Spatial Distribution of Two-Agent Clusters for Efficient Navigation

Jan De Mot Eric Feron

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{jdemot,feron}@mit.edu

***Abstract*— Coordinated navigation by two cooperating sensor-equipped agents in a partially known static environment is investigated. Each agent observes a local part of the otherwise unknown environment and shares the gathered data with the other agents. In general, dynamic programming techniques suitably model the navigation problem, but are computationally hard to solve. We propose a combined dynamic and linear programming framework to circumvent the curse of dimensionality and establish in the process a firm upper bound on the spatial separation of a two-agent cluster navigating on a structured arbitrarily large graph.**

I. Introduction

Groups of multiple cooperating autonomous agents have attractive advantages over single-agent navigation. In particular, during mission execution agent clusters are more robust than a single agent since failure of one agent does not jeopardize the success of the mission in the former case. Furthermore, agent groups can navigate in a more energy efficient way using formation flight [8] and individual threat exposure can be reduced using a cooperative navigation scheme, as opposed to flying in a rigid formation. Applications of multi-agent cluster navigation include coordinated navigation to a target, coordinated terrain exploration and search and rescue operations. Commonly, the general environment structure is known, but local temporal obstacles or deviations from the environment structure may appear in random fashion. Therefore, each agent gathers local information on the environment which is only known partially *a priori*, and shares this information with the other agents. The agents *cooperate*, using each others information to execute the mission as efficiently as possible.

In this paper, we focus on coordinated navigation towards a target in a partially known environment (see Figure 1) structured as an urban area, with a cluster of for example Unmanned Aerial Vehicles, hereafter referred to as agents. Street and avenue locations are known *a priori*, as opposed to threat or reward locations, of which an *a priori* probability distribution is given. The requirement is that the agents reach the target while minimizing the total agent threat exposure or maximizing any type of reward associated with traversing a particular zone. The agents use onboard sensors to gather local information which is shared with the other agents and is used to derive an optimal navigation policy. Since buildings limit the agent observation zone, the local information consists of the threat level or the reward in the streets straight ahead of each agent (see Figure 1). This type of information zone is denoted *tunnel vision*.

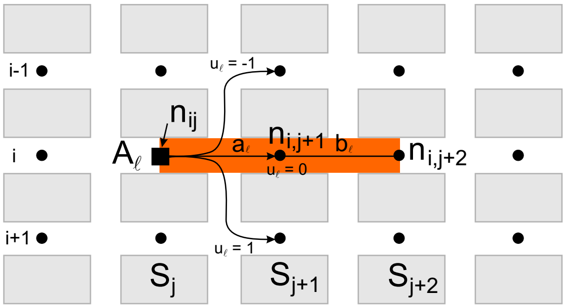


Fig. 1. The mapping of an urban environment into a graph. Grey areas denote buildings, the dots represent the nodes; at a node, each agent A_l can takes one of three decisions (diagonally up or down, or straight). The orange area is the local information zone of agent A_l at node n_{ij}. Vertical node array S_j denotes the jth stage. Not all links are shown.

In [6], we consider a cylindrically shaped graph. The spatial invariance in circumferential direction reduces the computational complexity significantly. There, we prove an upper bound on the separation of a two-agent cluster under an optimal policy using *dynamic programming* (DP). In this paper, we extend this work for an arbitrarily large plane graph with a similar structure. The state space for the latter environment is arbitrarily large, making it computationally impossible to solve the associated dynamic program. Using a combined dynamic and *linear programming* (LP) formulation, we show an upper bound on the agent separation, reducing the state space drastically while maintaining optimality. This work forms the basis of a framework that promises to solve larger multi-agent navigation problems optimally.

Many authors consider cooperative multi-agent navigation problems in different settings. Some focus on experimental aspects, integrating measurements from different cooperating agents, using heuristic guidance algorithms [5], [7], [9]. On the theoretical side [3] formulates the navigation problem as a partially observable markov decision problem and solves instances approximately. Furthermore, [4], [10] use a game-theoretic approach to the problem formulating it in a pursuit-evasion setting. Our work distinguishes itself by focusing on finding properties of optimal policies to optimally solve otherwise intractable multi-agent navigation problems.

The paper is organized as follows. Section II introduces the notation and states the navigation problem at hand. Section III formulates this problem as a DP and introduces an alternative and tractable formulation based on a combined DP-LP framework. Section IV proves the validity of the latter approach and presents the main result, *i.e.* Lemma 1, an upper bound on the spatial distribution of the agents. Section V discusses these findings and the paper is concluded in Section VI.

