Multiple UAV Cooperative Path Planning via Neuro-Dynamic Programming

Dario Bauso, Laura Giarré and Raffaele Pesenti

Abstract— **In this paper, a team of n Unmanned Air-Vehicles (UAVs) in cooperative path planning is given the task of reaching the assigned target while i) avoiding threat zones ii) synchronizing minimum time arrivals on the target, and iii) ensuring arrivals coming from different directions. We highlight three main contributions. First we develop a novel hybrid model and suit it to the problem at hand. Second, we design consensus protocols for the management of information. Third, we synthesize local predictive controllers through a distributed, scalable and suboptimal neuro-dynamic programming (NDP) algorithm.**

I. Introduction

Cooperative path planning is usually a subtask of the broader class of cooperative search problems also including cooperative target assignment, coordinated UAV intercept, feasible trajectory generation and asymptotic trajectory following [1]. Cooperative path planning is based on the novel notion of coordination variables and coordination functions [5], In [2] a number of suboptimal approximate DP algorithms are developed, which reduce computational complexity to polynomial on the number of air-vehicles. In [7] the idea is to control the planar dynamics of a group of UAVs by an artificial potential force, to avoid collisions, plus an alignment force, to attain a common heading for the vehicles. A real time method to solve optimal path planning problems under cooperative conflict avoidance is developed in [6]. In [4] cooperative conflict avoidance is possible through a twofold team-game theoretic approach.In [8] uncertainty of the motions of other aircraft is modeled within the framework of non-cooperative zero-sum dynamic games.

In this paper, a team of UAVs is given the task of searching a region with potential hazards and opportunities. The common objective is to maximize reward from visiting targets, while avoiding threats. To enhance the element of surprise or to provide different perspectives, each target is prosecuted simultaneously in minimum time with multiple UAV from different directions. Assume that a task assignment problem is solved at high level, such that subgroups of UAVs are assigned to single targets as depicted in Fig. 1 (a subgroup of three UAVs assigned to one target). UAVs assigned to the same target communicate each other their position and heading thus to coordinate (align) their paths towards the target. To solve the above problem we build an hybrid model, able to connect the n decoupled motion dynamics with n coupled dynamics describing the information flow among the UAVs. Given this, we design consensus protocols to extract an efficient coordination variable and propose a suboptimal and scalable NDP algorithm for the optimal synthesis of the local controllers [3].

II. Hybrid Model

A. System Dynamics

UAVs solve the first subtask (threat avoidance) by discretizing the xy-plane through a Voronoi Map as discussed in the following subsection.

1) Voronoi Map: Starting from threats, vehicles and target position, we construct the Voronoi Graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ as in Fig. 1. Each vehicle is located in a node $v'_{i'} \in \mathcal{V}'$, where $i' \in \Gamma' := \{1, 2, ..., n'\}$, and each path segment is an edge $e = (v'_{i'}, v'_{j'}) \in \mathcal{E}'$; $i', j' = 1, 2, ..., n'$. The Voronoi Graph is planar and partitions the plane in regions called cells. It reduces the cooperative path planning to a finite dimensional graph search.

D. Bauso is with DINFO, Università di Palermo, 90128 Palermo, Italy bauso@ias.unipa.it

L. Giarré is with DIAS, Università di Palermo, 90128 Palermo, Italy giarre@unipa.it

R. Pesenti is with DINFO, Università di Palermo, 90128 Palermo, Italy pesenti@unipa.it

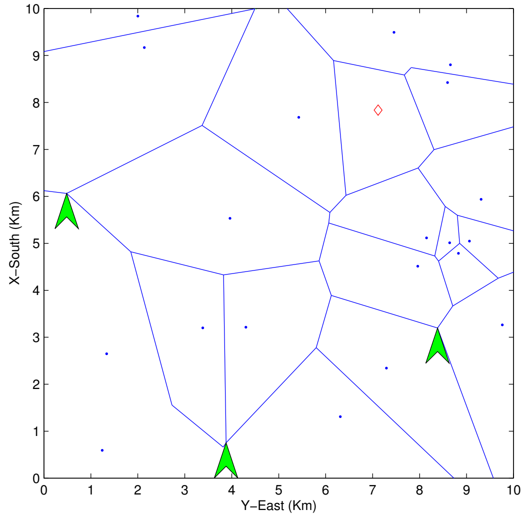

Fig. 1. Region explored by UAVs (green arrowheads), with potential threats (blue circles) and one target (red diamond). Bi-dimensional space discretization via Voronoi Map (blue solid lines).

