CoUAV: A Multi-UAV Cooperative Search Path Planning Simulation Environment

Jens Happe
MacDonald, Dettwiler & Associates
Ltd. Richmond, B.C., Canada
jhappe@mdacorporation.com

Jean Berger
DRDC Valcartier
Quebec City, Canada
jean.berger@drdc-rddc.gc.ca

Keywords: Unmanned aerial vehicles, multi-agent coordination, path planning, decision support, learning.

Abstract

Sophisticated multi-unmanned aerial vehicle (UAV) simulation environments developed so far intrinsically paid significant attention to high-fidelity flight control system components to realistically account for low-level decision support. However, the use of these simulators often incurs a large overhead when focusing on cooperative high-level decision tasks, such as planning in mobile sensor networks. Therefore, a new discrete-event simulation environment, specially designed to investigate multi-agent search path planning coordination problems for surveillance and reconnaissance is proposed. Named CoUAV, the simulation capability gives the flexibility to define and customize simulation configurations from high-level abstract key components and stochastic events specifically aimed at exploring team coordination strategies for distributed information gathering. It abstracts away costly low-level system specifications. The environment provides the user with problem definition, visualization and post-simulation solution analysis capabilities. The versatility and flexibility of the environment is well-suited to explore the strengths and weaknesses of new and existing coordination strategies through comparative performance analysis over a variety of resource-bounded search path planning problem conditions. As an example, simulation results are presented for a military multi-UAV reconnaissance/target search mission comparing two solution designs.

1. INTRODUCTION

Efficient construction of a recognized air picture (RAP) for military or civil local area surveillance and reconnaissance missions is often critical to ensure and maintain situational awareness. In many cases, given the low cost and risk associated with resource utilization, the related RAP process increasingly relies on a team of mobile sensor agents or unmanned aerial vehicles (UAVs) (both terms are used interchangeably) to close the gap between information need and information gathering. Early work on related search problems emerges from search theory [1], [2]. But most solutions focus on effort allocation rather than path construction. Robot motion planning alternatively explored search path planning, considering constrained shortest path type solutions for coverage problem instances [3]-[5]. Emphasis has then been shifted to self-organization, where teammates autonomously manage their own resources and coordinate their behavior to achieve a common global objective [6]–[10]. Some extensions address the critical information-sharing dimension of cooperative search path planning [11]–[14], considering decentralized partially observable Markov decision processes, combining communication (information-sharing) and control decisions (COM-DEC-POMDP) [15], [16]. Recently, UAV search path planning work started to focus on coupling information-sharing and control action coordination through explicit communication of intents (agent path plans) subject to constrained communication and limited bandwidth. Given that simultaneous problem-solving for communication and control decision variables is intractable [22], solution approaches often resort to simulation to solve them. However, current simulation environments mainly focus on reliable low-level flight control UAV components, typically require significant user knowledge or expertise, incur overhead in managing loosely coupled components and infrastructure, and are possibly prohibitive to suitably adapt or conduct domain-specific experiments [17], [3], [18], [19], [20]. Most importantly, they often pay insufficient attention to key high-level coordination and decision task components and show limited flexibility in accommodating real-world constraints and requirements (e.g. particular coordination strategies and limited bandwidth handling).

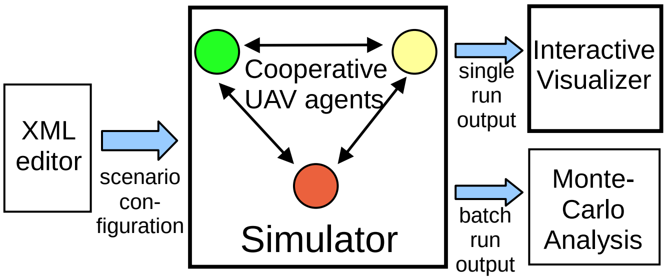


Figure 1. Running simulations with CoUAV. The simulator and visualizer are part of the environment (thick borders). Scenario configurations can be edited, and batch run outputs analyzed, using off-the-shelf XML editors and statistical analysis packages, respectively.

This paper presents a flexible discrete-event simulation environment, named CoUAV, to explore and analyze distributed resource management decision models and algorithms for multi-UAV cooperative search path planning in surveillance and reconnaissance missions. The environment supports the user with problem definition, visualization and post-simulation solution analysis capabilities, as drawn schematically in Figure 1. A scenario can be configured from a variety of different component models, which are abstract (representing the environment as a discrete cellular grid of target probabilities) but aim to add realistic constraints, such as fixed-wing motion constraints and communication bandwidth and range limitations. This provides a suitable and flexible capability to investigate the performance of UAV plan coordination strategies and information-sharing policies in a dynamic uncertain environment. The user's selection of component models and its parameters is specified and read into the simulator as an XML file. The simulator can be run either interactively, producing output data that can be displayed in the CoUAV visualizer, or in batch mode, producing statistical data on multiple simulation runs over different configurations, which allows comparative analysis of algorithms and/or parameter settings using an analysis tool such as Matlab.

.

The remainder of the paper is structured as follows. Section 2 first introduces the cooperative search path planning problem, the main problem characteristics, and the assumptions made. It then gives an overview of the component models, functions and capabilities implemented in the Simulator. Section 3 outlines the visualization functions, while section 4 presents the basic environment architecture, the prescribed procedure to initiate and carry out simulations, and the analysis capability. Computational results comparing two solution designs are presented in section 5 for a military multi-UAV reconnaissance/target search mission, validating a novel approach to multi-agent cooperative search using co-evolution proposed in [24]. Finally, a summary and a conclusion are given in section 6.

2. PROBLEM MODELING

2.1. Description

We consider a hierarchical multi-objective problem, in which a team of heterogeneous UAVs cooperatively searches stationary targets in a bounded and uncertain environment over a limited time horizon *L*. Note that all parameters given in this section are fully configurable on an agent-by-agent basis in the simulation environment.

