Spatial Distribution Results for Efficient Multi-Agent Navigation

Jan De Mot Vishwesh Kulkarni Sommer Gentry Eric Feron [1]

Abstract

Coordinated navigation to a single perfectly known target by multiple cooperating sensor-equipped agents in a partially known static environment is investigated. Original results on the spatial distribution of the agents under an optimal policy are derived using dynamic programming and neurodynamic programming techniques. Practical implications are discussed. ■

Keywords: cooperative path planning, multi-agent navigation, stochastic shortest path, dynamic programming, neurodynamic programming ■

1 Introduction

Clusters of multiple autonomous agents are posing exciting navigation and coordination questions [7, 13]. Applications of multi-agent clusters include coordinated navigation to a target [10], coordinated terrain exploration, search and rescue operations, and safe highways [1, 13]. In these applications, each agent gathers local information on the environment which is, at best, known only partially *a priori* and it is essential that the agents *cooperate*, i.e. take into account each other's local information in order to execute the mission as efficiently as possible. For instance, the agent strategies should be such that a cluster of cooperating Unmanned Aerial Vehicles (UAV's) executes a search and rescue mission more efficiently than a single UAV. The efficiency increase depends on how well the agents respond to each other's local information. As opposed to the specific case of *formation flying operations* [6, 14], this demands that rather than adhering to a rigid formation, the agents adopt a policy which allows freedom in their spatial distribution. In this paper, *our goal* is to characterize the *spatial distribution* of the agents while using an optimal strategy. The spatial distribution has important practical consequences on the design of the communication infrastructure since the communication power and efficiency depend strongly on the distance and obstacles between the communicating agents [16, 17].

In this paper, we focus on a specific application, viz. the *coordinated navigation to a single perfectly known target* by multiple cooperating sensor-equipped agents in a partially known static environment. The premise is that the environment is littered with obstacles and the requirement is that the agents reach the target zone *as quickly as possible* while minimizing the danger and discomfort incurred in the obstacle encounters. Thus, rather than move back and forth, the agents are constrained to *move towards* the target. They use onboard sensors to gather local information, which is shared with all other agents and is used to derive an efficient navigation policy. Intuitively, the agents should stay away from each other so that the redundancy in the local information is minimized; however, there is a conflict as follows. A poorly located dynamically constrained agent might not be able to traverse to the local information area communicated to it by a better located agent. Thus, if the separation is too large, the cooperation utility of this information is low. Furthermore, the higher the separation, the greater the communication cost of this information. Hence, it is vital to characterize the spatial distribution of the agents under an optimal policy. Novelty of this paper is in investigating this unexplored issue.

Single agent navigation to a known target in an unknown environment has been widely studied [5, 15]. Orioli *et al* [15] use a repeated sequence of map building and motion planning, where the agent has limited visual local information. Blei-Kaelbling [4] formulate this problem as a partially observable Markov decision process. The multi-agent extensions are well discussed in [8, 9] which cast them in the framework of pursuit-evasion games wherein the evader is a moving target and the pursuers are the cooperating agents.

The paper is organized as follows. Section 2 introduces the notation and states two optimal navigation problems. Section 3 formulates these problems as a dynamic program and presents an exact dynamic programming solution and a neurodynamic programming solution. Section 4 describes some important features of these

[1] The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139-4307. Emails:jdemot@mit.edu, vishwesh@mit.edu, sommerg@mit.edu, feron@mit.edu. Research supported in parts by Navy-ONR Young Investigator Award Grant N000-99-1-0668, Air Force Research Laboratory Grant F33615-01-C-1850 and Department of Energy Computational Science Graduate Fellowship DE-FG02-97ER25308.