II. Notation and Problem Formulation

A. 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 node is an *overhead*; e.g. if the node represents a particularly inaccessible region, its overhead is high. The *cost* of the link 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 n_2. The structured transition graph $\mathcal{G}$ considered (see Fig. 1) has an infinite number of *stages* S_j having an infinite number of nodes each. Each stage is a vertical node line, the horizontal node lines are referred to as *lanes*. The ith node on stage S_j is denoted n_{ij} and the set of nodes $\{n_{i(\cdot)}\}$ is denoted the ith lane (i integer; choose an arbitrary lane with index 0). Given a two-agent cluster on the stage S_j, we choose the agent on the node with the lowest index, say on n_{ij}, as the first agent and denote it A_1. Then, the other agent located on node n_{kj}, where $k \geq i$, is referred to as A_2 and the *separation* s between A_1 and A_2 is defined as $s \doteq k - i$.

At a node n_{ij}, the set of links whose costs are known via sensory information an agent located on n_{ij} gathers, is denoted its *local information zone*. For the remainder of this paper, we use the local information zone that consists of the *straight ahead links* (see Fig. 1), *i.e.* the link to $n_{i,j+1}$ (cost a, a_l) and the link from $n_{i,j+1}$ to $n_{i,j+2}$ (cost b, b_l), where l is the agent index ($l \in \{1,2\}$). Note that the *cross links*, *i.e.* the link to $n_{i-1,j+1}$ and the link to $n_{i+1,j+1}$, are not contained in the local information zone.

For a two-agent cluster, observation $\lambda = \{a_1, b_1, a_2, b_2\}$ denotes the set of link costs in the local information zones; Λ denotes the set of all possible observations. At each stage, every agent makes a decision: go straight ahead or go ahead using one of the two cross links. The decision set U of the cluster comprises all combinations of these three options available to the individual agents.

B. Problem Statement

Let a cluster of two agents be stationed at stage S_0 of graph $\mathcal{G}$. Let the link costs be independent identically distributed random variables. Let the agents have the link cost information of the local information zones and communicate the local information between each other instantaneously. Let the cluster be restricted to always move *towards* a target t located arbitrarily away from the starting position of the agents. Then, find the navigation strategy for each agent so that the expected discounted aggregate cost in reaching the target is minimized.

III. DP and LP Formulations

We now present a DP formulation for the problem. Since the state space has infinite size, solving the DP numerically is impossible. Hence, we present an alternative approach using a combined DP-LP framework in Section III-B, shown to hold under technical conditions in Section IV.

A. DP Formulation

Optimal navigation by a two-agent cluster on $\mathcal{G}$ can be cast as a discounted cost, infinite-state, infinite horizon DP problem as follows (see [1] for an in-depth treatment of DP formulations). Let $\chi \doteq (s, \lambda)$ denote the state of the cluster at a certain stage. Let decision $\mathbf{u} = (u_1, u_2)$, $\mathbf{u} \in U$; let $u_l = -1, 0, 1$ denote the cases where agent A_l chooses the cross link diagonally upwards, the link straight ahead and the cross link diagonally downwards, respectively. Given χ and $\mathbf{u}$, the cluster moves into a new state $f(\chi, \mathbf{u})$, where $f(\cdot,\cdot)$ is the state transition function. The incurred cost, denoted $g(\chi, \mathbf{u})$, 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 χ. Then the discounted cost to move infinitely towards t, starting from stage S_0 and state χ_0, using policy μ is given as

$$J_\mu(\chi_0) = \lim_{N\to\infty} E\left[\sum_{i=0}^{N} \varepsilon^i g(\chi_i, \mu(\chi_i))\right],$$

where, for the remainder of this paper, the expectation $E[\cdot]$ is taken over all unknown link costs and ε is the discount factor, $0 \leq \varepsilon < 1$. The cost function J_μ satisfies

$$J_\mu(\chi) = E\left[g(\chi, \mu(\chi)) + \varepsilon J_\mu(f(\chi, \mu(\chi)))\right].$$

The corresponding Bellman equation is

$$J^*(\chi) = \min_{\mathbf{u}\in U} E\left[g(\chi, \mathbf{u}) + \varepsilon J^*(f(\chi, \mathbf{u}))\right], \tag{1}$$

where J^* is the optimal value function.

B. Combined LP-DP Formulation

First we describe a combined LP-DP approach for the general two-agent navigation problem on $\mathcal{G}$ (see [2] for an in-depth treatment of LP formulations). In particular, we compute a value function using a simplified DP for separations $s \leq \sigma$ and using a sequence of LPs for all $s > \sigma$, where σ needs to be determined. Then, we describe the LP sequence for $s > \sigma$ for a specific parameter set and finally we give a standard form representation.