2.1.1. Environment Model

The search environment model largely follows the model proposed in [25]. It discretizes the space into a grid of *N* cells, populated by non-cooperative stationary targets with unknown locations and cardinality. Based on prior domain knowledge, individual cells are characterized by an initial probability of target occupancy, which is user-configurable on a cell-by-cell basis. Starting at individually configured initial positions, a team of *n* autonomous UAVs synchronously explores this environment. A vehicle's configuration state is defined by a given position (cell location) and a specific orientation/heading (*N,S,E,W,NE,SE, SW,NW*). Vehicles are assumed to fly at a constant velocity, reaching a horizontally, vertically or diagonally adjacent cell in one time step and at slightly different altitudes to avoid colliding with each other. Their turning radius is constrained, in that they can change direction by maximally 45 degrees in each time step.

2.1.2. Problem Objectives

The user can choose from a variety of objectives, such as: maximize information gain (or equivalently, minimize team uncertainty or entropy [21] of target occupancy over the grid), minimizing target discovery time, or minimize resource utilization (namely, energy consumption). The solution quality is ranked along the hierarchy of given objectives, in their order as configured: the second objective acts as a tie-breaker for two solutions that are equal with respect to the first objective, etc.

2.1.3. Agent Cognitive State

The agents act as stand-off imperfect sensors, which gather observations. As the system is distributed, each agent must continuously build and update its own information state. Other than through local observations, the agents also learn by exchanging state and plan information with one another. We call each agent within range a "neighboring agent"; these are the only other agents an agent is aware of (small world assumption). A UAV's cognitive map captures its local knowledge of the environment state. It comprises a target occupancy belief distribution, positions and orientations of neighboring agents, its own direction and position, resource level, sensor observations and their sources (observing agent), as well as a past communication log with other agents. A typical agent cognitive map at a given point in time is illustrated in Figure 2**Error! Reference source not found.**. In the current setting, the agents' capabilities (e.g. sensor observation models, maximum communication range and other properties) do not change over time; every agent has prior knowledge about its teammates' capabilities, which it can exploit. For instance, an agent could use it to estimate another agent's information state before deciding whether or not to exchange information.

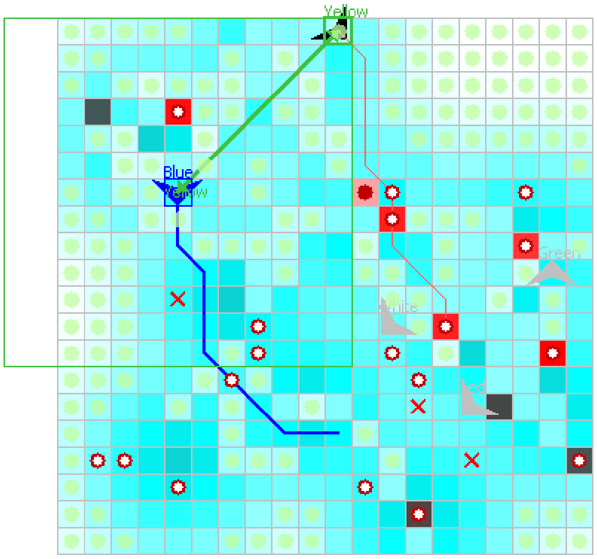


Figure 2. Agent uncertainty/cognitive map at time step *t*. Local agent beliefs are displayed through multi-level shaded cell areas. Projected agent plans are represented as possible paths (blue) and peer-to-peer communications as straight lines (green). Filled circles refer to discovered targets.

2.2. Decision Concept

Cooperative search consists in jointly constructing agent path plans to optimize solution quality along the given team objective(s).Each agent must align its behavior toward reaching this objective. The main multi-UAV target search decision concept is based on an information-theoretic framework to address information-sharing (communication) and cooperative path planning (control) in a environment. Each time step, an agent's decisions are sequentially decomposed in two stages, namely, information–sharing (past observations) and planning. Both information-sharing and path planning are subject to limited on-board computational resources; informatino-sharing is also subject to range and bandwidth constraints. Decisions are made on what information (observations) to exchange at the beginning of the current time step, and what move (control action) to execute during the next time step. Moves planned over the next (receding) plan time horizon are also computed or revised. The time interval Δt_t is divided in two time subintervals, accounting for information-sharing and planning respectively, as shown in Figure 3.

The agent information-sharing policy is based on maximum information/belief divergence in exchanging the most valuable information on state estimation with neighbors. It is based on the premise that maximizing belief consistency (common belief-sharing) among cooperative teammates at each time step is likely to improve the joint planning solution. Combined temporal and communication bandwidth constraints along with message type sizes define a maximum number of messages to be exchanged in both directions during a single episode. Accordingly, a bound on the number of messages have been considered for the information-sharing stage of the decision process. High-value or latest observations are exchanged first between neighbor agents (see section 2.3.4).

The remaining time is devoted to cooperative planning. The agents use open-loop with feedback decision models (e.g. entropy minimization) over a receding horizon subject to limited computational and communication resources, as well as communication cost, Path plans are locally computed by agent algorithms over a given horizon, and the agents periodically exchange intents (action plans) with their neighboring agents. This user-specified coordination mechanism ensures local (neighborhood) team coherence.

Information-sharing communication – state/observations	Planning planning iterations & periodic intent sharing
← $^{s}\Delta t_{c,t}$ →	← $\Delta t_{pl,t}$ →

←Action execution: move to the next cell and observe→

Time step *t* (episode) with duration Δt_t

Figure 3. A two-stage decision process

In order to ensure agent synchronization, we assume that each stage has a constant predetermined duration, which imposes temporal constraints on both information-sharing and planning. Decision-making during the current episode takes place concurrently with action execution. Therefore, during time interval Δt_t, the vehicle executes a previously planned action to move to the next cell, and makes an observation in that cell. The process can be summarized as follows:

```
t=1
Repeat  -- agent search path planning behavior
      Control action execution (planned at t-1)
      Information_Sharing ( $t(^{s}\Delta t_{c,t})$ ) – Stage I
      Path_Planning ( $t(\Delta t_{pl,t})$ ) – Stage II
      Observation and cognitive map update
      End of episode t; t=t+1
Until (end of search mission horizon: t=L+1)
```

2.3. Models

Based on the search environment and global cognitive map state models characterizing the problem introduced in section 2.1, environment component models are further described below.