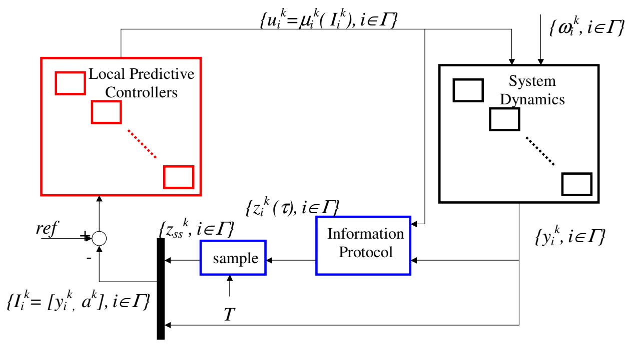

Fig. 2. Block Diagram of the closed loop inventory system.

We indicate with ω_i^k the position of the target assigned to the group of vehicles. We assume that the target is fixed and its position is perfectly known by all vehicles. Thus indices i and k may be dropped. We, also, define the set of vertices $D := \{v_i'\}$ delimiting the Voronoi cell that includes the target itself, i.e., $\omega \in cell(D)$. We call the set D the *final domain*.

2) UAVs Motion Dynamics: Consider a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; each UAV is a node $v_i \in \mathcal{V}$, where $i \in \Gamma := \{1, 2, ..., n\}$, and each communication link is an edge $e = (v_i, v_j) \in \mathcal{E}$; $i, j = 1, 2, ..., n$. Let $n = |\mathcal{V}|$, where $|S|$ indicates the cardinality of the set S.

Assume $x_i^k \in \Gamma'$ be the index indicating the vertex where the ith vehicle is located at instant k. Decision $u_i^k \in \Gamma'$ is the index of the next vertex to reach. Thus, the formation flight dynamics can be described as

$$x_i^{k+1} = u_i^k, \quad \text{for all } i \in \Gamma. \tag{1}$$

The UAVs may only move from one vertex to any neighbor one without entering the final domain except at the last two stages $N-1, N$. In any case all paths must end in a vertex of the final domain D at stage N. Indicating with $H_{x_i^k}$ the set of neighbor nodes reachable in one-step by the ith UAV at stage k, the above constraints are described as follows

$$u_i^k \in U_i^k(x_i^k) = \begin{cases} H_{x_i^k}\backslash D & k < N-1 \\ H_{x_i^k} & k = N-1 \\ H_{x_i^k} \cup D & k = N \end{cases}, \quad \text{for all} i \in \Gamma.$$

The ith output y_i^k, referred to as sensed information, is

$$y_i^k = x_i^k, \tag{2}$$

i.e., each UAV observes only its position.

B. Consensus Protocols

The information flow is managed through a *distributed* protocol $\Pi = \{(f_i, h_i, \phi_i) : \text{for all} i \in \mathcal{V}\}$

$$\dot{z}_i^k(\tau) = f_i(z_j^k(\tau), \text{ for all } j \in N_i),\ 0 \leq \tau \leq T \tag{3a}$$
$$z_i^k(0) = h_i(y_i^k), \tag{3b}$$
$$a^k = \phi_i(z_{ss}^k), \tag{3c}$$

where:

- $f_i : \Re^n \rightarrow \Re$ describes the dynamics of the transmitted information of the ith node as a function of the information both available at the node itself and transmitted by the other nodes, as expressed in (3a);
- $h_i : \mathcal{Z} \rightarrow \Re$ generates a new transmitted information vector given its state at the kth stage, as described in (3b);
- $\phi_i : \Re \rightarrow \mathcal{Z}$ estimates, based on current information, the aggregate info (3c).