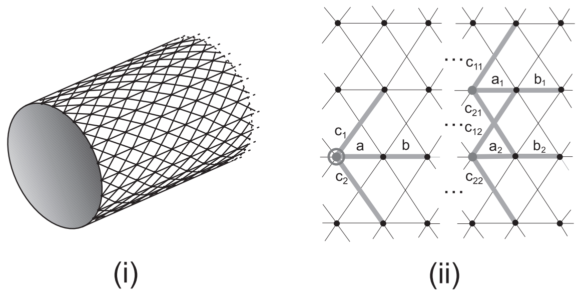

Figure 1: We abstract out the navigation terrain as the cylindrical graph $\mathcal{G}_m$ shown in (i). The observed links terminology is illustrated in (ii) for the cases when the agents are together and when they are separate, on an arbitrary stage.

solutions and presents a result, viz. Lemma 1, on the spatial distribution of the agents. Section 5 discusses these findings and the paper is concluded in Section 6. A proof of Lemma 1 is given in the Appendix.

2 Notation and Problem Formulation

2.1 Notation

We grid the navigation terrain into sectors and represent a sector by a node. This reduces the navigation problem to a graph traversal problem. Associated with each link is an *overhead*; e.g. if the link represents a particularly unfavourable region, its overhead is high. The *cost* of a link l from a node n_1 to a node n_2 is the sum of the traversal cost from the region represented by n_1 to the region represented by n_2, and the overhead of l. The transition graph $\mathcal{G}_m$ of interest has an infinite number of stages S_j having m nodes each. Rather than $\mathcal{G}_m$ being a two dimensional graph, we let it *wrap around* like a cylinder as shown in Figure 1. Then, each stage is a ring of nodes on the cylinder surface. The cylindrical structure of the graph gives us the technical advantage, while losing no generality, of keeping the dynamic programming solution simple by avoiding the boundary effects. Furthermore, as will be made clearer in Section 3, this leads to a spatial invariance which simplifies computation of the value functions in our neurodynamic programs. A line of nodes which runs parallel to the cylinder axis is referred to as an *array*. The i-th node on the stage S_j is denoted n_{ij} and the set of nodes $\{n_{i(\cdot)}\}$ is denoted the i-th array. Given an n-agent cluster on the stage S_j, we choose the first agent arbitrarily, say on n_{ij}, and denote it A_1. Then, the agent located on the next node, say n_{kj}, as the index i increases is referred to as A_2 and the *separation* s_1 between A_1 and A_2 is defined as $s_1 \doteq k - i$. The rest of the nodes and the separations s_i between A_i and A_{i+1} are defined likewise. Let $s \doteq [s_1 \; s_2 \; \ldots \; s_n]^T$, where $(\cdot)^T$ denotes transpose of the vector $(\cdot)$. Premultiply s by a cyclic permutation matrix enough number of times so that the last entry of the resultant vector $\hat{s}$ is its largest. Then, the *separation vector* σ is defined as $\sigma \doteq [\hat{s}_1 \; \hat{s}_2 \; \ldots \; \hat{s}_{n-1}]^T$. With a slight abuse of terminology, we will refer to the elements of σ as s_i. Note that $n-1$ separations suffice to fully describe the position of n agents, since the graph is cylindrical. At a node n_{ij}, the set of links whose costs are known via sensory information is denoted its *local information zone* (see Figure 1): the *straight-ahead links*, viz. the link to $n_{i,j+1}$ (cost denoted a, a_k) and the link from $n_{i,j+1}$ to $n_{i,j+2}$ (cost denoted b, b_k), and the *cross links*, viz. the link to $n_{i-1,j+1}$ (cost denoted c_1, c_{1k}) and the link to $n_{i+1,j+1}$ (cost denoted c_2, c_{2k}), where k is the agent index. For an n-agent cluster, λ denotes the set of link costs in the local information zones, and α denotes the set of the link costs a_k. At each stage, every agent makes a decision: go straight ahead or go ahead using use a cross link. The decision set U of the cluster comprises all combinations of these three options available to the individual agents. Given that two agents are on the same stage, then we say that they are in *configuration* $\mathcal{C}_0$ if they are on the same node, in $\mathcal{C}_1$ if they are on adjacent nodes and in $\mathcal{C}_{\ell+1}$ ($\ell \geq 1$) if the number of nodes in between them equals ℓ. The problems to be solved are as follows.