1) General Problem: Let the decision set $U^{\overline{s}}(s)$ comprise all combinations of the three options available to each agent that lead to $s' \leq \overline{s}$, where $(\cdot)'$ denotes a shorthand notation for the next stage value of $(\cdot)$. Let $J^{\overline{s}}$ denotes the value function for the problem where the agent separation is artificially bounded by $\overline{s}$. Then, $J^{\overline{s}}(\chi)$ is the solution of the following Bellman equation:

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

where $J^{\overline{s}}(\chi)$ is defined for $s \leq \overline{s}$ and for $\lambda \in \Lambda$. The corresponding optimal policy $\mu^{\overline{s}}(\chi)$ is then computed as:

$$\mu^{\overline{s}}(\chi) = \arg\min_{\mathbf{u} \in U^{\overline{s}}(s)} E\left[g(\chi, \mathbf{u}) + \varepsilon J^{\overline{s}}(f(\chi, \mathbf{u}))\right].$$

Let us assume there exists a separation $\bar{\sigma}$ for which the following property holds:

$$J^{\bar{\sigma}}(\chi) = J^{\bar{\sigma}+2}(\chi) \text{ for all } \chi = (s, \lambda), \qquad (3)$$
$$\text{where } s \leq \bar{\sigma}.$$

In particular, under $\mu^{\bar{\sigma}+2}(\chi)$ and for $s \leq \bar{\sigma}$, the separation at the next stage $s' \leq \sigma$. This indicates but doesn't show that σ is an upper bound of the agent separation under an optimal policy. Let σ denote the smallest $\bar{\sigma}$ satisfying (3).

Next, we obtain an expression using LP for the value function for $s \geq \bar{\sigma} + 1$. In [1], an LP approach for solving a DP is described: J^* is the 'largest' J that satisfies the following set of linear constraints:

$$J(\chi) \leq E[g(\chi, \mathbf{u}) + \varepsilon J(f(\chi, \mathbf{u}))],$$
$$\text{for all } \chi \in \mathcal{S}, \text{ and } \mathbf{u} \in U.$$

In particular, J^* solves the LP of maximizing $\sum_{\chi \in \mathcal{S}} J(\chi)$ subject to the above set of constraints. Let x_λ^s denote the LP variables $J(\chi)$, and let $\mathbf{x}^s$ denote the vector of x_λ^s, for all $\lambda \in \Lambda$. Let the decision set $\overline{U}$ comprise all decision combinations whereby the agent separation strictly decreases and the decision combination whereby the agents take the straight ahead links, with cost a_1 and a_2. Then, we consider the following (smaller) set of LP constraints, for $s \geq \sigma + 1$:

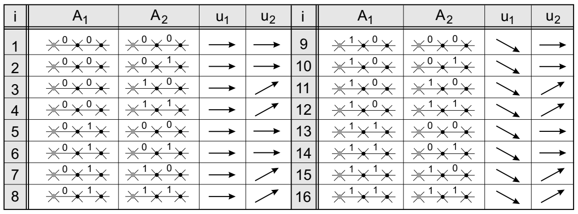
$$x_\lambda^s \leq E[g((s, \lambda), \mathbf{u}) + \varepsilon x_{\lambda'}^{s'}],$$
$$\text{for all } \chi \in \mathcal{S}, \text{ and } \mathbf{u} \in \overline{U}.$$

The resulting value function is valid under the premise that the separation strictly decreases or both agents take the straight ahead link, for $s \geq \sigma$. Note that in all constraints $s' \leq s$, which allows to decompose the LP into a sequence of finite dimensional LPs. In particular, given $\mathbf{x}^s$, for $s \leq \alpha$, then $\mathbf{x}^{\alpha+1}$ is the solution of:

$$\begin{aligned} &\text{maximize} && \sum_\lambda x_\lambda^{\alpha+1} && (4) \\ &\text{subject to} && x_\lambda^{\alpha+1} \leq E[g((\alpha+1, \lambda), \mathbf{u}) + \varepsilon x_{\lambda'}^{s'}], \\ &&& \text{for all } \mathbf{u} \in \overline{U} \text{ and } \lambda \in \Lambda. \end{aligned}$$

Given $\mathbf{x}^s$ for $s \leq \sigma$ as solution of (2) with $\bar{s} = \sigma$, the initializing LP for $\alpha = \sigma$ can be solved. Note that $\mathbf{x}^{\alpha-1}$ and $\mathbf{x}^\alpha$ only enter in the right hand side of the inequality constraints of the LP solving for $\mathbf{x}^{\alpha+1}$, assuring that the LP sequence leads to the same solution as the LP solving for $\mathbf{x}^s$ for all s at once.

