of controlling the movement of the vehicles and ultimately assigning them to target points so as to maximize the total reward collected by visiting target points within a given mission time T or until a specified condition is satisfied. Note that the goal of the mission does not involve rewards collected from bases; such rewards affect only indirectly the state of a vehicle, as explained later. This setting gives rise to a complex stochastic optimal control problem. One can invoke dynamic programming as a solution approach [Bertsekas, 1995, Puterman, 1994, Cao, 2007], but this is computationally intractable even for relatively simple mission control settings [Wohletz et al., 2001],[Curry et al., 2002] or, more generally, vehicle routing problems [Li et al., 2005]. Because of the complexity of the overall problem, it is natural to decompose it into various subproblems at different levels, but even so the problem remains intractable. An alternative to this "functional decomposition" approach is one based on time decomposition. The main idea is to solve an optimization problem seeking to maximize the total **expected** reward accumulated by the team over a given time horizon, and then continuously extend this time horizon forward (either periodically or in purely event-driven fashion). This idea is in the spirit of receding horizon schemes associated with model-predictive control, e.g., [Mayne and Michalska, 1990], [Dunbar and Murray, 2004], [Richards and How, 2004], [Frazzoli and Bullo, 2004]. The resulting cooperative control scheme dynamically determines vehicle trajectories by solving a sequence of optimization problems over a **planning** horizon and executing them over a shorter **action** horizon. We must emphasize that the optimization problem involved does not attempt to make any explicit vehicle-to-target assignments, but only to determine headings that, at the end of the current planning horizon, would place vehicles at positions such that a total expected reward is maximized. A somewhat surprising result [Li and Cassandras, 2006] is that vehicle trajectories automatically converge to target points, despite the fact that our approach, by its nature, was never intended to perform any such discrete vehicle-to-target-point assignment. With this seemingly salient property, an advantage of this approach is that it integrates the three tasks of (*i*) vehicle assignment to target points, (*ii*) routing of the vehicles to their assigned target points, and (*iii*) real-time trajectory generation, all in one function: controlling vehicle headings in real time. In what follows, we outline the CRH approach.

The location of the ith target point is $y_i \in \mathbb{R}^2$, and that of the jth vehicle at time t is $x_j(t) \in \mathbb{R}^2$. In order to distinguish the effectiveness of vehicles relative to a target point i, we define a **capability factor** $p_{ij}(t) \in [0,1]$, which reflects the probability that a vehicle j visiting point i at time t will complete the task and collect the reward $R_i\phi_i(t)$. In practice, this is a function of the vehicle state at time t. For example, in an evacuation scenario, $p_{ij}(t)$ may represent the fraction of individuals that vehicle i can carry when visiting target point j; in a resource delivery scenario, $p_{ij}(t)$ may represent the fraction of required resources that vehicle i can deliver to target point j. If during a visit to target point j a vehicle i successfully completes its task, it will collect the corresponding reward and, at the same time, target point i is no longer of interest to the mission and it is removed from consideration. Since a visit at point i is related to the consumption of vehicle j's resources, its capability factor $p_{ij}(t)$ may decrease after the visit. This justifies the use of rewards $R_{bj}(t)$ at bases which allows vehicles to make decisions on whether to visit a base as opposed to a target point as part of its mission trajectory. In addition, if while visiting i vehicle j is damaged, a negative reward is collected, and j is removed from the set of available vehicles or sent to a base.

The cooperative structure of a mission is manifested by vehicles dynamically partitioning the mission space and implicitly allocating regions of it among themselves. Given an arbitrary point $y \in \mathbb{R}^2$ in the mission space (not necessarily a target point), we would like to assign this point to vehicles at time t so that y is assigned to the closest vehicle with the highest probability. To formalize this idea, we define a **neighbor set** $\mathcal{N}^r(y,t)$ to include the r closest vehicles to $y \in \mathbb{R}^2$ at time t, where $r \in \{1,\ldots,M\}$. Let $N^l(y,t)$ be the lth closest vehicle to y so that $\mathcal{N}^r(y,t) = \{N^1(y,t),\ldots,N^r(y,t)\}$. We then define the **relative distance function**, $\delta_j(y,t)$, as follows:

$$\delta_j(y,t) = \begin{cases} \dfrac{\|x_j(t)-y\|}{\sum_{k\in\mathcal{N}^r(y,t)} ||x_k(t)-y||} & \text{if } j \in \mathcal{N}^r(y,t) \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

where of particular interest is the case $r = 2$. Next, we define a **relative proximity function** $q_j(y,\delta_j)$ to be any nonincreasing function of δ_j such that $q_j(y,0) = 1$, $q_j(y,1) = 0$. We can view $q_{ij}(\delta_{ij})$ as the probability that target point i is assigned to vehicle j at time t, based on the value of $\delta_{ij}(t)$, and observe that $\sum_j q_{ij}(\delta_{ij}) = 1$.

Clearly, in a ground transportation network, we cannot assume that the mission space is modeled in a plane and must instead operate on a given network defined by roads and/or railroad tracks and modeled through a graph. Our setting allows for such a graph-based topology, as seen in a specialized transportation problem involving elevator systems, where elevator cars are obviously constrained to move under specific constraints [Wesselowski and Cassandras, 2006]. In what follows, however, we limit ourselves to a plane so as to describe the basic ideas involved. Then, we assume a vehicle travels at constant velocity throughout the team mission, i.e.,

$$\dot{x}_j(t) = V_j \begin{bmatrix} \cos u_j(t) \\ \sin u_j(t) \end{bmatrix}, \quad x_j(0) = x_{j0} \tag{2}$$

where $u_j(t) \in [0, 2\pi]$ is the controllable heading of vehicle j and V_j is the corresponding velocity.

The CRH controller we design generates a set of trajectories for each vehicle in the team over a given mission time T. This controller is applied at time points denoted by t_k, $k = 0,1,\ldots$, during the mission time. At t_k, the controller operates by solving an optimization problem $\mathbf{P}_k$, whose solution is the control vector $\mathbf{u}_k = [u_1(t_k)\ldots u_M(t_k)]$. Next, we explain how $\mathbf{P}_k$ is formulated. Suppose that vehicles are assigned headings $u_1(t_k),\ldots,u_M(t_k)$ at time t_k, intended to be maintained for a planning horizon denoted by H_k. Then, at time $t_k + H_k$ the planned positions of the vehicles are given by $x_j(t_k + H_k) = x_j(t_k) + \dot{x}_j(t_k)H_k$. Define

$$\tau_{ij}(\mathbf{u}_k, t_k) = (t_k + H_k) + ||x_j(t_k + H_k) - y_i||/V_j \tag{3}$$

and note that $\tau_{ij}(\mathbf{u}_k, t_k)$ is the earliest time that vehicle j can reach point i under the condition that it starts at t_k with control dictated by $\mathbf{u}_k$ and then proceeds directly from $x_j(t_k + H_k)$ to the target point y_i. We are interested in the maximal reward that vehicle j can extract from target i if it reaches it at time $\tau_{ij}(\mathbf{u}_k, t_k)$. For convenience, define $\tilde{\phi}_{ij}(\mathbf{u}_k, t_k) = \phi_i[\tau_{ij}(\mathbf{u}_k, t_k)]$. Similarly, we define $\tilde{p}_{ij}(\mathbf{u}_k, t_k) = p_{ij}[\tau_{ij}(\mathbf{u}_k, t_k)]$ and $\tilde{q}_{ij}(\mathbf{u}_k, t_k) = q_{ij}[\delta_{ij}(t_k + H_k)]$. We can now present the optimization problem $\mathbf{P}_k$, formulated at time t_k, as follows:

$$\max_{\mathbf{u}_k} \sum_{i=1}^{N} \sum_{j=1}^{M} R_i \tilde{\phi}_{ij}(\mathbf{u}_k, t_k) \cdot \tilde{p}_{ij}(\mathbf{u}_k, t_k) \cdot \tilde{q}_{ij}(\mathbf{u}_k, t_k) \tag{4}$$

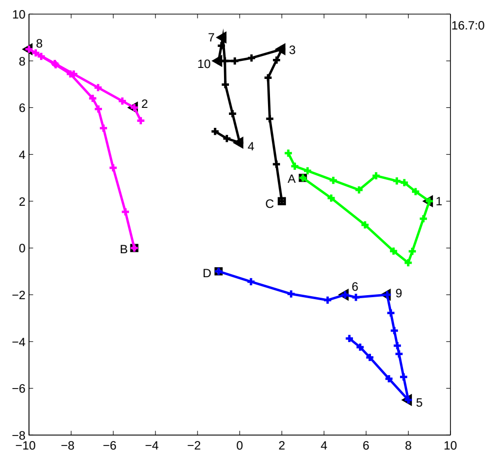

Figure 1: Example of a mission with 4 vehicles and 10 target points using CRH control.