2.3.1. Observation

The observation model governing a UAV sensor's perception accounts for partial world state observability. During each time step, a UAV visits a cell and searches it for target occupancy. (This assumes that the sensor footprint covers exactly the cell the UAV is visiting.) A sensor reading z_t at time *t* may be positive or negative and is governed by an observation model, which accounts for uncertainty through conditional probability of detection and false alarm, given the cell target occupancy/vacancy state (*X*=1/0):

z_t : observation of cell occupancy at the end of period *t*:

{positive=1, negative=0}

$p_c = p(z_t = 1 \mid X = 1)$ probability of correct observation

$p_f = p(z_t = 1 \mid X = 0)$ probability of false alarm

In addition to probabilistic outcomes introduced by imperfect sensor readings, vehicles have limited sensing ranges within which they can perceive and recognize other agents.

The user can customize the probability of detection on a cell-by-cell basis to capture dependencies or reflect the impact of exogenous events or particular conditions (e.g. weather, topography, vegetation/traffic density, obstacles, clutter, etc.).

2.3.2. Bayesian Filtering

Based on the latest observation, local cell target occupancy beliefs (*p(X=1)*) can be modified using Bayesian filtering for data fusion:

$$p_t(X \mid z_t) = \frac{p(z_t \mid X)\, p_{t-1}(X)}{p(z_t)} \qquad (1)$$

Where

$$p(z_t) = \sum_{x \in \{0,1\}} p(z_t \mid X = x)\, p_{t-1}(X = x) \qquad (2)$$

p_{t-1} and p_t refer to prior and posterior cell target occupancy probability (belief) respectively. Sensor readings may eventually be shared with neighboring agents after each observation episode (cell visit) to further update local beliefs and progressively build a more consistent cognitive map.

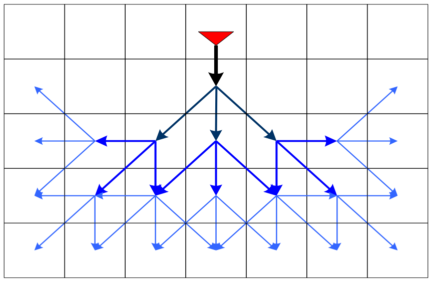


Figure 4. Agent's region of interest displayed as forward move projection span (possible paths) over a 3–step time horizon.

2.3.3. Agent Path Planning

Each agent makes decisions at each episode about the next T cells to be visited over a receding time horizon $t+1,\ldots,\ t+T$. Accounting for physical acceleration, the action decision is limited to three possible directions with respect to its current heading,: *ahead, right* and *left*, as depicted in Figure 4. A path plan is a sequence of T base-level control action decisions; equivalently, it can be viewed as the sequence of configuration states uniquely defined by these action decisions and the agent's current state.

As stated in section 2.1, the primary objective consists in minimizing the entropy (target occupancy uncertainty) over the entire grid. The entropy function E is borrowed from information theory [21]:

$$E = -\sum_{x\in\{0,1\}} p(x)\log_2 p(x)\ , \qquad (3)$$

where p refers to the current probability/belief of cell target occupancy. A cell entropy of 0 (1) means absolute certainty (total uncertainty) about occupancy or vacancy.

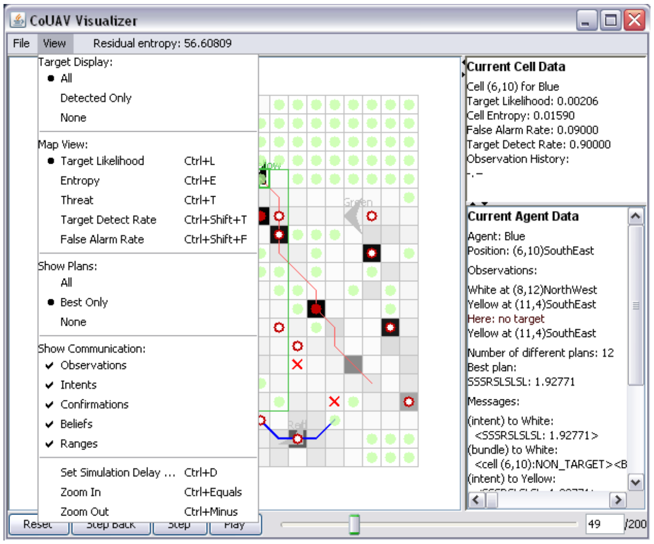
2.3.4. Communication

The agent team primarily behaves as a particular vehicular ad hoc network. Vehicles have self-localization capability, can recognize neighboring agents, and rely on a unicast or peer-to-peer communication scheme based on perfectly reliable communication channels. The model assumes limited communication range restricting neighborhood interaction, and a physically predetermined or pre-allocated finite bandwidth bounding the number of messages to be exchanged with a neighbor in each time step. Agent communication with neighbors takes place concurrently, delivering/receiving messages on separate channels in parallel. Encoded as messages, communication decisions translate into observation streams, beliefs and/or intents to be shared. Based on the aforementioned small-world assumption, we also assume instantaneous message-passing (negligible network latency). The cost of information exchange is measured in terms of energy use, which is quadratic in terms of the distance r connecting two agents ($\alpha\, r^2 + \beta$, where α and β are suitable constants).

3. VISUALIZATION

A simulation run for a specific scenario can be displayed through the CoUAV visualizer, a customizable graphical user interface showing different dimensions of problem specifications, solution visualization and simulation execution. Multiple information layers implemented as cognitive map overlays may enrich the user's perspective of the problem or solution. Overlays can be combined or explored separately trough a pull-down menu (Figure 5) specifying elements on a single map view. These elements can be visualized at any time during simulation execution. These layers can further be customized by the user: right-clicking on an agent/empty cell presents an agent-centric/ team-centric perspective. The data panels on the right display information about the state of the current cell (top) and of the selected agent (bottom) at the current time step. The user simply accesses local cell state information by selecting a specific cell on the map to the left; the corresponding information displayed on the top right section includes the cell coordinates, likelihood/entropy, and observation history. Agent-related state and log information presented on the bottom right pane include position, heading, intents (path plan) and detailed communication information (observation and intent exchange). Integrated into the menu bar at the top, information about the simulation performance is displayed (here: residual entropy summarized over the entire grid).