2.2 Problem Statement

Let the *a priori* information on the link costs be that they are independent uniformly distributed random variables. Let the agents communicate the local information between each other instantaneously and be restricted to always move *towards* a target t located arbitrarily away from the starting position of the agents.

Problem 1 Let a cluster of two agents be stationed at a stage in $\mathcal{G}_m$. Let every agent have the straight-ahead link cost information. Then, find the path for each agent such that the expected aggregate cost in reaching the target is minimized. ∎

Problem 2 Let a cluster of n agents be stationed at a stage in $\mathcal{G}_m$. Let every agent have the straight-ahead as well as the cross link cost information. Then, find the path for each agent such that the expected aggregate cost in reaching the target is minimized. ∎

3 DP and NDP Formulations

We now present a *dynamic programming* (DP) solution to Problem 1 and formulate Problem 2 as a DP. The computational complexity of solving Problem 2 via DP is too high. Hence, we present its approximate solution via *neurodynamic programming* (NDP).

3.1 DP Formulation

Optimal navigation by an n-agent cluster on $\mathcal{G}_m$ can be cast as a discounted cost, finite-state, infinite horizon DP problem as follows. Let $x \doteq (\sigma, \lambda)$ denote the state of the cluster at a certain stage. Given x and a decision $u \in U$, the cluster moves into a new state $f(x,u)$, where $f(\cdot,\cdot)$ is the state transition function. The incurred cost, denoted $g(x,u)$, is the aggregate cost of the links traversed by the agents to advance to the next stage. Policy μ is a mapping $\mu : \mathcal{S} \to U$ where $\mathcal{S}$ is the set of all possible states x. Then the discounted cost to move infinitely towards t, starting from stage S_0 and state x_0, using the policy μ is given as

$$J_\mu(x_0) = \lim_{N\to\infty} E\left[\sum_{i=0}^{N} \varepsilon^i g(x_i, \mu(x_i))\right],$$

where the expectation is taken over all unknown link costs and ε is the discount factor, $0 \le \varepsilon < 1$. The cost function J_μ satisfies

$$J_\mu(x) = E[g(x,\mu(x)) + \varepsilon J_\mu(f(x,\mu(x)))]. \qquad (1)$$

The corresponding Bellman equation is

$$J^*(x) = \min_{u\in U} E\left[g(x,u) + \varepsilon J^*(f(x,u))\right], \qquad (2)$$

where J^* the optimal cost-to-go function [2].

3.2 NDP Formulation

NDP is a simulation based approximate dynamic programming method in which a parametrized function $\tilde{J}(x,r)$ is chosen where the *parameter vector* r is tuned such that $\tilde{J}$ approximates the Bellman equation solution. The general form of the approximation function $\tilde{J}(x,r)$ is $\tilde{J}(x,r) \doteq r^T\phi(x)$ where ϕ is a vector of the *base functions*. The base functions need only depend on the controllable part of the state x [3, Example 2.2, pp. 48]. Since the controllable part of x comprises σ and α, we specify $\tilde{J}(x,r)$ as

$$\tilde{J}(x,r) \doteq r_c + \tilde{J}_\alpha(\alpha, r_\alpha) + \tilde{J}_\sigma(\sigma, r_\sigma)$$

where r_c is a constant and the functions $\tilde{J}_\alpha(\alpha, r_\alpha)$ and $\tilde{J}_\sigma(\sigma, r_\sigma)$ are decided as follows. Every link cost $a_j \in \alpha$ has an associated parameter $r_{a,j}$. Spatial invariance due to the cylindrical shape of the graph reduces the set of parameters $r_{\alpha,j}$ to a single parameter r_α. Hence,