2) LP-formulation for particular parameter values: For the remainder of this document, we assume that $\varepsilon = 0.9$ and that the link costs are independently distributed over the set of possible link costs $\mathcal{L} \doteq \{0, 1\}$. Let p be the probability of a 0 link cost. Since $|\mathcal{L}| = 2$, there are 16 possible λ's, λ_i, $i = 1, \ldots, 16$. Let $\mathbf{u}_i(s) \in \overline{U}$ be the LP-optimal decision for the two-agent cluster at separation s and with $\lambda = \lambda_i$. Numerical solutions of the LP-sequence (4) for $p = 0.5$ indicate that $\sigma = 3$ and that for all $s \geq 4$, and for all i, decision $\mathbf{u}_i(s)$ is independent of s. In Fig. 2 $\mathbf{u}_i$ is depicted for all i, indirectly describing the optimal corner point, or optimal *basic feasible solution* (bfs) of each LP (4). Since p can vary continuously between 0 and 1, there exist $\overline{p}$ and $\underline{p}$, where $0 \leq \underline{p} \leq 0.5 \leq \overline{p} \leq 1$, so that for $p \in [\underline{p}, \overline{p}]$ the bfs described in Fig. 2 is optimal. Consequently, for each $s \geq 4$ and for $p \in [\underline{p}, \overline{p}]$, the following set of 16 equations in 16 unknowns holds:



i	A_1	A_2	u_1	u_2	i	A_1	A_2	u_1	u_2
1	0 0	0 0	→	→	9	1 0	0 0	↘	→
2	0 0	0 1	→	→	10	1 0	0 1	↘	→
3	0 0	1 0	→	↗	11	1 0	1 0	↘	↗
4	0 0	1 1	→	↗	12	1 0	1 1	↘	↗
5	0 1	0 0	→	→	13	1 1	0 0	↘	→
6	0 1	0 1	→	→	14	1 1	0 1	↘	→
7	0 1	1 0	→	↗	15	1 1	1 0	↘	↗
8	0 1	1 1	→	↗	16	1 1	1 1	↘	↗

Fig. 2. Indirect description of the optimal bfs for each LP, suggested by numerical simulation. The first column of each table half denotes index i, the second and third column depicts the information zone of A_1 and A_2 respectively, whereby the gray dot denotes the agent position, and the fourth and the fifth column denote the optimal agent decision suggested by the LP.

$$x_{\lambda_i}^s = E[g((s, \lambda_i), \mathbf{u}_i) + \varepsilon x_{\lambda'}^{s'}], \text{ for all } i. \qquad (5)$$

From Fig. 2, it can be seen that for each $s \geq 4$, the set of unknowns $\{x_{\lambda_i}^s\}$ can be reduced to a smaller set of independent unknowns, *i.e.* the set $\{x_\lambda^s\}$, for $\lambda \in \Lambda_r = \{\lambda_1, \lambda_2, \lambda_3, \lambda_6, \lambda_7, \lambda_{11}\}$ since:

$$\begin{aligned} x_{\lambda_2}^s &= x_{\lambda_5}^s, \\ x_{\lambda_3}^s &= x_{\lambda_4}^s = x_{\lambda_9}^s = x_{\lambda_{13}}^s, \\ x_{\lambda_7}^s &= x_{\lambda_8}^s = x_{\lambda_{10}}^s = x_{\lambda_{14}}^s, \\ x_{\lambda_{11}}^s &= x_{\lambda_{12}}^s = x_{\lambda_{15}}^s = x_{\lambda_{16}}^s. \end{aligned}$$

For example, $x_{\lambda_2}^s = x_{\lambda_5}^s$ for symmetry reasons, while $x_{\lambda_3}^s = x_{\lambda_4}^s$ since the locally incurred cost g and the state at the next stage χ' are equal. Let $\bar{l} = p$ be the expected value of any unknown link cost. Let $\mathbf{x}_r^s$ denote the vector containing x_λ^s

for $\lambda \in \Lambda_r$, ordered with increasing index i. Then, (5) can be written in the following matrix expression for $s \geq 4$:

$$\mathbf{x}_r^s = \mathbf{C}_1 \mathbf{x}_r^{s-1} + \mathbf{C}_2 \mathbf{x}_r^{s-2} + \mathbf{D}, \qquad (6)$$

where

$$\mathbf{C}_1 = \begin{pmatrix} p^4c & 2p^3rc & 2p^2rc & p^2r^2c & 2pr^2c & r^2c \\ p^4\varepsilon^2 & 2p^3r\varepsilon^2 & 2p^2r\varepsilon^2 & p^2r^2\varepsilon^2 & 2pr^2\varepsilon^2 & r^2\varepsilon^2 \\ p^3\varepsilon & 2p^2r\varepsilon & pr\varepsilon & pr^2\varepsilon & r^2\varepsilon & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p^2\varepsilon & 0 & pr\varepsilon & p\varepsilon \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{C}_2 = \begin{pmatrix} p^4d & 2p^3rd & 2p^2rd & p^2r^2d & 2pr^2d & r^2d \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ p^4\varepsilon^2 & 2p^3r\varepsilon^2 & 2p^2r\varepsilon^2 & p^2r^2\varepsilon^2 & 2pr^2\varepsilon^2 & r^2\varepsilon^2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ p^4\varepsilon & 2p^3r\varepsilon & 2p^2r\varepsilon & p^2r^2\varepsilon & 2pr^2\varepsilon & r^2\varepsilon \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} \frac{2\varepsilon^2 \bar{l} r}{1-\varepsilon p^2} \\ \varepsilon\bar{l} \\ \bar{l} \\ 2\varepsilon\bar{l} \\ \bar{l} \\ 2\bar{l} \end{pmatrix},$$