Here N_i is the neighborhood of the ith UAV, $N_i = \{j \in \Gamma : (v_i, v_j) \in \mathcal{E}\} \cup \{i\}$, i.e., the set of all the UAVs j that are connected to i and i itself and

$$z_{ss}^k = \lim_{\tau \rightarrow T^-} z_i(kT + \tau), \quad \text{for all } i \in \Gamma,$$

represents the steady state value assumed by $z_i^k(\tau)$ within the interval $[kT, (k+1)T]$. For given scenario, defined by the full state vector, $x^k = \{x_i^k, \text{ for all } i \in \Gamma\}$, the converging value of the transmitted information, a_i^k (coordination variable), plus the sensed information, y_i^k constitute the *partial information* vector, $I_i^k = [y_i^k, a_i^k]$ available to the ith UAV.

C. Local Predictive Controllers

During the approaching maneuver, the UAVs seek to synchronize their arrival time on the target, while at the same time minimizing it. On this purpose, first, the UAVs cooperatively select the minimum time over target N. For sake of simplicity and without loss of generality, this corresponds to the minimum number of steps to reach the target by all UAVs. Actually, this last assumption is realistic when UAVs implement speed control. Note that N will depend on the distance of the furthest UAV from the target. Then, each vehicle chooses the path thus to let the formation center move as fast as possible to the target.

The local controllers compute the following cost over a finite horizon with length N

$$J_i(I_i^k, u_i^k) = g_i(I_i^N) + \sum_{\hat{k}=k}^{N-1} (\alpha^{\hat{k}} g_i(I_i^{\hat{k}}, u_i^{\hat{k}})) \quad (4)$$

where α^k is the discount factor at time t. The stage cost $g_i(I_i^k, u_i^k, k)$ penalizes the distance of the center mass from the target,

$$g_i(I_i^k, u_i^k, k) = \|a_i^k - \omega\|_2^2, \quad \text{for all } i \in \Gamma. \quad (5)$$

Equation (4) represents the cost incurred by the ith UAV over the finite horizon window.

To compute the cost, the controllers must predict the evolution of the information vector I_i^k upon which the stage cost is defined. From (1) and (2), prediction is possible through the following equation

$$\hat{I}_i^{k+1} = \begin{bmatrix} \hat{y}_i^{k+1} \\ \hat{a}_i^{k+1} \end{bmatrix} = \begin{bmatrix} u_i^k \\ \psi_i(a_i^k, u_i^k) \end{bmatrix} \quad (6)$$

where $\psi_i(a_i^k, u_i^k)$ is a simulation-based tunable predictor.

We report hereafter the formalization of the problem under concern.

Given a team of UAVs reviewed as dynamic agents of a communication network with topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

Problem (Local Controllers Synthesis) *For each ith UAV, determine the path planning policy $u_i^k = \mu(I_i^k)$, that minimizes the N-stage individual payoff defined in (4).*

Subproblem (Protocols Design) *Determine a distributed protocol Π in order to extract an efficient coordination variable, a^k .*

III. Path Planning with Full Information

As benchmark scenario, we study in this section the Multiple UAV Cooperative Path Planning Problem with full information on the state. This corresponds to assume that the ith UAV knows the current position of all other UAVs at each stage.

The Full Information Algorithm shown below receives as input: the time over target, i.e., the horizon length N, the number of UAVs n, all starting positions x^0, the target position ω, and the Voronoi Graph, specified in the set of parameters $\Theta = \{G'\}$.

(Full Information Algorithm)

Input. $N, \Gamma, x^0, \omega, \Theta$.
Step 1. FOR k from 1 to N,
  FOR $i \in \Gamma$,
   Compute U_i^k and R_i^k.
Step 2. FOR k from N to 1,
  FOR each $x^k \in R^k = \{R_i^k; i \in \Gamma\}$,
   FOR each $u^k \in U^k = \{U_i^k; i \in \Gamma\}$,
    FOR each $i \in \Gamma$,
    verify NE via backwards DP:
    $\mu_i^k(x^k) = \text{argmin}_{u_i^k \in U_i^k(x^k)} g_i^k(x^k, u^k) + \alpha^{k+1} J_i^{k+1}(x^{k+1})$.
Step 3. Simulate forwards for given x^0.
Return. All NE strategies, paths and costs.