$$\tilde{J}_\alpha(\alpha, r_\alpha) = r_a \sum_{j=1}^{n} a_j.$$

For $\tilde{J}_\sigma(\sigma, r_s)$, we define $n-1$ sets of parameters $\mathcal{P}_j$ containing one parameter for each possible value of s_j. In particular, $\mathcal{P}_j \doteq \{r_{s,j}(d) \mid d \in \mathcal{D}_j\}$ where $\mathcal{D}_j$ is the set of possible values of s_j and $r_{s,j}(d)$ is the parameter corresponding to s_j when $s_j = d$. Then,

$$\tilde{J}_\sigma(\sigma, r_\sigma) = \sum_{j=1}^{n-1} r_{s,j}(s_j).$$

Again, the spatial invariance allows us to relax the dependence of $r_{s,j}(s_j)$ on the agent index j so that

$$\tilde{J}(x,r) = r_c + r_a \sum_{j=1}^{n} a_j + \sum_{j=1}^{n-1} r_s(s_j). \qquad (3)$$

For tuning the parameter vector, we use a variation of Sutton's simulation based TD(0) (*Temporal Differences*) tuning method [18]. Under a fixed policy μ, a sequence of states x_k is simulated as $x_{k+1} = f(x_k, \mu(x_k))$ with an arbitrary initial state x_0. For each k, the current parameter vector r_k, is updated according to

$$r_{k+1} = r_k + \gamma_k d_k \nabla_r \tilde{J}(x_k, r_k), \qquad (4)$$

where γ_k is the step size, and d_k is the temporal difference given by

$$d_k \doteq g(x_k, \mu(x_k)) + \varepsilon \tilde{J}(x_{k+1}, r_k) - \tilde{J}(x_k, r_k). \qquad (5)$$

Under some decreasing step size conditions, r_k converges to a vector r_μ such that $\tilde{J}(x, r_\mu)$ is an approximate solution of (1). Then, the algorithm works as follows: we start with a policy μ, execute TD(0) till r converges and then compute a new policy using

$$\mu_r(x) = \arg\min_{u\in U} E\left[g(x,u) + \varepsilon \tilde{J}(f(x,u), r)\right], \qquad (6)$$

and repeat. It is common to change this policy μ at each update r_k of the parameter vector. This version of the algorithm is called *optimistic* TD(0). Furthermore, we encourage exploration in the simulated state sequence. Specifically, with probability $p = 0.5$, we choose the next state x_{k+1} using a random decision u, i.e. not the current optimal decision $u_k = \mu_{r_k}(x_k)$ given by (6). However, the temporal difference d_k given by (5) and, subsequently, the parameter update r_k given by (4) are still computed as if the state x_{k+1} were chosen using the current optimal decision u_k. This important technicality is adopted routinely to reduce suboptimality inherent in the NDP solutions [3, 12].

4 Main Results

We implemented a dynamic programming solution to Problem 1 in MATLAB. The unknown link costs were assumed to be independent and uniformly distributed over $\{0, 0.5, \ldots, 3\}$. Navigation efficiency increases with the number of agents employed, as shown in Figure 3. These efficiency results are for the infinite horizon graph $\mathcal{G}_7$ with the discount factor ε taken to be 0.8. Four different initial position configurations are possible, viz. $\mathcal{C}_0$ through $\mathcal{C}_3$. The average discounted cost per agent was computed by choosing the initial position configurations using a uniform probability distribution. A simulation result for a link cost scenario is shown in

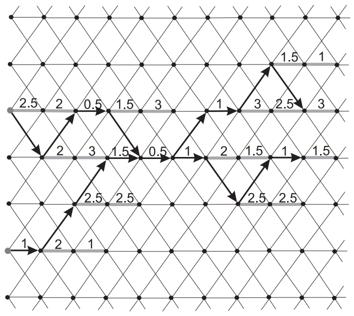

Figure 2: DP solution to Problem 1. The bold arrows show the path of progression and the medium dark lines show the observed link costs at the traversed nodes.