where $r = 1-p$, $c = \frac{2\varepsilon^3 pr}{1-\varepsilon p^2}$ and $d = \frac{\varepsilon^3 r^2}{1-\varepsilon p^2}$. Note that $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{D} \geq 0$, for all $0 \leq \varepsilon < 1$ and for all $0 \leq p \leq 1$ and hence for all $\underline{p} \leq p \leq \overline{p}$.

3) Standard form: We present the reduction of each LP (4) to an equivalent LP in standard form, for the suggested parameter values. Let $\hat{x}^s$ denote the expected value of the value function at separation s, given no link cost information:

$$\hat{x}^s = \sum_{\lambda \in \Lambda_r} P(\lambda) x_\lambda^s, \qquad (7)$$

where $P(\lambda)$ denotes the sum of the occurrence probabilities of λ and its symmetric instances in Λ. Similarly, $\hat{x}_d^s$ denotes the expected value of the value function at separation s, given $a_1 = d$ ($d \in \mathcal{L}$):

$$\hat{x}_d^s = \sum_{\lambda \in \Lambda_r^d} P_d(\lambda) x_\lambda^s, \qquad (8)$$

where Λ_R^d denotes a subset of Λ_r containing the λ's with $a_1 = d$ and where $P_d(\lambda)$ denotes the sum of the occurrence probabilities of λ_d and its symmetries in Λ^d, subset of Λ containing the λ's with $a_1 = d$. Note that for symmetry reasons, we need not distinguish between the cases where respectively a_1 and a_2 are known. Then, for all $s \geq 4$, and introducing slack variables $\mathbf{y}^s \geq 0$, we transform the inequality constraints into equality constraints leading to the following standard form formulation:

$$\begin{aligned} &\text{maximize} && \mathbf{e}^T \mathbf{x}_r^s && (9) \\ &\text{subject to} && \mathbf{A} \begin{pmatrix} \mathbf{x}_r^s \\ \mathbf{y}^s \end{pmatrix} = \mathbf{b}^s \\ &&& \mathbf{x}_r^s, \mathbf{y}^s \geq 0, \end{aligned}$$

where $\mathbf{e} = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}^T$, $\mathbf{A} = \begin{pmatrix} \mathbf{A}' & \mathbf{I} \end{pmatrix}$, with $\mathbf{I}$ the identity matrix of appropriate size and $\mathbf{A}'$ only function of parameters ε and p, and where the elements of $\mathbf{b}^s$ are affine functions of $\hat{x}^{s-2}$, $\hat{x}_0^{s-1}$, and $\hat{x}_1^{s-1}$. Remark that only $\mathbf{b}^s$ depends on value functions at lower separations. Note that the constraint $\mathbf{x}_r^s \geq 0$ is redundant, but necessary for the standard form formulation.

IV. Main Results

Our main result, *i.e.* Lemma 1, shows that after converging towards each other, the maximum agent separation of a two-agent cluster navigating on $\mathcal{G}$ under an optimal policy is bounded above by three. This is established in the following lemma, using the combined LP-DP formulation of Section III-B. In particular, let μ^* denote the optimal navigation policy for the two-agent cluster:

$$\mu^*(\chi) = \arg\min_{\mathbf{u} \in U} E[g(\chi, \mathbf{u}) + \varepsilon J^*(f(\chi, \mathbf{u}))].$$

Let $\{s_k\}$, $k = 0, \dots$ denote the agent separation at stage S_k, under μ^*. Note that s_0 is the initial agent separation. Let S_d, $d \geq 0$, be the first stage where $s_k \leq 3$. Then:

Lemma 1: Given $\varepsilon = 0.9$, and $p = 0.5$ (uniformly distributed link costs over $\mathcal{L}$). Then, for $k < d$, the separation sequence $\{s_k\}$ is such that $s_k \geq s_{k+1}$, where equality can only hold for $\mathbf{u} = (0,0)$ at S_k. For $k \geq d$, on the other hand, the separation sequence $\{s_k\}$ satisfies: $\max_k\{s_k\} \leq 3$.