The expression $R_i \tilde{\phi}_{ij}(\mathbf{u}_k, t_k) \cdot \tilde{p}_{ij}(\mathbf{u}_k, t_k) \cdot \tilde{q}_{ij}(\mathbf{u}_k, t_k)$ in (4) can be seen as the expected reward that vehicle j collects from target point i, evaluated at time t_k using a planning horizon H_k (we can also model the elimination of vehicles by including negative terms above).

Problem $\mathbf{P}_k$ is parameterized by the planning horizon H_k, which is critical in obtaining desirable properties for this CRH controller. In particular, selecting H_k to be the smallest "distance" (in time units) between any target point and any capable vehicle at time t_k allows us to rigorously show [Li and Cassandras, 2006] that vehicle trajectories actually converge to target points. Upon getting the optimal $\mathbf{u}_k$ for (4) based on H_k and all state information available at t_k, all vehicles follow this control for an **action horizon** $h_k \leq H_k$. The process is then repeated at time $t_{k+1} = t_k + h_k$, $k = 0, 1, \ldots$ The value of h_k is determined either by an unexpected event (e.g., a vehicle fails) or as a prespecified amount of time. The CRH controller terminates when a given condition is satisfied, e.g., all the target rewards are collected. An example of a mission executed under CRH control is shown in Fig. 1 where the crossmarks indicate points where the controller is invoked to adjust vehicle headings.

Problem $\mathbf{P}_k$ is readily extended to incorporate additional features of a specific team mission environment (see also [Li and Cassandras, 2004],[Yao et al., 2010]). Thus, for disaster response missions in a given transportation system, our goal is to specify the environment and develop an appropriate version of this problem that incorporates the features already mentioned, including: (i)

Rewards associated with multiple bases which are not part of the work done to date, (ii) Constraints on the controllable heading vectors $\mathbf{u}_k$ at the kth decision point which are a consequence of the transportation network's graph topology, (iii) The nature of the tasks to be performed in a mission, which results in how rewards and capability factors are defined, as well as the possible need for "rendez vous" of multiple vehicles at some target points or bases, similar to the approach used in [Yao et al., 2010]. It is also obviously crucial to ensure that any approach intended for real-time execution is **scalable**. As shown in [Li and Cassandras, 2006], the complexity of the CRH approach is independent of the number of target points N and of order G^M overall, where M is the number of vehicles and G is the number of feasible headings (which in a graph setting is generally small.) We also note that another potentially attractive approach to solving complex problems of the type described here is to use Approximate Dynamic Programming (ADP) techniques as discussed in Section 3.2 for an evacuation problem formulated as a type of reward maximization mission similar to the one we have considered for the emergency response problem.

3 Controlled Evacuation

In this section we discuss approaches for evacuating an affected area from individuals, transporting them to temporary shelters where they can be housed and provided appropriate care.

As before, we model the affected area as a **network** whose **nodes** correspond to locations where the affected individuals can be gathered and whose **links** correspond to the roads, rail lines, and other means the transportation system can use for the evacuation. We consider two different scenarios:

1. **Scenario I – a static view**. In the first scenario we assume that the affected area is stable and under control and poses no significant harm to people. Homes and other structures have been damaged (e.g., earthquake, hurricane) but people can gather as instructed at pre-specified points – the nodes – and evacuated from those nodes. We also assume that people gather at a constant rate at those nodes and have to be transported to some other node(s) of the network that corresponds to a shelter(s). The various resources of the transportation system – the vehicles – are tasked with visiting the nodes, loading the people, and delivering them to the intended destination(s), potentially by transferring them to another vehicle if that results in a more efficient evacuation. A basic question we address is how to **design effective routes and coordination strategies for the vehicles**, given node positions and the rates at which people gather at those nodes. We will consider schemes that utilize multiple vehicles so that one can scale the size of the network to be served. We are interested in bounds on the

time required to transport an individual from a node to the shelter(s). Obtaining such bounds would allow us to offer **performance guarantees**. Another key concern we tackle is the **scalability** of the proposed schemes.