CoUAV provides an execution timeline at the bottom of the user interface window to select any specific time step over the predetermined simulation horizon. The buttons to the left of the timeline allow the user to reset, move one time step forward or backward, or play a continuous animation of a simulation run. The execution tempo (time step duration) can be adjusted.



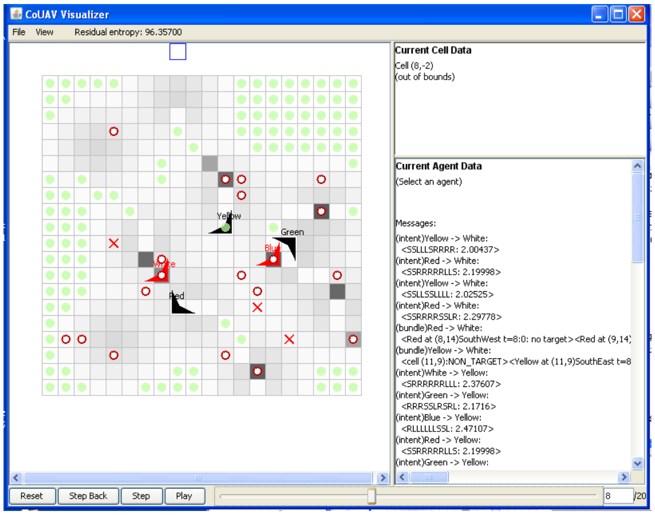
Current Cell Data
Cell (6,10) for Blue
Target Likelihood: 0.00206
Cell Entropy: 0.01590
False Alarm Rate: 0.09000
Target Detect Rate: 0.90000
Observation History:
-.–

Current Agent Data
Agent: Blue
Position: (6,10)SouthEast
Observations:
White at (8,12)NorthWest
Yellow at (11,4)SouthEast
Here: no target
Yellow at (11,4)SouthEast
Number of different plans: 12
Best plan:
SSSRSLSLSL: 1.92771
Messages:
(intent) to White:
<SSSRSLSLSL: 1.92771>
(bundle) to White:
<cell (6,10):NON_TARGET><B
(intent) to Yellow:

Figure 5. Customizable multi-layer cognitive map presentation.

3.1. Target, Threat, Entropy Views

Cognitive map elements such as agent locations and identifiers (triangles with labels), real target locations ('o'), threats ('x'), and cell entropy/ agent beliefs can be displayed by the user anytime. Evolving beliefs (probability) on cell occupancy and entropy distribution are shown in Figure 6–7.

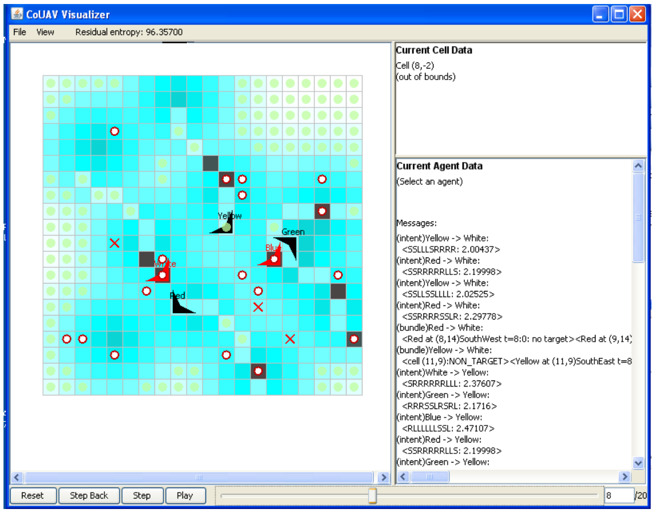


Current Cell Data
Cell (8,-2)
(out of bounds)

Current Agent Data
(Select an agent)

Messages:
(intent)Yellow -> White:
<SSLLLSRRRR: 2.00437>
(intent)Red -> White:
<SSRRRRRLLS: 2.19998>
(intent)Yellow -> White:
<SSLLSSLLLL: 2.02525>
(intent)Red -> White:
<SSRRRRSSLR: 2.29778>
(bundle)Red -> White:
<Red at (8,14)SouthWest t=8:0: no target><Red at (9,14)
(bundle)Yellow -> White:
<cell (11,9):NON_TARGET><Yellow at (11,9)SouthEast t=8
(intent)White -> Yellow:
<SRRRRRRLLL: 2.37607>
(intent)Green -> Yellow:
<RRRSSLRSRL: 2.1716>
(intent)Blue -> Yellow:
<RLLLLLLSSL: 2.47107>
(intent)Red -> Yellow:
<SSRRRRRLLS: 2.19998>
(intent)Green -> Yellow:

Figure 6. Cell occupancy probability distribution represented by different color shades. Ring symbols ('o') indicate real target locations to be discovered by team members. The darker the cell shade, the stronger the occupancy belief. Filled circles correspond to cells confirmed with certainty (red: positive, green: negative).