Proof: Let $J(\chi)$ denote the following value function: $J(\chi) = J^3(\chi)$, for $s \leq 3$, and $J(\chi) = x_\lambda^s$ for $s \geq 4$. Let μ denote the policy corresponding to J:

$$\mu(\chi) = \arg\min_{\mathbf{u} \in U} E[g(\chi, \mathbf{u}) + \varepsilon J(f(\chi, \mathbf{u}))].$$

Shortly, we establish that $J(\chi)$ satisfies the Bellman equation (1), implying $J(\chi) = J^*(\chi)$ and policy $\mu(\chi) = \mu^*(\chi)$. It then follows (see Sec. III-B) that $\mu^*(\chi)$ corresponds to the following policy: for $s \leq 3$, $\mu^*(\chi) = \mu^3(\chi)$, and for $s \geq 4$, $\mu^*(\chi)$ is the policy as described in Fig. 2. It can be seen that policy $\mu^*(\chi)$ leads to a separation trajectory $\{s_k\}$ that is such that for $s_k \leq 3$ implies $s_{k+1} \leq 3$. Furthermore, $s_k \geq s_{k+1}$, for $s_k \geq 4$, where equality can only hold for $\mathbf{u} = (0,0)$ at S_k. Thus, policy $\mu^*(\chi)$ has the properties called for in the lemma statement. It remains to be shown that $J(\chi)$ satisfies the Bellman equation.

For this purpose, three properties of $J(\chi)$ need to be established. First, $J(\chi)$, for $s \leq 3$, satisfies the Bellman equation. Second, it needs to be established that for $s \geq 4$, no information zone exists where a decision in $U \setminus \overline{U}$ is preferable to a decision in $\overline{U}$. Third, it needs to be established that the bfs described in Fig. 2 is the optimal bfs for all LPs corresponding to separations $s \geq 4$. These three properties are established as follows.

For $\varepsilon = 0.9$ and $p = 0.5$, the first property can easily be verified numerically. Lemma A.1 establishes the second property by demonstrating that for each decision $\mathbf{u}_b \in U \setminus \overline{U}$

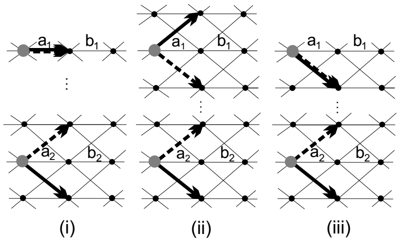


Fig. 3. The three instances where a decision $\mathbf{u}_b \in U \setminus \overline{U}$ (bold arrows) is shown to be worse than a decision $\mathbf{u}_d \in \overline{U}$ (dashed arrows). The other two decisions $\mathbf{u}_b \in U \setminus \overline{U}$ are symmetric cases.

(see Fig. 3, *b*old arrows), there is a preferable decision $\mathbf{u}_d \in \overline{U}$ (see Fig. 3, *d*ashed arrows). For example, in the case in Fig. 3 (i), $\mathbf{u}_d$ is preferable to $\mathbf{u}_b$ if:

$$a_1 + \bar{l} + \varepsilon \hat{x}_{b_1}^{s-1} < a_1 + \bar{l} + \varepsilon \hat{x}_{b_1}^{s+1},$$

where s is the separation at the current stage. Lemma. A.1 implies that this condition is satisfied for all a_1 and b_1. A similar reasoning holds for the cases is Fig. 3 (ii) and Fig. 3 (iii).

The third property is established as follows. A sensitivity analysis shows that for all $\mathbf{b}^s$, $s \geq 4$, the same bfs is optimal. Let $\mathbf{B}^s$ denote the optimal basis for the LP (9) corresponding to separation s. Note that

$$\begin{pmatrix} \mathbf{x}_r^s \\ \mathbf{y}^s \end{pmatrix} = (\mathbf{B}^s)^{-1}\mathbf{b}^s.$$

Taking the limit for $s \to \infty$ of (6), we obtain $\mathbf{x}^\infty$:

$$\mathbf{x}_r^\infty = (\mathbf{I} - (\mathbf{C}_1 + \mathbf{C}_2))^{-1}\mathbf{D}.$$

Using (7) and (8), we can compute $\hat{x}^\infty$, and $\hat{x}_d^\infty$, for all $d \in \mathcal{L}$. Further, vector $\mathbf{b}^s$, for $s \geq 4$, can be expanded as follows:

$$\mathbf{b}^s = \mathbf{b}_0 + \hat{x}_0^{s-1}\mathbf{b}_1 + \hat{x}_1^{s-1}\mathbf{b}_2 + \hat{x}^{s-2}\mathbf{b}_3,$$