2. **Scenario II – a dynamic view**. The second scenario we consider is one where the affected area is not stable and not considered safe for the affected individuals (e.g., chemical, biological, or nuclear release). Furthermore, the rate at which individuals gather at the nodes is highly variable. In this setting, we propose to study a scheme according to which the vehicles respond to evacuation requests as they emerge. We will assume that the nodes have the capability to wirelessly transmit such requests over a relative long range (using an emergency response communication system) and the vehicles the ability to listen to them. We propose a framework involving dynamic programming and we will concentrate on **distributed approximate dynamic programming** methods that are tractable and appropriate for our setting.

3.1 Scenario I – a static view

The approach we propose leverages our recent work in [Paschalidis and Moazzez-Estanjini, 2010]. We consider a network with N nodes. We assume that at node i individuals that need to be transported to node j gather at a rate of $\lambda(i,j)$. In this subsection we take a **static view** and assume that the $\lambda(i,j)$'s are known (estimated) with certainty and are constant over time. Assume also that we have M vehicles responsible for transporting the individuals. The objective is to design the vehicles' routes such that the delay in evacuating people is minimized.

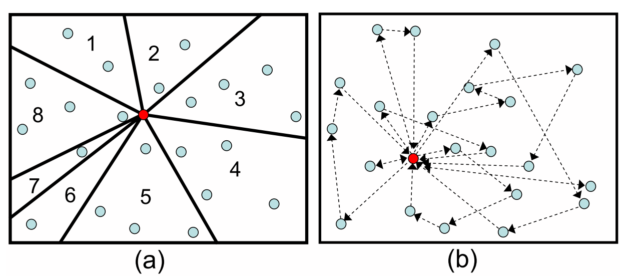

Figure 2: Examples of partitioning a network into 8 segments (the blue circles represent the Snodes, and the red circle represents the Mnode contact point); (a) "Contiguous" partitioning (b) "Non-contiguous" partitioning.

In preliminary work we developed a scheme we call **Centralized Vehicle Relaying (CVR)**. Fig. 2(a) depicts a simple geographic partitioning of the entire network. There is a single vehicle assigned to each segment. Within each **cycle**, each vehicle collects the people gathered at the nodes. Later in the same cycle, all vehicles meet at some pre-determined **contact point** and transfer among themselves the individuals they have gathered. Each vehicle's route consists of a **Hamiltonian cycle** [1] that includes all its assigned nodes and the contact point. The partition-

[1]A Hamiltonian cycle is a cycle in an undirected graph which visits each node exactly once and returns to the starting node.

ing presented in Fig. 2(a) has been used in prior work and is a "contiguous" partitioning in the sense that each segment assigned to a vehicle is geographically contiguous. However, in our CVR scheme we allow the segments to be "non-contiguous." This allows assigning any combination of nodes to any vehicle and results in greater flexibility (see Fig. 2(b)).

We have analyzed the delay achieved by CVR. One important concern is the **scalability** of the proposed scheme. Our scalability analysis in [Paschalidis and Moazzez-Estanjini, 2010] relies on bounding inter-vehicle transfer times and leverages graph-theoretic results. We are able in [Paschalidis and Moazzez-Estanjini, 2010] to obtain an upper bound on the **maximum** and **average** delay needed to transfer one individual from the node she/he was picked up to the destination node.

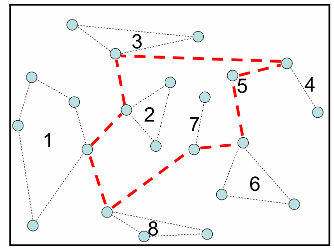

Figure 3: An example illustrating the BVR scheme - the blue circles represent the nodes, the black lines show the routes of 8 vehicles, and the red dashed line shows the route of the bus.