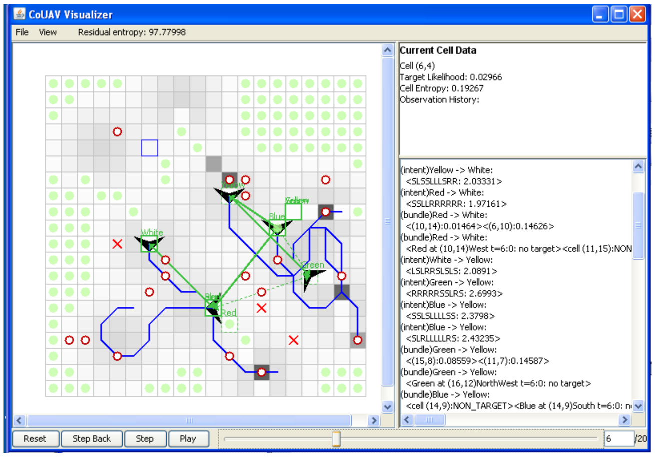


```
Residual entropy: 96.35700

Current Cell Data
Cell (8,-2)
(out of bounds)

Current Agent Data
(Select an agent)

Messages:
(intent)Yellow -> White:
  <SSLLLSRRRR: 2.00437>
(intent)Red -> White:
  <SSRRRRRLLS: 2.19998>
(intent)Yellow -> White:
  <SSLLSSLLLL: 2.02525>
(intent)Red -> White:
  <SSRRRRSSLR: 2.29778>
(bundle)Red -> White:
  <Red at (8,14)SouthWest t=8:0: no target><Red at (9,14)
(bundle)Yellow -> White:
  <cell (11,9):NON_TARGET><Yellow at (11,9)SouthEast t=8
(intent)White -> Yellow:
  <SRRRRRRLLL: 2.37607>
(intent)Green -> Yellow:
  <RRRSSLRSRL: 2.1716>
(intent)Blue -> Yellow:
  <RLLLLLLSSL: 2.47107>
(intent)Red -> Yellow:
  <SSRRRRRLLS: 2.19998>
(intent)Green -> Yellow:

8 /20
```

Figure 7. Entropy distribution represented by different color shades. The darker the cell shade, the larger the entropy (uncertainty about occupancy). Additionally, beliefs are shown as color gradients (blue: low, gray: medium, red: high probability).

3.2. Plans and Communication

Evolving or current best computed plans and communication interactions may be optionally shown to the user at each time step. Agent plans (piecewise linear path starting from an agent) and communication interactions (straight line connecting two agents) may be easily identified in Figures 5 and 8.



```
Residual entropy: 97.77998

Current Cell Data
Cell (6,4)
Target Likelihood: 0.02966
Cell Entropy: 0.19267
Observation History:

(intent)Yellow -> White:
  <SLSSLLLSRR: 2.03331>
(intent)Red -> White:
  <SSLLRRRRRR: 1.97161>
(bundle)Red -> White:
  <(10,14):0.01464><(6,10):0.14626>
(bundle)Red -> White:
  <Red at (10,14)West t=6:0: no target><cell (11,15):NON
(intent)White -> Yellow:
  <LSLRRSLSLS: 2.0891>
(intent)Green -> Yellow:
  <RRRRRSSLRS: 2.6993>
(intent)Blue -> Yellow:
  <SSLSLLLLSS: 2.3798>
(intent)Blue -> Yellow:
  <SLRLLLLLRS: 2.43235>
(bundle)Green -> Yellow:
  <(15,8):0.08559><(11,7):0.14587>
(bundle)Green -> Yellow:
  <Green at (16,12)NorthWest t=6:0: no target>
(bundle)Blue -> Yellow:
  <cell (14,9):NON_TARGET><Blue at (14,9)South t=6:0: n

6 /20
```

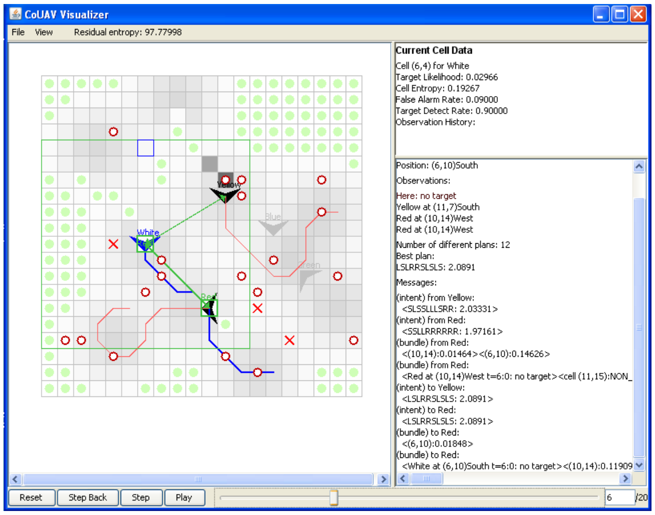
Figure 8. Team-centric perspective at a given time step. Overall team cognitive map resulting from the fusion of local cognitive maps. The right pane contains detailed global state information, including all messages communicated between any two agents in the current time step.

3.3. Team-Centric Perspective

By default, the CoUAV platform displays to the user a team cognitive map combining the cognitive map of all UAV agents. It merges all agents' information about target and entropy distributions, agent plans (or respective current best plans) and team communications as shown in Figure 8. Usually unknown by individual agents, this "collective" map gives the user an understanding of the global simulation performance.

3.4. Agent-Centric Perspective

The single agent-centric perspective delivers local cognitive map information, as shown in Figure 9. Accordingly, the agent position is tagged as a thick blue triangle located on a specific cell, from which the best computed plan (sequence of moves – piecewise connected lines) projected over the next planning horizon is pictured, as well as its interaction with neighbor agents (straight line connecting two agents). Perceived neighborhood is depicted by a green zone centered around the agent position. Agent local cell states (beliefs) and internal agent state are displayed at the top ('Current Cell Data' section) and at the bottom ('Current Agent Data' section) on the right pane respectively. Additional information on best known neighbor intents can also be added.



```
Residual entropy: 97.77998

Current Cell Data
Cell (6,4) for White
Target Likelihood: 0.02966
Cell Entropy: 0.19267
False Alarm Rate: 0.09000
Target Detect Rate: 0.90000
Observation History:

Position: (6,10)South
Observations:
Here: no target
Yellow at (11,7)South
Red at (10,14)West
Red at (10,14)West
Number of different plans: 12
Best plan:
LSLRRSLSLS: 2.0891
Messages:
(intent) from Yellow:
  <SLSSLLLSRR: 2.03331>
(intent) from Red:
  <SSLLRRRRRR: 1.97161>
(bundle) from Red:
  <(10,14):0.01464><(6,10):0.14626>
(bundle) from Red:
  <Red at (10,14)West t=6:0: no target><cell (11,15):NON_
(intent) to Yellow:
  <LSLRRSLSLS: 2.0891>
(intent) to Red:
  <LSLRRSLSLS: 2.0891>
(bundle) to Red:
  <(6,10):0.01848>
(bundle) to Red:
  <White at (6,10)South t=6:0: no target><(10,14):0.11909

6 /20
```

Figure 9. Agent-centric perspective at a given time step, displaying the cognitive map of agent "White" (shown in blue). The view optionally displays the following elements (as shown): the agent's communication range (green frame), its communication with neighbors (straight green lines connecting agents), and its knowledge of their respective best computed plans, as recently communicated to "White". Related agent state information is shown on the right (bottom). Cell-based agent state information is displayed on the right (top).