where $\mathbf{b}_v$, $v = 0, 1, 2, 3$ are constant vectors. Let $\delta_{(\cdot)}^s = \hat{x}_{(\cdot)}^\infty - \hat{x}_{(\cdot)}^s$. From Lemma A.1 we know that the sequences $\{\hat{x}_{(\cdot)}^{2k}\}$ and $\{\hat{x}_{(\cdot)}^{2k+1}\}$, for $k \geq 2$, converge monotonically increasing to $\hat{x}_{(\cdot)}^\infty$. Therefore the following set contains $\mathbf{b}^{2k}$, $k \geq 3$:

$$\Gamma^e = \{\mathbf{b} | \mathbf{b} = \mathbf{b}^6 + \theta_1\delta_0^5\mathbf{b}_1 + \theta_2\delta_1^5\mathbf{b}_2 + \theta_3\delta^4\mathbf{b}_3, \\ 0 \leq \theta_1, \theta_2, \theta_3 \leq 1\},$$

and similarly, Γ^o contains $\mathbf{b}^{2k+1}$, $k \geq 3$:

$$\Gamma^o = \{\mathbf{b} | \mathbf{b} = \mathbf{b}^7 + \theta_1\delta_0^6\mathbf{b}_1 + \theta_2\delta_1^6\mathbf{b}_2 + \theta_3\delta^5\mathbf{b}_3, \\ 0 \leq \theta_1, \theta_2, \theta_3 \leq 1\}.$$

We can check numerically that the bfs is optimal for separations $s \leq 7$. Optimality conditions (*i.e.* dual feasibility conditions) are not affected by a change in $\mathbf{b}^s$. We therefore only need to examine the primal feasibility conditions

$$(\mathbf{B}^6)^{-1}(\mathbf{b}^{2k}) \geq \mathbf{0}, \text{ and} \quad (10)$$

$$(\mathbf{B}^7)^{-1}(\mathbf{b}^{2k+1}) \geq \mathbf{0}, \text{ for } k \geq 3. \quad (11)$$

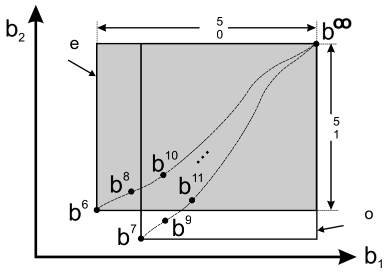


Fig. 4. Conceptual illustration of establishing that suggested bfs remains optimal for all $s \geq 6$. Identify a set of regions Γ that contain $\mathbf{b}^s$ for all $s \geq 6$. Then show that for all $\mathbf{b}^s$ in the set of regions primal feasibility conditions hold.

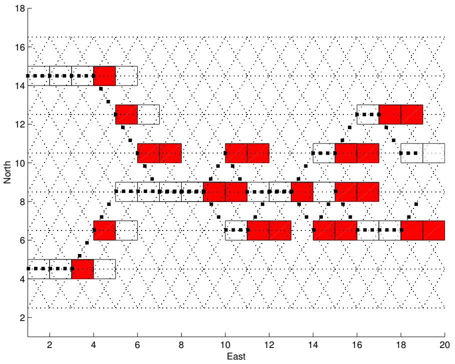


Fig. 5. Simulation example. The black squares represent the agents: A_1 moves to the north of A_2. The target is located arbitrarily far eastward. The dashed line represents the grid, the squares denote the information zones. Red (white) squares represent a link cost of one (zero).

Since all $\mathbf{b}^{2k} \in \Gamma^e$ and all $\mathbf{b}^{2k+1} \in \Gamma^o$, it suffices to check (10) for all $\mathbf{b} \in \Gamma^e$ and (11) for all $\mathbf{b} \in \Gamma^o$. This can be verified numerically to hold by checking the conditions for the eight corner points of each set Γ. Fig. 4 illustrates the principle conceptually for a fictitious 2D example. This concludes the proof. ∎

V. Discussion

In practice, the agents only have limited local information. The local information communicated by an agent is of any navigational use to another agent only if it can reach that zone. Indeed, Lemma 1 shows that the members of a two-agent cluster, navigating optimally, remain within close distance after reaching 'steady state' (i.e. after converging). It quantifies the intuition that only when the agents stay close together can an agent exploit local information observed by the other (see Fig. 5 for an example). Note that due to the initial large separation the agents converge first after which steady state cooperative behavior is reached. In addition, Lemma 1 provides specifications on the communication infrastructure which the members of a two-agent cluster need. In particular, the upper-bound on maximum agent separation in steady state is directly related to the maximum communication power needed.