The bound on maximum delay is quite useful as it provides a **performance guarantee**; such guarantees have not been available for any scheme so far in the related literature. Moreover, and this is our main result, we can show that if the number of vehicles scales linearly with the number of nodes then **constant** (in M and N) **per node throughput with constant worst-case delay is achievable** using the CVR scheme.

Several avenues for future work remain unexplored. We plan to: (i) explore improvements to CVR; and (ii) devise and analyze alternatives to CVR vehicle coordination schemes that may be more appropriate for particular emergency situations.

With respect to goal (ii) we propose below a different scheme we plan to explore and analyze. We call this scheme **Bus Vehicle Relaying (BVR)**. Fig. 3 shows an example illustrating the BVR scheme with a single bus. In this scheme, N nodes are assigned to $(M-1)$ vehicles and people are transferred among vehicles via the remaining vehicle which is called **the bus**. The bus meets with each vehicle at a contact point exactly once in each cycle. As with CVR we would like to analyze BVR, obtain bounds on delay, and devise methods to optimally configure it.

3.2 Scenario II – a dynamic view

Next we consider scenarios where the process according to which people gather at the nodes is **bursty**. As a result, "servicing" these nodes by a scheme that is periodic and designed for an almost constant accumulation rate is not efficient.

We have explored problems of this type in [Pennesi and Paschalidis, 2007] and [Pennesi and Paschalidis, 2010]. In particular, and similarly to Sec. 2, we considered a fairly general class of coordination problems formulated as **reward collection** problems. Assume a

2-dimensional **coverage area**, $\mathcal{S}$, in which there is a set of N nodes whose positions are indicated, at time k, by $\mathbf{m}_k^i \in \mathcal{S}$, for $i = 1, \ldots, N$. To each node i we associate a **reward** $R_k^i \in \mathbb{R}_+$. The reward essentially reflects the amount of people gathered at that node and the urgency to evacuate them.

The coverage area is being explored by M vehicles whose positions at time k are indicated by $\mathbf{x}_k^j \in \mathcal{S}$. To each vehicle j we associate a capacity $C_k^j \in \mathbb{R}_+$ (e.g., reflecting physical capacity and energy constraints of the vehicle). When a vehicle approaches a node it collects a reward which depends on the available reward at the node and the capacity of the vehicle. Every reward collection has also the effect of depleting a part of the vehicle's capacity. Every vehicle j **navigates** in $\mathcal{S}$ and, from time to time, returns to a **depot** (e.g., at the origin) which has the effect of replenishing the vehicle's capacity to its initial value C_0^j (e.g., by uploading the people it carries and refueling).

The scenario we are considering is completed by describing how vehicles get informed about the available reward at nodes. We assume that the nodes have the ability to transmit short messages directly to vehicles within some range, or potentially, via other nodes that act as relays. As a result, vehicle j receives information on the amount of reward available at node i within some distance from it and can compute an "intensity signal," denoted by $s_k^i(R_k^i, \mathbf{m}_k^i, \mathbf{x}_k^j)$. This signal reflects the desirability of approaching node i and can, for instance, be proportional to the reward R_k^i at that node and inversely proportional to the distance $\|\mathbf{x}_k^j - \mathbf{m}_k^i\|$.

Given this setup we are interested in a policy that guides the vehicles in the coverage area to maximize the long-term average total reward collected by them. Note that this is similar to the reward maximization missions of Sec. 2. The problem at hand can be formulated as a **Dynamic Programming** problem [Bertsekas, 1995, Puterman, 1994, Cao, 2007]. Decisions involve how each vehicle should move in the coverage area and the state includes the position of each vehicle. Even if each vehicle acts independently, the state space would be as large as the coverage area which can be enormous. It follows that standard DP algorithms have no chance of tackling realistic instances. To combat Bellman's so called curse of dimensionality, **Approximate Dynamic Programming (ADP)** (see, e.g., [Bertsekas and Tsitsiklis, 1996, Powell, 2007]) is particularly useful.

In [Pennesi and Paschalidis, 2010] we started from a particular ADP approach which is appropriate for a **Markov Decision Process (MDP)** where decisions are made by a single agent. In particular, we considered the **actor-critic algorithms** developed in [Konda and Tsitsiklis, 2003]. In these algorithms one adopts a randomized class of policies parametrized by a (low-dimensional) parameter vector $\boldsymbol{\theta}$ and optimizes policy performance with respect to $\boldsymbol{\theta}$ by using a simulation (or a realization) of the MDP.