Lemma 3.1: The Full Information Algorithm applied to the Multiple UAV Cooperative Path Planning Problem returns all Nash equilibrium path planning strategies, paths and costs for given initial positions x^0. The computational complexity is exponential on the number of UAVs, $O(nN\Delta^{n(N+1)})$, where Δ is the maximum Voronoi graph degree.

Proof: Let us compute the complexity of the two steps of the algorithm. As regards Step 1, the set of feasible decisions, $U_i^k(x_i^k)$ has cardinality at most $|U|$. Thus, one needs $|U|$ computations for each agent $i \in \Gamma$, for each stage $k = 1, 2, ..., N$, and for each state $x_i^k \in R_i^k$. To get the set of reachable states, R_i^k one needs other $|R|$ computations. Thus, complexity of Step 1 is $O(nN|R||U|)$.

As regards Step 2, for each stage $k = 1, 2, ..., N$, the whole space of reachable states R^k requires at most $|R|^n$ computations. The set of feasible decisions U^k implies at most $|U|^n$ computations. To verify whether u^k is a Nash equilibrium one needs n iterations over the agents. Thus, complexity of Step 2 is $O(nN|R|^n|U|^n)$.

The computational complexity is $O(nN|R||U|) + O(nN|R|^n|U|^n) = O(nN|R|^n|U|^n)$. To complete the proof it is sufficient to note that $|U| = \Delta$ and $|R| = \Delta^N$.

However, it must be said that the algorithm presents an additional preprocessing subroutine with respect to the above general algorithm. In particular a feasibility checking is necessary in order to avoid conflicts in the final state. Yet, the preprocessing does not influence the complexity of the algorithm. ■

Example 1: Let us consider a group of three UAVs exploring the region shown in Fig. 3. The UAVs start from three different nodes of the Voronoi Graph $x^0 = [3, 11, 22]'$. The final domain is the set of five vertices of the Voronoi cell (the polygon) that contains the target. The Full Information Algorithm selects the minimum time over target, $N = 5$. Indeed, this is the minimum horizon length that allows all the UAVs to reach the final domain. Then, Fig. 3 displays Nash equilibrium paths as computed by the Full Information Algorithm.

IV. Path Planning with Partial Information

In this section, we study the Multiple UAV Cooperative Path Planning Problem with partial information on the state. This corresponds to assume that the ith UAV knows its current position and the position of the formation center, chosen as coordination variable.

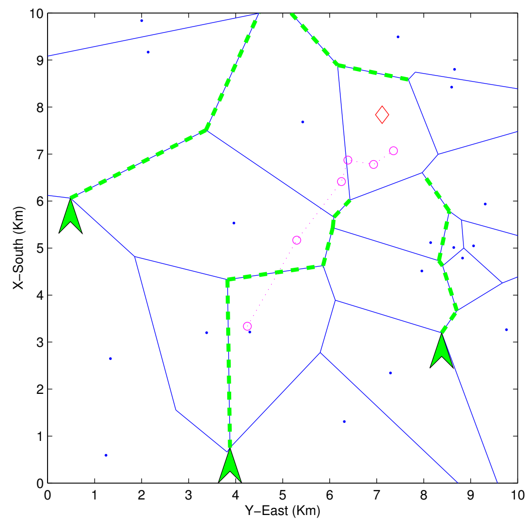

Fig. 3. Nash equilibrium paths (green dotted lines), and trajectory of the formation center (magenta circles) during the approaching maneuver.

A. Formation Center Estimation via Consensus Protocols

In this subsection, we discuss issues concerned with the estimation of the position of the formation center, a^k through distributed consensus protocols.

We assume that the transmitted information is the current estimate of the position of the formation center. The current estimate $z_i(\cdot)$ is re-initialized to x_i^k at the beginning of each time interval $[kT, kT + 1]$.