4. SIMULATOR

4.1. Architecture

The CoUAV simulator architecture has been designed to be modular and extensible. Key CoUAV components such as the planning module, the sensor observation module, and the communication module, are implemented as interfaces. They are instantiated by concrete components through the configuration file. Hence, an experimenter can configure a simulation plug & play -style from different components. Similarly, basic concepts (e.g. measures of performance, entropy, map, information divergence, output logs) can be easily instantiated or specialized to meet specific requirements. Developers can further extend CoUAV by creting new plug-in components instantiating the generic interfaces (e.g. a new path planning algorithm).

Implemented in Java, the simulation environment is platform-independent and runs on any Java Virtual Machine (5.0 or later). Further details may be found in [23].

4.2. Configuration Parameters

CoUAV allows what-if simulations with different target probability distributions, and an exploration or comparison of coordination strategies, path planning and information sharing policies/algorithms, myopic vs. non-myopic scheme, centralized vs. decentralized algorithms, weighted multi-objectives, observation models, sensor capabilities, communication constraints and channel characteristics, among others.

Simulation specifications are represented through an XML configuration file that can be selected through the 'File' menu of the user interface. It captures scenario characteristics including search environment and agent elements (planner, observer, communicator). Search environment attributes are first defined, specifying grid dimensions, team size, number of targets, cell occupancy probability distributions (initial collective belief), and spatial target/threat distribution. Agent sensing/communicating and cognitive properties are featured next, on an agent-by-agent basis. This allows configuring agents with heterogeneous capabilities. Initial position, heading, visual range, communication bandwidth, and sensor parameters characterize the initial agent state, whereas mission goals, measures of performance (e.g. entropy, number of discovered targets, energy consumption) and planning algorithms specify agent behavior. Unicast communication protocols, information-sharing policies, and related parameters may be customized as well. Communication characteristics include range, channel bandwidth, message size requirements and related communication cost individualized by message type, frequencies, and bandwidth allocation for observations, beliefs and intents exchanges, respectively.

4.3. Simulation and Analysis

Based on the aforementioned user-defined configuration, simulation execution may occur interactively (single run, through the CoUAV visualizer) or in batch mode (multiple runs, through the CoUAV command-line version). The former is well-suited to analyze agent behavior during a single simulation run. The latter is specially designed to support Monte Carlo simulation and statistical comparative performance analysis, using off-the-shelf analysis tools. The stand-alone, command-line CoUAV simulator may run multiple times over the same configuration or over different configurations in sequence. Simulation parameters that vary between simulation runs, as well as run-time CoUAV settings (e.g. memory), are easily specified by the user.

4.4. Analysis

The graphical user interface (section 3) with its information labels and panels allows an experimenter to traverse and analyze a simulation run and evaluate details of the agents' behaviors under the chosen configuration and algorithms. The 'File' menu allows saving a human/machine-readable XML log data file used to support 'anytime' simulation execution and/or plan rehearsal. The recording/playback capability gives the user the flexibility to freely navigate through a particular sequence of events and closely investigate agent/team behaviors (control and communication actions, information exchanged, team interactions and information drill-down) at any time step, and consistently move backward or forward as desired. Alternatively, a partial/full simulation run may be recorded into an animated Graphics Interchange Format (GIF) image, which can be displayed by web browsers and image editors, and disseminated to other stakeholders, who could use this animation to validate an overall computed plan or rehearse it. The start and end frame (time step) to be recorded can be interactively selected via a dialog box. Finally, the 'File' menu allows storing performance data over time, in the same format as that produced in batch mode.

Auxiliary post-simulation analysis tools facilitate solution testing/validation, output visualization and performance (cost and solution quality) evaluation of coordination and information-sharing policies, and baseline comparison in terms of performance graphs. CoUAV outputs its performance data in tabular form, which can be easily read into most off-the-shelf analysis tools. Scripts are provided for Matlab and Gnuplot, displaying the performance of one single configuration, or the comparative performance of several different configurations, each averaged over multiple simulation runs, to obtain statistically significant results, along with variance information if required. Using these scripts, graphs such as those shown in section 6 can be created and displayed, depicting entropy, number of discovered targets (true and false), and energy consumption over time, respectively.

5. COMPUTATIONAL RESULTS

A computational experiment has been conducted to illustrate the value of action control coordination and information-sharing. Two approaches were compared, namely, Co-evolutionary Search Path Planning [24] and Approximate Dynamic Programming [11].

5.1.1. Co-evolutionary Search Path Planning

An open-loop agent solution to a multi-objective problem subject to limited computational resources and communication constraints is gradually constructed at each episode and progressively extended over a receding *T*-step horizon, while adjusting its path plan based on additional feedback [24]. Episodic search path planning relies on co-evolution to learn agent coordination. Agents evolve their own population of individuals while sharing information about neighbor agent intents. Individuals are represented as chromosomes encoding for a given time step, a feasible path plan expressed as a sequence of intended control actions (physical moves $a_{t+1} \dots a_{t+T}$) to be executed over a specific time horizon *T.* Agents evolve their own path plan individuals through natural selection, recombination and mutation mechanisms over successive generations while periodically exchanging their best individual with neighbor agent populations. An individual's fitness is determined by combining its own local path plan along with the latest best-known neighbor agents' plans and evaluating the resulting joint plan; overlaps between paths are suitably penalized. The process is then repeated until the end of the planning period. Having thus computed its best path plan, each agent executes the first control action of that plan at the next time step. The anytime coordination algorithm is coupled to an information-sharing policy based on maximum belief divergence in exchanging the most valuable information on state estimation with neighbor. This observation-sharing policy best exploits the limited communication bandwidth, assuming that cooperative autonomous agents maintaining high belief consistency (consensus) are more likely to make decisions reflecting desirable behaviors in the best interest of the team.