Our main contribution in [Pennesi and Paschalidis, 2010] is that we developed a **Distributed**

Multi-agent Actor-Critic (D-AC) algorithm. Our algorithm allows us to use multiple agents (vehicles for the application we are considering) to simultaneously explore the state-control space. Each vehicle maintains its own $\boldsymbol{\theta}$ and updates it based on local information and information received from a subset of other vehicles (e.g., the ones within a certain communication range). This updating follows a **consensus-like algorithm**; such algorithms and their analysis go back to [Tsitsiklis, 1984] and have garnered renewed interest [Jadbabaie et al., 2003, Blondel et al., 2005]. Under suitable conditions, we show that all agents reach consensus and converge to the optimal $\boldsymbol{\theta}$. In the algorithm we presented in [Pennesi and Paschalidis, 2010], agents update their $\boldsymbol{\theta}$'s **asynchronously**, which is appropriate for our setting.

The D-AC algorithm provides a useful framework for vehicle coordination in dynamic environments. What is particularly appealing is that we solve a dynamic problem benefiting from the parallel exploration of the state-control space while incurring a relatively small communication overhead (vehicles only exchange their $\boldsymbol{\theta}$'s). We note that even though many less complex schemes have been proposed for agent coordination (e.g., they tend to be heuristic or solve a sequence of static problems [Li and Cassandras, 2005]). Our scheme attempts to approximately solve an MDP using an ADP method and it fills a void since there has not been much attention in the ADP literature on distributed approaches.

There are several directions we plan to explore in our future work:

- **3-dimensional coverage areas**. First, there is nothing special about coverage areas being subsets of $\mathbb{R}^2$. We plan to extend the approach we have already developed to $\mathbb{R}^3$ which will allow us to deal with scenarios where nodes and airborne vehicles are deployed in mountainous areas and their location includes elevation.

- **Transportation infrastructure**. The work in [Pennesi and Paschalidis, 2010] discretizes the coverage area and assumes that vehicles move on a grid. More generally, there may be an underlying transportation network and vehicles may be required to move along existing routes. In that case, one can think of a **transportation graph**, where nodes represent node locations and links correspond to various routes connecting the nodes (see also the setup of Sec. 2). The vehicles are confined to moving on this graph. We expect that similar techniques to our D-AC algorithm would apply but the structure of the policy may have to change. In particular, distance between two nodes can no longer be measured using a Euclidean distance but other distance metrics (e.g., hop-count, Manhattan distance) may be appropriate.

4 Conclusions

In post-disaster response and recovery, arguably, two critical functions transportation can perform are: (a) to provide to the affected areas and first responders the necessary supplies and resources needed to control the situation, and (b) to evacuate affected individuals to shelters where they can be more effectively cared for. A systematic approach to implementing these functions in a highly complex and uncertain environment requires solving complex dynamic optimization problems with real-time capabilities. The methodologies we put forth are built on rigorous techniques that provide such capabilities, while recognizing the need for scalability and for satisfying the inherent constraints of the transportation system. The resulting schemes are dynamic in nature and are designed to provide efficient responses to uncertainties as they arise and an organized evacuation approach using delay (rather than distance) as a key optimization metric. We envision a fully integrated system that takes advantage of wireless sensor network and communication technologies in implementing these ideas for the use of emergency response agencies (at the local and federal level), city planners, and local transportation authorities.

References

[Bertsekas and Tsitsiklis, 1996] Bertsekas, D. and Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

[Bertsekas, 1995] Bertsekas, D. P. (1995). *Dynamic Programming and Optimal Control*, volume I and II. Athena Scientific, Belmont, MA.

[Blondel et al., 2005] Blondel, V., Hendrickx, J., Olshevsky, A., and Tsitsiklis, J. (2005). Convergence in Multiagent Coordination, Consensus, and Flocking. In *44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, pages 2996–3000, Seville, Spain.

[Cao, 2007] Cao, X. (2007). *Stochastic Learning and Optimization: A Sensitivity-Based Approach*. Springer-Verlag New York, Inc. Secaucus, NJ, USA.