Thus, we have for each agent $i \in \Gamma$

$$z_i^k = x_i^k.$$

The continuous-time average-consensus protocol takes on the form

$$\dot{z}_i(kT+\tau) := \sum_{j \in J_i} (z_j(kT+\tau) - z_i(kT+\tau)), \quad 0 \le \tau \le T,$$

which we rewrite as

$$\dot{z}_i(kT + \tau) = -L_{i\bullet} z(kT + \tau), \quad 0 \le \tau \le T,$$

where L is the Laplacian matrix of the communication network topology. We can rewrite the protocol in compact form

$$\begin{cases} h_i(x_i^k) & = & x_i^k \\ f_i(z(kT+\tau)) & = & -L_{i\bullet}z(kT+\tau) \\ \phi(z_i(kT+\tau)) & = & n(\lim_{t\to T^-} z_i(kT+\tau)). \end{cases}$$

B. Nash Equilibrium Path Planning

The Partial Information Algorithm, as well as the Full Information Algorithm, receives as input the time over target (the length of the horizon) N, the number n and the initial position of all UAVs x^0, the target position ω, and the Voronoi Graph, specified in the set of parameters $\Theta = \{G'\}$. We discretize the planar position of the formation center by assuming that its coordinates assume only integer values within the set $\mathcal{Z}$, i.e., $x, y = 1, 2, ..., |\mathcal{Z}|$.

(Partial Information Algorithm)

Input. $N, \Gamma, x^0, \omega, \Theta$.
Step 1. FOR k from 1 to N,
FOR $i \in \Gamma$,
compute U_i^k and R_i^k.
Step 2. WHILE not converging,
2a-i. FOR $i \in \Gamma$; get/update $\hat{a}_i^1, \hat{a}_i^2, ..., \hat{c}$
FOR k from N to 1,
FOR each $I_i^k \in (R_i^k \times \Gamma)$,
find NE strategies via DP
$\mu_i^k(I_i^k) \;=\;$ argmin$_{u_i^k \in U_i^k(x_i^k)} g(I_i^k, \imath$
$+ \;\; \alpha^{k+1} \tilde{J}_i^{k+1}(y_i^{k+1}, a^{k+1}).$
2a-ii. Simulate forwards for giver
2a-iii. **(Local Search)**
Perturb the jth state trajectory
$p_i = (x_i^k, x_i^{k+1}, ..., x_i^{k+\Delta}).$
2a-iv. Run the protocol over the
horizon.
Return. NE strategies, paths and costs.

Lemma 4.1: Each iteration of the Partial Information Algorithm applied to the Multiple UAV Cooperative Path Planning Problem, for given initial state x^0, has computational complexity polynomial on the number of UAVs $O(nN|\mathcal{Z}|\Delta^{(N+1)})$.

Proof: The only difference with the Full Information Algorithm is Step 2. As computed in the proof of Lemma 3.1, the complexity of Step 1 is $O(nN|R||U|)$.

Now, it is left to compute complexity of Step 2. We have that the reduced ith state space I_i^k is composed by the space of reachable states R_i^k and the space $\mathcal{Z}$ of values assumed by the aggregated information a^k. For each agent $i \in \Gamma$, for each stage $k = 1, 2, ..., N$ and for each reduced state $I_i^k \in (R_i^k \times \mathcal{Z})$ we must consider the set of feasible decisions U_i^k to write down the Bellman's equation. With a slight abuse of notation we indicate $|\mathcal{Z}|$ the cardinality of the finite set of values of a^k. Thus, complexity of Step 2 is $O(nN|R||\mathcal{Z}||U|)$. The computational complexity of the Partial Information Algorithm is then $O(nN|R||U|)+O(nN|R||\mathcal{Z}||U|) = O(nN|R||\mathcal{Z}||U|)$. To complete the proof it is sufficient to consider that $|U| = \Delta$ and $|R| = \Delta^N$. ∎