5.1.2. Approximate Dynamic Programming

The approximate dynamic programming (limited look-ahead) method reported in [11] assumes for each agent a single step planning time horizon and periodic unicast single hop communication with close (within communication range) neighbors only (small world assumption). In regular intervals of predetermined length T_b, neighboring agents are assumed to communicate their last T_b observation streams, and if two agents become neighbors after being separated for some time, they mutually exchange as much of their respective cognitive map information content as possible. The approach ignores agent intent-sharing, considering its cost too prohibitive.

5.1.3. Performance Results

The performance of the two approaches was compared over a variety of alternate techniques and key problem parameters (e.g. range, bandwidth) for a typical set of scenarios involving 3-10 agents on a 400-cell (20x20) grid. A series of 30 simulation runs was conducted for each scenario, and performance was averaged over all runs. Typical results for some scenarios are presented.

The value of action coordination of the proposed co-evolutionary approach over the baseline approximate dynamic programming (limited look-ahead) method reported in [11] for a 5-UAV team scenario is clearly shown in Figure 10.

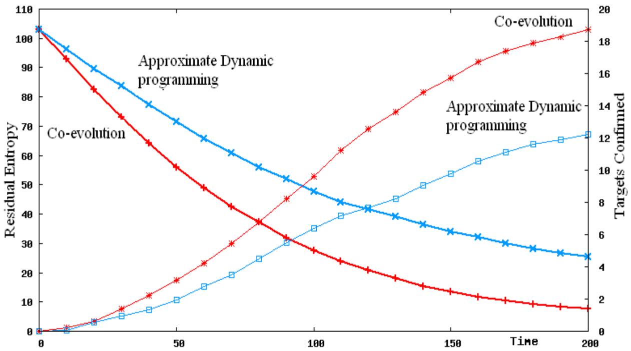


Figure 10. Entropy and confirmed targets over time: co-evolution [24] vs. approximate dynamic programming from [11].

In [11], cooperative path planning ignores explicit communication to coordinate actions but rather relies on local environment sensor information and potential fields to avoid multi-agent interference in concurrently visiting the same cells. The performance gap is explained by explicit action coordination between agents taking place through intent sharing. However, as shown in Figure 11, this gain comes at a higher communication cost, mainly due to continuous neighboring agent interaction in periodically exchanging revised intents, which is omitted in [11]. In contrast, the need to communicate observations decreases over time, as cells increasingly get confirmed. The longer the simulation, the heavier the relative communication burden associated with intents exchange. This overhead could be reduced by restricting the exchange of intents to a maximum frequency and distance between agents, in proportion to the likelihood of interference with one another, without limiting communication of observations.

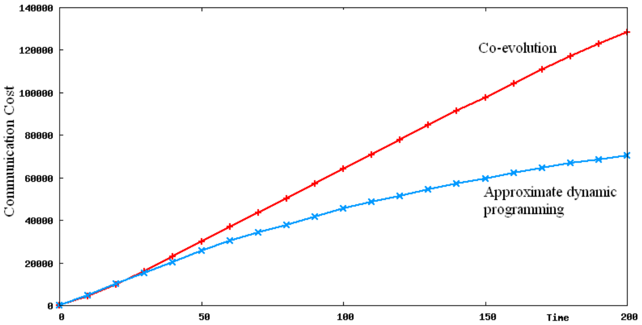


Figure 11. Communication cost over time: co-evolution [24] vs. approximate dynamic programming from [11].

The divergence-based information-sharing policy proposed in 6.1.1 has been compared to the policy used in [11]**,** which relies on fixed periodic exchanges and full cognitive map exchange when meeting team members. Both policies have been implemented in the co-evolutionary framework and simulated. Depicting similar performance (no significant statistical difference) in terms of entropy and number of targets discovered, both policies differ in terms of communication cost as shown in Figure 12 for a 5-UAV team scenario.

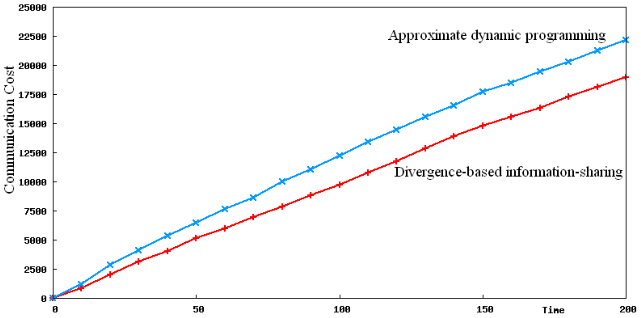


Figure 12. Communication cost over time: Divergence-based vs. Periodic/opportunistic [11] information-sharing policies.

Both information-sharing policies are equally efficient in timely providing needed information, but exchanging observations based on belief divergence turns out to be significantly cheaper (~ 20%). The relative savings are even higher when disregarding the cost of communicating intents as emphasized in [11]. The difference is due to two reasons. First, agents exchange observations only within each other's region of interest, which drastically reduces the number of observations communicated. Second, divergence-based information-sharing enables limited consensus through initial belief-sharing on mutually relevant regions. The required communication overhead is counterbalanced by the fact that only high-value observations (on cells with large belief divergence) are exchanged, resulting in a better use of the available communication bandwidth. As the policy proposed in [11] does not take advantage of potentially shared beliefs over cells or regions, it keeps consuming limited bandwidth on low-value observations.

6. SUMMARY AND CONCLUSION

A flexible discrete-event simulation environment has been proposed to explicitly investigate multi-agent resource-bounded coordination in the context of cooperative search path planning in a dynamic uncertain environment. It provides the user with a capability to configure a scenario from a variety of problem models, defining an operational mission environment, UAV agents, sensor observations, distributed planning and information-sharing algorithms, and communication and control action execution. Each model in turn can be parameterized in multiple dimensions. The environment further supports customized visualization and post-simulation analysis. Using the simulation platform, some comparative performance analysis for two solution designs was briefly discussed.