From a technical point of view, the method presented in Sections III-B and IV provides a tractable solution to an infinite state space problem. In particular, the two-agent navigation on an infinite sized graph $\mathcal{G}$ can be modelled suitably using DP. However, computing the optimal value function using traditional numerical algorithms is impossible. The maximum separation property, *i.e.* Lemma 1, implies that it suffices to compute the optimal value function for only finitely many separations, thereby avoiding the curse of dimensionality. In particular, given an initial configuration $\mathcal{C}_q$ at stage S_0, we merely need J^* for separations $s \leq q$. For $s \leq 3$, J^* can be computed exactly solving a simple DP problem (see (2) with $\bar{s} = 3$); (6) yields J^* for $4 \leq s \leq q$ recursively. Intuitively, increasing the number of agents leads to more efficient task execution; on the other hand, the computational complexity increases significantly. It seems that for n agents to navigate efficiently, agent subgroups need to remain within close distance as inferred from Lemma 1. Therefore, similar approaches for n-agent clusters can significantly reduce the computational complexity and provide extensions to the spatial distribution results obtained in this paper. The presented approach is the basic framework to solve a wide range of multi-agent navigation problems. Extensions involving more agents with different information zones, time-varying irregular environments, and asyncronously movement come to mind.

VI. Conclusion and Future Directions

The problem of coordinated navigation by two cooperating sensor-equipped agents in a partially known static environment is investigated. For a specific local information zone, and under technical conditions, we establish in Lemma 1 that the spatial distribution of a two-agent cluster optimally navigating on an arbitrarily large structured graph is upper-bounded. The presented combined DP-LP approach is the basic framework for the computation of optimal strategies and spatial distribution results in more general cases of the local information zone and for n-agent clusters ($n \geq 3$).

VII. Acknowledgements

This research is supported in part by Cooperative Control of Distributed Autonomous Vehicles in Adversarial Environments, 0205-G-CB222, University of California and by STTR16, Reinforced Learning and Genetic Learning Classifier Systems for Sensor Management, Nascent Technology Corporation.

VIII. Appendix

Lemma A.1: We have

$$\mathbf{x}_r^s \leq \mathbf{x}_r^{s+2} \quad \forall s \geq 2 \tag{A.1}$$

$$\hat{x}^s \leq \hat{x}^{s+2} \quad \forall s \geq 2 \tag{A.2}$$

$$\hat{x}_d^s \leq \hat{x}_d^{s+2} \quad \forall d \in \mathcal{L},\ \forall s \geq 2, \tag{A.3}$$

where the variables are as defined in Section III.

Proof: We give a proof by induction. Let $\mathbf{x}_r^s = J^3(\chi)$, for $0 \leq s \leq 3$. The initialization step, *i.e.* $\mathbf{x}_r^s \leq \mathbf{x}_r^{s+2}$ for $s = 2$ and 3, can easily be checked numerically to hold. The induction goes as follows. Given $\mathbf{x}_r^{l-2} \leq \mathbf{x}_r^l$, for $l \leq k$. From (6), we have

$$\mathbf{x}_r^{k-1} = \mathbf{C}_1\mathbf{x}_r^{k-2} + \mathbf{C}_2\mathbf{x}_r^{k-3} + \mathbf{D}, \text{ and} \tag{A.4}$$

$$\mathbf{x}_r^{k+1} = \mathbf{C}_1\mathbf{x}_r^{k} + \mathbf{C}_2\mathbf{x}_r^{k-1} + \mathbf{D}. \tag{A.5}$$

Subtracting Eqs. (A.5) and (A.4), gives

$$\mathbf{x}_r^{k+1} - \mathbf{x}_r^{k-1} = \mathbf{C}_1(\mathbf{x}_r^{k} - \mathbf{x}_r^{k-2}) + \mathbf{C}_2(\mathbf{x}_r^{k-1} - \mathbf{x}_r^{k-3}).$$

From the induction hypothesis we have

$$\begin{aligned} \mathbf{x}_r^{k} - \mathbf{x}_r^{k-2} &\geq 0, \text{ and} \\ \mathbf{x}_r^{k-1} - \mathbf{x}_r^{k-3} &\geq 0, \end{aligned}$$

and thus, since $\mathbf{C}_1 \geq 0$ and $\mathbf{C}_2 \geq 0$, we have that

$$\mathbf{x}_r^{k-1} \leq \mathbf{x}_r^{k+1}.$$

The inequalities (A.2) and (A.3) follow from (A.1). Hence the proof.

IX. References

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

[2] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997.

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

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

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

[6] J. De Mot, V. Kulkarni, S. Gentry, and E. Feron. Spatial distribution results for efficient multi-agent navigation. In *IEEE CDC*, 2002.

[7] 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.

[8] G. Ribichini and E. Frazzoli. Energy-efficient coordination of multiple-aircraft systems. In *IEEE Conference on Decision and Control*, 2003.

[9] S. Thrun. A probabistic online mapping algorithm for teams of mobile robots. In *International Journal of Robotics Research*, volume 20(5), pages 335–363, 2001.

[10] R. Vidal, O. Shakernia, J. Kim, D. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation. In *IEEE Transactions on Robotics and Automation*, October 2002.