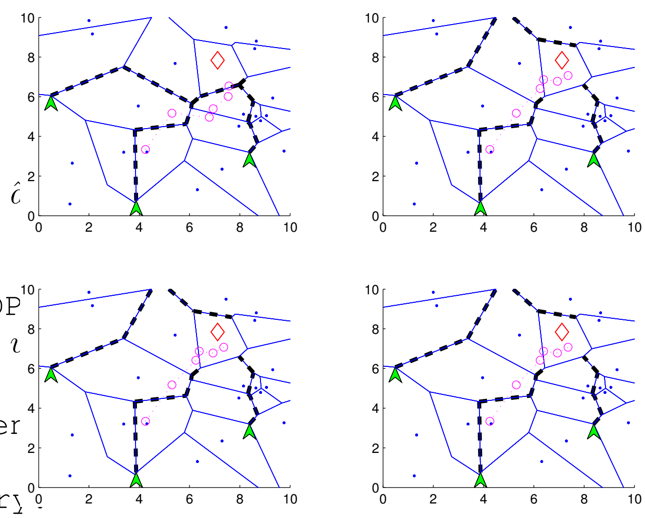

Fig. 4. NE paths (black dashed lines), and trajectory of the FC (magenta circles).

Example 2: (Example 1 cont'd) Let us consider again the case of three UAVs searching a region as in Example 1. At a first iteration, each UAV optimizes its path based only on its own position. No communication occurs among the UAVs. Single paths are shown in Fig. 4, top-left. Note that UAVs approach the target heading differently along the path. No alignment is provided by the team of UAVs. At a second iteration, UAVs communicate their planned single paths through a consensus protocol that returns the predicted position of the formation center. Based on this new aggregated information, UAVs update their planning thus to align their paths as evident from Fig. 4, top-right. In this case the algorithm converges as the third and fourth iterations (bottom-left, bottom-right) return the same path.

V. NDP Solution Algorithm

In this section, we cast the hybrid model within the framework of neuro-dynamic programming (NDP).

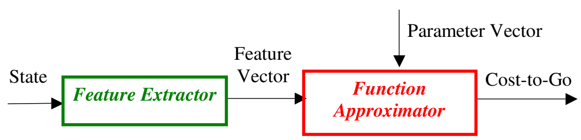

Fig. 5. The information flow management uses consensus protocols to extract the features.

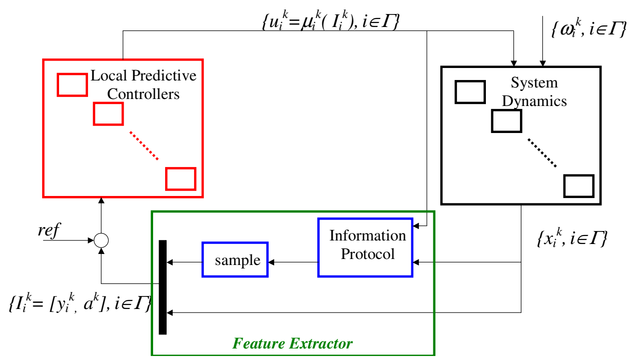

Fig. 6. Block Diagram of the closed loop system.

A. Consensus on Features a_i^k

To review the features as a compact description of the behavior of the other UAVs, we consider i) the NDP architecture based on feature extraction (see e.g. [3]) displayed in Fig. 5 and ii) the block diagram of the Hybrid Model displayed in Fig. 6.

The full state vector of the hybrid model, x^k becomes, in the approximation architecture, the input to the feature extractor. The information flow management block can be reviewed as the feature extractor. The full state vector reduces to the partial information vector $I_i^k = [y_i^k, a_i^k]$ available to the ith UAV. Each local controller implement a function approximator, which receives the partial information vector and returns the individual cost-to-go $\widetilde{J}_i^k(I_i^k, r)$ over the horizon.

B. Linear Architecture

We assume that the probability distribution over all potential values assumed by a^k propagates according to the linear dynamics $a^{k+1} = a^k \Psi^k$ where $\Psi^k = \{\psi_{ij}^k, i, j \in \Gamma\}$. In this case we have i) a matrix of weights r that coincides with the transition probability matrix of the predictor, namely, $r = \Psi = \{\Psi^k, k = 1, 2, ..., N\}$, and ii) *basis functions* $\widetilde{J}_i^{k+1}(I_i^{k+1}, a^{k+1})$ representing different future costs associated to different a^{k+1}.