Figure 2. It indicates that, for the topology considered, a cluster of two agents does not proceed towards the target with more than a single node separation, the agents remain within two nodes from each other. Indeed, we have the following result.

Lemma 1 [Cluster Separation Lemma]
Given a two agent cluster in $\mathcal{G}_m$ ($m \leq 7$) using optimal paths derived as the solution to Problem 1, configurations $\mathcal{C}_0, \mathcal{C}_1$ and $\mathcal{C}_2$ do not evolve into configurations $\mathcal{C}_\ell$ having $\ell > 2$. Furthermore, the configuration $\mathcal{C}_3$ always evolves into either $\mathcal{C}_2$ or $\mathcal{C}_1$ unless it is optimal for the agents to choose links a_1 and a_2. □

Proof: See Appendix.

Thus, once the agents are in $\mathcal{C}_2$, they do not separate any further. Intuitively, adding another node in each stage should not cause the separation to grow. Therefore, we believe that Lemma 1 is strong enough to hold for any $\mathcal{G}_m$ and conjecture the following result.

Conjecture 1 [Cluster Separation Conjecture]
Given a two agent cluster in $\mathcal{G}_m$ using optimal paths derived as the solution to Problem 1, configurations $\mathcal{C}_0, \mathcal{C}_1$ and $\mathcal{C}_2$ do not evolve into configurations $\mathcal{C}_\ell$ having $\ell > 2$. Furthermore, a configuration $\mathcal{C}_\ell$, where $\ell > 2$, does not evolve into a configuration $\mathcal{C}_\nu$ such that $\nu > \ell$.

We implemented the NDP algorithm for Problem 2 in which a three agent cluster navigates towards a target on $\mathcal{G}_{12}$. The unknown link costs were taken to be uniformly distributed over the set $\{0, 0.05, \ldots, 3\}$ and $\varepsilon = 0.8$ was chosen to be the discount factor. The approximate value function was taken to be

$$\widetilde{J}(s_1, s_2, a_1, a_2, a_3) = r_a \sum_{i=1}^{3} a_i + \sum_{i=1}^{2} r_s(s_i),$$

Number of Agents	Policy	Cost per Agent
1	Random Motion	7.50
1	Optimal Policy	5.19
2	Optimal Policy	5.03

Figure 3: Clusters and their navigation costs.

Separation	0	1	2	3	4	5	6
Occurrence (%)	62.9	8.2	7.9	7.4	6.7	5.1	1.8

Figure 4: Frequency of separations under μ_g.

where the constant term r_c is absorbed in the $r_s(s_i)$ terms. The NDP algorithm converged to $r_a = 0.54$ and $r_s(s) = [4.28\ 4.22\ 4.25\ 4.25\ 4.26\ 4.26\ 4.27]$. The second element of $r_s(s)$ is the least, which indicates that the separations $s = 1$ are weighted the least. Under the policy μ_g, which is greedy with respect to $\widetilde{J}$, the expected discounted cost per agent is $J^n_{\mu_g} = 3.63 \pm 0.03$ cost units, with 95% probability. The average cost per stage per agent is $h^n_{\mu_g} = 0.73 \pm 0.01$ cost units. Under the policy, there are 22% stages on which at least one of the agent does not choose the least cost link. Unlike the Problem 1 situation, a deterministic bound on the separations does not exist: they can become large, but only with a low probability (see Figure 4). These simulations were carried out over 100,000 stages.

The corresponding results for an optimal policy μ_o for three *non-cooperating* agents are as follows. The policy μ_o is computed simply by solving a single agent dynamic program exactly in which each agent uses its own local link costs only. The expected discounted cost for the single agent navigation problem is $J_{\mu_o} = 3.64$ cost units and the average cost per stage is $h_{\mu_o} = 0.73$ cost units. Under the policy μ_o, there are 17.7% stages on which at least one agent does not choose the least cost link. Under the policy μ_g, this figure is slightly higher at 22%, which suggests that the *exploration bonus* (see [4]) increases with cooperation.