[Curry et al., 2002] Curry, M. L., Wohletz, J. M., Castañón, D. A., and Cassandras, C. G. (2002). Modeling and Control of a Joint Air Operations Environment with Imperfect Information. In *Proc. of SPIE 16th Annual Intl. Symposium*, pages 41–51.

[Dunbar and Murray, 2004] Dunbar, W. B. and Murray, R. M. (2004). Receding Horizon Control of Multi-Vehicle Formations: A Distributed Implementation. In *Proc. of 43rd IEEE Conf. on Decision and Control*, pages 1995–2002.

[Frazzoli and Bullo, 2004] Frazzoli, E. and Bullo, F. (2004). Decentralized Algorithms for Vehicle Routing in a Stochastic Time-Varying Environment. In *Proc. of 43rd IEEE Conf. on Decision and Control*, pages 3357–3363.

[Jadbabaie et al., 2003] Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):2953–2958.

[Konda and Tsitsiklis, 2003] Konda, V. R. and Tsitsiklis, J. N. (2003). Actor-critic algorithms. *SIAM Journal on Control and Optimization*, 42(4):1143–1166.

[Li et al., 2005] Li, F., Golden, B., and Wasil, E. (2005). Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers Opns. Res.*, 32:1165–1179.

[Li and Cassandras, 2004] Li, W. and Cassandras, C. G. (2004). Centralized and Distributed Cooperative Receding Horizon Control of Autonomous Vehicle Missions. *Journal of Mathematical and Computer Modelling*, 43(9-10):1208–1228.

[Li and Cassandras, 2005] Li, W. and Cassandras, C. G. (2005). Distributed Cooperative Coverage Control of Sensor Networks. In *Proc. of 44th IEEE Conf. on Decision and Control*, pages 2542 – 2547.

[Li and Cassandras, 2006] Li, W. and Cassandras, C. G. (2006). A Cooperative Receding Horizon Controller for Multi-Vehicle Uncertain Environments. *IEEE Trans. Automatic Control*, 51(2):242–257.

[Mayne and Michalska, 1990] Mayne, D. Q. and Michalska, L. (1990). Receding Horizon Control of Nonlinear Systems. *IEEE Trans. Automatic Control*, AC-35(7):814–824.

[Paschalidis and Moazzez-Estanjini, 2010] Paschalidis, I. C. and Moazzez-Estanjini, R. (2010). The capacity of sparse networks under controlled mobility. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, Georgia.

[Pennesi and Paschalidis, 2007] Pennesi, P. and Paschalidis, I. C. (2007). Solving sensor network coverage problems by distributed asynchronous actor-critic methods. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5300–5305, New Orleans, Louisiana.

[Pennesi and Paschalidis, 2010] Pennesi, P. and Paschalidis, I. C. (2010). A distributed actor-critic algorithm and applications to mobile sensor network coordination problems. *IEEE Trans. Automat. Contr.*, 55(2):492–497.

[Powell, 2007] Powell, W. (2007). *Approximate Dynamic Programming*. John Wiley and Sons.

[Puterman, 1994] Puterman, M. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc. New York, NY, USA.

[Richards and How, 2004] Richards, A. and How, J. (2004). Decentralized Model Predictive Control of Cooperating UAVs. In *Proc. of 43rd IEEE Conf. on Decision and Control*, pages 4286–4291.

[Tsitsiklis, 1984] Tsitsiklis, J. N. (1984). *Problems in Decentralized Decision Making and Computation*. PhD thesis, Department of EECS, MIT.

[Wesselowski and Cassandras, 2006] Wesselowski, K. and Cassandras, C. G. (2006). The elevator dispatching problem: Hybrid system modeling and receding horizon control. In *Proc. of 2nd IFAC Conference on Analysis and Design of Hybrid Systems*, pages 136–141.

[Wohletz et al., 2001] Wohletz, J. M., Castañón, D. A., and Curry, M. L. (2001). Closed-Loop Control for Joint Air Operations. In *Proc. of 2001 American Control Conference*, pages 4699–4704.

[Yao et al., 2010] Yao, C., Ding, X. C., and Cassandras, C. G. (2010). Cooperative receding horizon control for multi-agent rendezvous problems in uncertain environments. In *Proc. of 49th IEEE Conf. Decision and Control*, pages 4511–4516.