The approximation architecture linearly parameterizes the future costs associated to all possible behaviors of the other UAVs over the horizon. This can be described as

$$\sum_{a^{k+1}=1}^{|\mathcal{Z}|} \Psi^k_{a^k, a^{k+1}} \widetilde{J}_i^{k+1}(I_i^{k+1}, a^{k+1}) = \psi^k_{a^k \bullet} \widehat{J}_i^{k+1}(I_i^{k+1}, \bullet)^T,$$

where $\psi^k_{a^k \bullet}$ is the row of the transition probabilities from a^k to all possible a^{k+1}, and $\widehat{J}_i^{k+1}(I_i^{k+1}, \bullet)^T$ is the transposed row of the associated future costs.

C. The NDP Algorithm

The NDP Algorithm shown below is organized in two steps. In the first step the UAVs compute the set of admissible decisions U_i^k and reachable states R_i^k over the horizon. The second step presents three substeps.

1) *Policy improvement*. For given prediction Ψ, we improve the policy via the stochastic Bellman's equation backwards in time.
2) *Value iteration*. The improved policy is valued through repeated Quasi-Monte Carlo simulations. Active exploration guarantees that initial states are sufficiently spread over the local minima. During the value iteration we compute and store the number of times a transition Ψ_{ij} occurs during the repeated finite length simulations. At the end of each simulation, the protocol runs over the

horizon and returns the training set for the next step.

3) *Temporal Difference*. We use the training set to update the transition probabilities of the predictor.

The tree substeps are iteratively repeated until convergence of policies.

```
 (NDP Algorithm)

Input. N, Γ, x^0, ω, Θ, numbersim.
Step 1. FOR k from 1 to N,
         FOR i ∈ Γ,
          compute U_i^k and R_i^k.
Step 2.  WHILE not converging,
  2a-i. (policy improvement)
      FOR i ∈ Γ; commit to Ψ,
       FOR k from N to 1,
        FOR each I_i^k ∈ (R_i^k × Γ),
         Stochastic backwards DP:
        μ_i^k(I_i^k)  =  argmin_{u_i^k ∈ U_i^k(x_i^k)} [g_i(I_i^k, u_i^k, k)
                      +  α^{k+1} ψ_{a^k •} J̃_i^{k+1}(I^{k+1}, •)].
  2a-ii. (value iteration)
      FOR simtest from 1 to numbersim,
         - simulate forwards for given x^0
          via active exploration
         - run the protocol over the horizon.
  2a-iii. (temporal difference)
     Update transition probabilities Ψ
Return.NE strategies, paths and costs-to-go.
```

Lemma 5.1: Each iteration of the NDP algorithm, for given initial state x^0, has computational complexity polynomial on the number of UAVs, i.e., $O(nN|\mathcal{Z}|\Delta^{(N+1)})$.

Proof: Step 1 is $O(nN|R||U|)$ as in the Full and Partial Information Algorithm. The main difference with the Partial Information Algorithm is in Step 2. For each policy improvement step, we perform now a number equal to $numbersim$ of stochastic simulations in order to extract the statistics. The central point is to observe that, even in the Partial Information Algorithm the complexity of Step 2a-i is dominant. We extend this consideration to the NDP Algorithm with the only condition that the number of simulations, $numbersim$, during the value iteration, Step 2a-ii, must remain below a certain threshold. This threshold depends on the ratio between the order of complexity of Step 2a-i and Step 2a-ii. ∎

Assuming that convergence is achieved in a finite number of iterations, the NDP Algorithm returns stochastic Nash equilibrium policies, paths and costs-to-go. Further efforts are still to be made, oriented to investigate the convergence conditions of this algorithm.

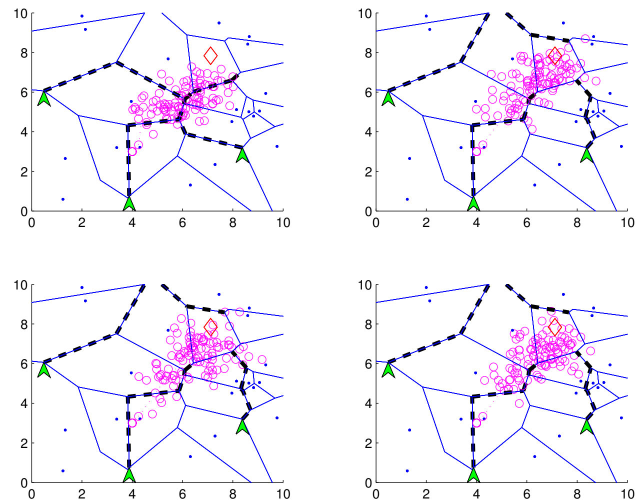

Fig. 7. Convergence in presence of noise.