A few areas to be further explored include: Problem model extensions considering communication range as a new decision variable to tackle power management; multiple mission tasks to be carried out by a search-and-respond team; a multiplicity of problem dimensions (target and threat behaviors, UAV capabilities, communication routing, stochastic information-sharing policies, network awareness, large scale coordination, etc.).

References

[1] Benkoski, S., Monticino, M., and Weisinger, J. 1991. "A survey of the search Theory Literature", *Nav. Res. Log*, 38.

[2] Stone, L. 1989. "What's happened in search theory since the 1975 Lanchester Prize?", *Operations Research*, 37 (3).

[3] How, J. et al., 2009. "Increasing Autonomy of UAVs – Decentralized CSAT Mission Management Algorithm", *IEEE Robotics & Automation Magazine*, June 09, 43:51.

[4] Agmon, N. et al., 2008. "The giving tree: constructing trees for efficient offline and online multi-robot coverage", *Ann Math Artif. Intell.* 52:143–168.

[5] Wagner, W. et al., 1999. "Distributed covering by ant-robots using evaporating traces", *IEEE Trans. Robot. Autom.*, vol. 15, no. 5.

[6] Mathews, G., Durrant-Whyte H. 2006. "Scalable Decentralised Control for Multi-Platform Reconnaissance and Information Gathering Tasks", *9th International Conference on Information Fusion*, Florence, Italy.

[7] Jin, Y., Liao, Y., Minai, A., and Polycarpou, M. 2006. "Balancing search and target response in cooperative unmanned aerial vehicle (UAV) Teams", *IEEE Trans on Systems, Man and Cybernetics. Part B*, 36(3).

[8] Flint, F., Fernandez-Gaucherand, E., and Polycarpou, M. 2004. "Efficient Bayesian Methods for Updating and Storing Uncertain Search Information for UAV's", *Proc 43rd IEEE Conf on Decision and Control*.

[9] Beard, R.,. McLain, T., 2003. "Multiple UAV cooperative search under collision avoidance and limited range communication constraints", *Proc. of the IEEE Conference on Decision and Control*.

[10] Finn, A. et al., 2007. "Design Challenges for an Autonomous Cooperative of UAVs", In *Proc. of the International Conf on Information, Decision and Control*.

[11] Liao, Y., Jin, Y., Minai, A. and Polycarpou, M. 2005. "Information Sharing in Cooperative Unmanned Aerial Vehicle Teams", *Decision and Control.*

[12] Velagapudi, Prokopiev, Sycara, Scerri. "Maintaining Shared Belief in a Large Mutiagent Team", *10th International Conference on Information Fusion*, Quebec City, Canada.

[13] Sycara, K. et al.2007. "An Analysis and Design Methodology for Belief Sharing in Large Groups", *10th International Conference on Information Fusion*, Quebec City, Canada

[14] Intanagonwiwat, C. et al. 2003. "Directed diffusion for wireless sensor networking", *IEEE/ACM Trans. on Networking*, 11(1):2–16.

[15] Xuan, P., Lesser, V., Zilberstein, S. 2001. "Communication decisions in multi-agent cooperation: Model and experiments", in *Proc. of the 5th International Conference on Autonomous Agents*, Montreal, Canada.

[16] Brooks, A., Makarenko, A., Williams, S., Durrant-Whyte, H. 2006. "Parametric POMDPs for Planning in Continuous State Spaces", In *Robot. & Auton. Sys.*, 54(11).

[17] Garcia, R. and Barnes, L. 2010. "Multi-UAV Simulator Utilizing X-Plane", *Journal of Intelligent and Robotic Systems*, Springer, 57(1-4).

[18] Rasmussen, S.J. et al. 2005. "Introduction to the MultiUAV2 Simulation and Its Application to Cooperative Control Research", *American Control Conference*, Portland, OR, USA.

[19] Myeong-wuk, J. et al. 2003. "An Actor-based Simulation for Studying UAV Coordination", *15th European Simulation Symposium*, Delft, The Netherlands.

[20] MADKIT: Multi-Agent Development Kit 2002, http://www.madkit.org/

[21] Cover, T. and Thomas, J., 2006. "*Elements of Information-Theory*", 2nd edition, Wiley.

[22] Bernstein, D. et al. 2002. "The complexity of decentralized control of Markov decision processes", *Mathematics of Operations Research*, 27 (4), 819-840.

[23] Happe, J. 2009. "Implementation of Multi-Agent Cooperative Information Sharing and Planning", Tech Report RX-RP-52-8511, MDA, Richmond, B.C., Canada.

[24] Berger, J., Happe, J. and Lau, M. 2009. "Co-evolutionary Information Gathering for a Cooperative Unmanned Aerial Vehicle Team", *12th International Conference on Information Fusion*, Seattle, Washington.

[25] Yang, Y., Minai, A. and Polycarpou, M. 2002. "Decentralized Cooperative Search in UAV's Using Opportunistic Learning". In *Proc. AIAA Guidance, Navigation and Control Conference.*

Biography

Jens Happe works as an R&D Analyst for MacDonald, Dettwiler & Associates Ltd. (MDA) in Richmond, Canada. He holds a Ph.D. in computing science (2005), and a M.Sc. in mathematics (1995) from Simon Fraser University (SFU), Burnaby, B.C., Canada, as well as a German University Diploma in mathematics and control theory from Technische Universität Clausthal (1995). His research interests include computer-assisted Intelligence, Surveillance and Reconnaissance. He has designed and implemented toolbox architectures and simulation testbeds for Situation Analysis, Information Fusion and Resource Management.

Jean Berger is a defense scientist with the Decision Support Systems Section of Defense Research and Development Canada - Valcartier, working in the field of information technology. He received BS and MS degrees in engineering physics from École Polytechnique de Montréal, Canada. His research interests include artificial intelligence, operations research and simulation applied to intelligent control, planning, routing and scheduling problems.