5 Discussion

In practice, the agents have only a limited local information zone and a limited communication range. Furthermore, they are dynamically constrained. The local information communicated by an agent is of any navigation use to another agent only if it can reach that zone without jeopardizing the mission efficiency. Every mission specifies the area, say the *reachability zone*, around its current location which an agent can cover over a unit time interval. For example, it a usually a disc for the exploration operations described in [10] and a cone, directed towards the target, for the target

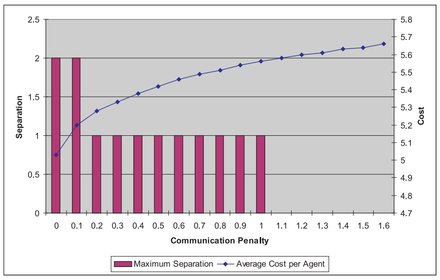

Figure 5: The average cost and the maximum separation of a two-agent cluster for the Problem 1 setting when a communication cost c is accounted for.

acquisition applications described in [8, 15]. Whether the observation zones and the reachability zones have a joint property that is of interest to the mission efficiency has remained an unexplored question. Such a property has an immense use in choosing the agents and the communication infrastructure appropriately.

In this paper, we have considered a special case of these zones: the observation zones are as specified in Section 2.1 and the reachability zone on the node n_{ij} is the set $\{n_{i,j+1}, n_{i-1,j+1}, n_{i+1,j+1}\}$. We have looked at the spatial distribution of the agents as the joint property of these zones and have discussed its impact on the mission efficiency, viz. the navigation cost, in Section 4. Lemma 1 states that when the separation is greater than $s = 2$, then, under an optimal policy, the agents in a two-agent cluster navigate such that the separation decreases. It quantifies the intuition that only when the agents stay close together can an agent exploit local information gathered by the other, since only then can it reach the area observed by the other. Our NDP based results for a different combination of the zones (see Section 3.2 and Section 4) show that the separation stays less than $s = 5$ with 93% probability.

In practice, it may be necessary to design optimal control laws when the communication infrastructure and dynamical constraints are specified *a priori*. Those constraints may be incorporated by simply modifying (2) to penalize the separation additively. Figure 5 shows the maximum separation and the average cost which a two-agent cluster incurs under such an optimal policy as a function of the separation penalty c where the rest of the conditions are as specified in Problem 1.

6 Conclusion and Future Directions

The problem of coordinated navigation to a single perfectly known target by multiple cooperating sensor-equipped agents in a partially known static environment is investigated. We have introduced the con-

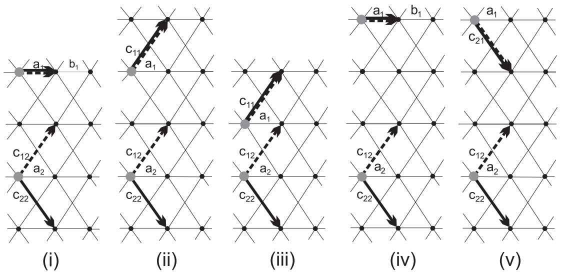

Figure 6: The cases (i), (ii), and (iii) are the instances where a policy μ_2 (bold arrows) exists which leads to $\mathcal{C}_3$ from $\mathcal{C}_l$, $l < 3$. The cases (iv) and (v) are the instances where a policy μ_2 (bold arrows) exists such that $\mathcal{C}_3$ is unchanged even if an agent chooses a cross link. The dashed arrows show the *preferrable* policy μ_1.

cepts of *information zone* and *reachability zone* and have identified their role in designing efficient navigation policies. For some special cases of these zones, the optimal and sub-optimal navigation policies are derived using, respectively, DP and NDP techniques (see Section 3 and Section 4). For a specific case, the spatial distribution of the agents is firmly characterized by Lemma 1. Derivation of the optimal strategies and the spatial distribution results for more general cases of these zones is a promising research direction. Firm spatial distribution results for these cases and for larger clusters necessitate the use of NDP techniques, which in turn necessiates a characterization of how close the value function is to J^*. This characterization is a promising research direction [11].