Example 3: (Example 1 cont'd) Let us consider again the case of three UAVs searching a region as in Example 1. We assume that the estimation of the position of the formation center is affected by white gaussian noise with signal to noise ratio, $snr = 20$. Though convergence is in general not guaranteed, we see from Fig. 7 that for particular initial positions path planning strategies converge very fast even in presence of noise.

By assuming that each UAV may potentially start in three different positions, we performed repeated simulations for a total of $3^3 = 27$ initial states. Fig. 8 displays the individual and global costs (black, red, blue, magenta and green stars) vs the initial positions for iterations 1, 2, 3, 4 and 5 respectively. Note that the costs associated to later iterations reduce progressively, in agreement with what is to be expected from a reinforcement-learning algorithm.

References

[1] R. W. Beard and T. W. McLain and M. A. Goodrich and E. P. Anderson, "Coordinated Target Assignment and Intercept for Unmanned Air Vehicles", *IEEE Trans. on Robotics and Automation*, vol.18, no.6, pp. 911-922, Dec. 2002.

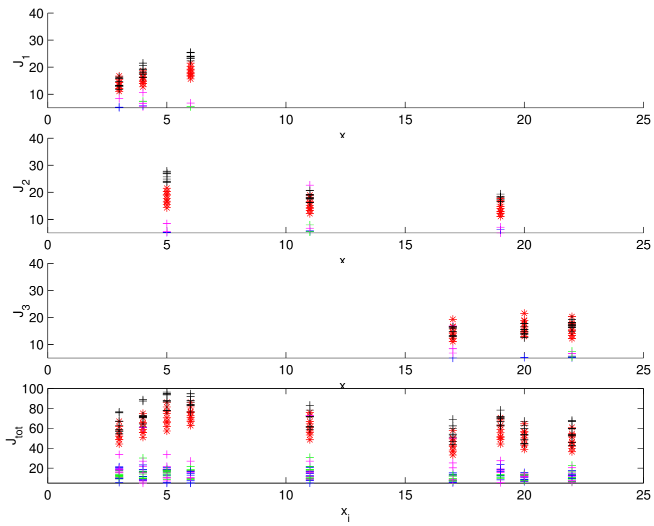

Fig. 8. Individual and global cost-to-go for different initial positions, computed by the NDP Algorithm. White Gaussian Noise (snr=10) affects estimation of the position of the formation center.

[2] R. W. Beard and T. W. McLain, "Multiple UAV Cooperative Search under Collision Avoidance and Limited Range Communication Constraints", *in Proc. of the IEEE Conference on Decision and Control*, Maui, Hawaii, Dec. 2003.

[3] D. P. Bertsekas and J. N. Tsitsiklis,"Neuro-Dynamic Programming", *Athena Scientific*, Bellmont, MA, 1996.

[4] A. Bicchi and L. Pallottino, "On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems", *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 4, Dec. 2000, pp. 221-231.

[5] T. W. McLain and R. W. Beard, "Coordination Variables, Coordination Functions and Coooperative Timing Missions", *in Proc. of the IEEE Conference on Decision and Control*, Maui, Hawaii, Dec. 2003,

[6] L. Pallottino, E. M. Feron and A. Bicchi, "Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming", *IEEE Trans. on Intelligent Transportation Systems*, vol. 3, no. 1, March 2002, pp. 3-11.

[7] H. G. Tanner, A. Jadbabaie and G. J. Pappas, "Coordination of Multiple Autonomous Vehicles", *Proc. of the 11th Mediterranean Conference on Control and Automation*, Jun 2003.

[8] C. Tomlin, G. J. Pappas and S. S. Sastry, "Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems", *IEEE Trans. on Automatic Control*, vol. 43, no.4, Apr 1998, pp. 509-521.