7 Appendix: Proof of Lemma 1

We present the proof for the case $m = 7$. The arguments can be used for the other cases as well. There are only three situations (see Figure 6, (i), (ii) and (iii)) in which there exists a policy that leads to the configuration $\mathcal{C}_3$ from $\mathcal{C}_\ell$ where $\ell < 3$. In each of the three cases, we will show that policy μ_1 is preferable to the policy μ_2 (see Figure 6 for the illustration).

Let $J^*_{S,j}(a_1, a_2, b_1, b_2)$ denote the expected aggregate discounted cost incurred in moving infinitely in the direction of t, given that the cluster members are separated by j nodes with $0 \le j \le \lfloor \frac{m}{2} \rfloor$ where the lookahead link costs are a_1, a_2, b_1, and b_2 as shown in Figure 1. Let $E_{\alpha,\beta}\left[f(\alpha,\beta,\gamma)|\gamma)\right]$ denote the conditional expected value of function $f(\cdot,\cdot,\cdot)$ given γ. In the Fig. 6 (i) situation, μ_1 is preferable to μ_2 if

$$a_1 + E_{c_{22},\phi_2,\psi_1,\psi_2}[c_{22} + \varepsilon J^*_{S,3}(b_1,\phi_2,\psi_1,\psi_2)|b_1] >$$
$$a_1 + E_{c_{12},\phi_2,\psi_1,\psi_2}[c_{12} + \varepsilon J^*_{S,1}(b_1,\phi_2,\psi_1,\psi_2)|b_1],$$

i.e., the expected values of c_{12} and c_{22} being equal, if

$$E_{\phi_2,\psi_1,\psi_2}[J^*_{S,3}(b_1,\phi_2,\psi_1,\psi_2)|b_1] > E_{\phi_2,\psi_1,\psi_2}[J^*_{S,1}(b_1,\phi_2,\psi_1,\psi_2)|b_1]. \quad (A.1)$$

Let J_k denote the approximate value function obtained after k value iteration steps. Although J^* is not precisely known, its value iteration upper bounds and lower bounds, say $\overline{J}_k$ and $\underline{J}_k$, are known [3] and those can be used to check validity of the inequality (A.1). Omitting the subscripts k for notational simplicity, it follows that if

$$E_{\phi_2,\psi_1,\psi_2}[\underline{J}_{S,3}(b_1,\phi_2,\psi_1,\psi_2)|b_1] > E_{\phi_2,\psi_1,\psi_2}[\overline{J}_{S,1}(b_1,\phi_2,\psi_1,\psi_2)|b_1] \quad (A.2)$$

holds for some k, then the inequality (A.1) holds as well. Using numerical data, (A.2) can be verified to hold for all b_1. Similarly, in the other two configurations, the following conditions need hold for some k:

$$E_{\phi_1,\phi_2,\psi_1,\psi_2}[\underline{J}_{S,3}(\phi_1,\phi_2,\psi_1,\psi_2)] > E_{\phi_1,\phi_2,\psi_1,\psi_2}[\overline{J}_{S,2}(\phi_1,\phi_2,\psi_1,\psi_2)], \quad (A.3)$$

$$E_{\phi_1,\phi_2,\psi_1,\psi_2}[\underline{J}_{S,3}(\phi_1,\phi_2,\psi_1,\psi_2)] > E_{\phi_1,\phi_2,\psi_1,\psi_2}[\overline{J}_{S,1}(\phi_1,\phi_2,\psi_1,\psi_2)]. \quad (A.4)$$

The above two inequalities can be verified to hold using numerical data.

It now needs to be established that the configuration $\mathcal{C}_3$ always evolves into either $\mathcal{C}_2$ or $\mathcal{C}_1$ unless it is optimal for the agents to choose links a_1 and a_2. There are two situations (see Fig. 6, (iv) and (v)) in which there exists a policy that keeps $\mathcal{C}_3$ in $\mathcal{C}_3$ should the agents choose not to take the straight links a_i (see Fig. 6 for the illustration and terminology). Using the arguments employed earlier, it follows that μ_1 is preferable to μ_2 if (A.4) and

$$E_{\phi_1,\psi_1,\psi_2}[\underline{J}_{S,3}(\phi_1,b_1,\psi_1,\psi_2)|b_1] > E_{\phi_2,\psi_1,\psi_2}[\overline{J}_{S,2}(b_1,\phi_2,\psi_1,\psi_2)|b_1] \quad (A.5)$$

hold for some k. Using numerical data, the inequalities (A.4) and (A.5) can be verified to hold. QED.

References

[1] S. Bana and P. Varaiya. Space division multiple access (SDMA) for robust ad-hoc vehicle communication networks. In *4th IEEE ITSC Conference*, August 2001.

[2] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, 1995.

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

[4] D. Blei and L. Kaelbling. Shortest paths in a dynamic uncertain domain. In *IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments*, 1999.

[5] R. Cheung. Iterative methods for dynamic stochastic shortest path problems. *Naval Research Logistics*, 45:769–789, 1998.

[6] A. Fax and R. Murray. Information flow and cooperative control of vehicle formations. In *IFAC World Congress*, 2002 (to appear).

[7] A. Girard, J. de Souza, and J. Hedrick. An overview of emerging results in networked multi-vehicle systems. In *IEEE CDC*, 2001.

[8] J. Hespanha, H. Kim, and S. Sastry. Multiple-agent probabilistic pursuit-evasion games. In *IEEE CDC*, December 1999.

[9] J. Hespanha and M. Prandini. Optimal pursuit under partial information. In *Proc. of the 10th Mediterranean Conf. on Control and Automation*, July 2002.

[10] M. Jun, A. Chaudhry, and R. D'Andrea. The navigation of autonomous vehicles in uncertain dynamic environments: A case study. In *IEEE CDC*, 2002 (to appear).

[11] Bo Lincoln and Anders Rantzer. Suboptimal dynamic programming with error bounds. Submitted to IEEE CDC, 2002.

[12] P. Marbach, O. Mihatsch, and J. Tsitsiklis. Call Admission Control and Routing in Integrated Service Networks Using Neuro-Dynamic Programming. *IEEE Journal on Selected Areas in Communications*, 18(2):197–208, 2000.

[13] R. Morris, J. Jannotti, F. Kaashoek, and J. Li. CarNet: A scalable ad hoc wireless network system. In *Proceedings of the 9th ACM SIGOPS European workshop: Beyond the PC: New Challenges for the Operating System*, September 2000.

[14] P. Ogren, E. Fiorelli, and N. Leonard. Formations with a mission: Stable coordination of vehicle group maneuvers. In *Proc. Symposium on Mathematical Theory of Networks and Systems*, August 2002.

[15] G. Oriolo, G. Ulivi, and M. Vendittelli. Real-time map building and navigation for autonomous robots in unknown environments. In *IEEE Transactions on Systems, Man and Cybernetics, Part B*, volume 28-3, pages 316–333, June 1998.

[16] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Sadle River, NJ, 1996.

[17] R. Steele, C. Lee, and P. Gould. *GSM, cdmaOne and 3G Systems*. John Wiley and Sons, New York, NY, 2001.

[18] R. Sutton. Learning to predict by the methods of temporal differences. In *Machine Learning*, volume 3, pages 8–44, 1988.