World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:8, No:6, 2014

A New Multi-Target, Multi-Agent Search-and-Rescue Path Planning Approach

Jean Berger, Nassirou Lo, Martin Noel

***Abstract*—**Perfectly suited for natural or man-made emergency and disaster management situations such as flood, earthquakes, tornadoes, or tsunami, multi-target search path planning for a team of rescue agents is known to be computationally hard, and most techniques developed so far come short to successfully estimate optimality gap. A novel mixed-integer linear programming (MIP) formulation is proposed to optimally solve the multi-target multi-agent discrete search and rescue (SAR) path planning problem. Aimed at maximizing cumulative probability of successful target detection, it captures anticipated feedback information associated with possible observation outcomes resulting from projected path execution, while modeling agent discrete actions over all possible moving directions. Problem modeling further takes advantage of network representation to encompass decision variables, expedite compact constraint specification, and lead to substantial problem-solving speed-up. The proposed MIP approach uses CPLEX optimization machinery, efficiently computing near-optimal solutions for practical size problems, while giving a robust upper bound obtained from Lagrangean integrality constraint relaxation. Should eventually a target be positively detected during plan execution, a new problem instance would simply be reformulated from the current state, and then solved over the next decision cycle. A computational experiment shows the feasibility and the value of the proposed approach.

***Keywords*—**Search path planning, search and rescue, multi-agent, mixed-integer linear programming, optimization.

Open Science Index, Computer and Information Engineering Vol:8, No:6, 2014 waset.org/Publication/9998555

I. Introduction

SEARCH and rescue path planning is an increasingly important problem for a variety of civilian and military domains such as homeland security and emergency management. The basic discrete SAR or optimal searcher path problem involving a stationary target is known to be NP-Hard [1]. SAR may be generally characterized through multiple dimensions and attributes including: one-sided search in which targets are non-responsive toward searcher's actions, two-sided, describing target behavior diversity (cooperative, non-cooperative or anti-cooperative), stationary Vs. moving target search, discrete Vs. continuous time and space search (efforts indivisibility/divisibility), observation model, static/dynamic as well as open and closed -loop decision models, pursued objectives, target and searcher multiplicity and diversity. Early work on related search problems emerges from search theory [2], [3]. Search-theoretic approaches mostly relate to the effort (time spent per visit) allocation decision problem rather than path construction. Based upon a mathematical framework, efforts have increasingly been devoted to algorithmic contributions to handle more complex dynamic problem settings and variants [4], [5]-[7]. In counterpart, many contributions on search path planning may be found in the robotics literature in the area of robot motion planning [8] and namely, terrain acquisition [9], [10] and coverage path planning [11]-[13]. Robot motion planning explored search path planning, primarily providing constrained shortest path type solutions for coverage problem instances [14], [15]. These studies typically examine uncertain search environment problems with limited prior domain knowledge, involving unknown sparsely distributed static targets and obstacles. Recent taxonomies and comprehensive surveys on target search problems from search theory and artificial intelligence/distributed robotic control perspectives may be found in [16], [5], [17]-[19] respectively.

Exact problem-solving methods for sequential decision search problem formulations show computational complexity to scale exponentially. For instance, dynamic programming [5], [19], [7], [20] or tree –based search techniques [21], [22] may satisfactorily work under specific constraints and conditions but ultimately face the curse of dimensionality, showing poor scalability even for moderate size problem. This paved the way to the development of efficient heuristic and approximate methods. Some early approaches simply reduce computational complexity by relaxing some hard constraints to keep the problem manageable. Methods inspired from search theory propose procedures mainly based on branch and bound [20], [7] or path finding A* types of techniques and variants. Despite the development of many heuristics and approximate problem-solving techniques for the SAR problem [5], [19], published procedures still deliver approximate solution and mostly fail to provably estimate real performance optimality gap for practical size problems, questioning their real expected relative efficiency.

In this paper, we propose a new exact mixed-integer linear programming formulation to optimally solve the multi-agent discrete search path planning problem aimed at detecting multiple stationary objects. In the proposed open-loop with anticipated feedback problem model, 'open-loop with anticipated feedback' refers to offline planning, while capturing information resulting from predicted agent observations (projected cell visit action outcome) as opposed to real feedback. Anticipated feedback augments pure open-loop formulations which simply ignore information feedback,

Jean Berger is with Defence Research and Development Canada – Valcartier, Quebec City, Qc, Canada (phone: 418-844-4000 4645; fax: 418-844-4538; e-mail: jean.berger@drdc-rddc.gc.ca).

Nassirou Lo is with T-OptLogic Ltd., Quebec City, Qc, Canada (e-mail: nassirou.lo@t-optlogic.com).

M. Noel is with Téluq, Université du Québec, Quebec City, Qc, Canada (e-mail: noel.martin@teluq.uqam.ca).

while significantly improving solution quality, and mitigating computational complexity limitations traditionally associated with closed-loop problem formulations (e.g. dynamic programming, and partially observable Markov decision processes). This contribution aims at extending a single agent search path planning decision model [23] to a multi-target multi-agent environment in which feasible agent actions are further expanded to any possible neighboring move directions substantially increasing computational complexity, while capturing anticipated feedback information resulting from possible observation outcomes occurring from projected path execution. In that setting, the open-loop with anticipated feedback information (observations) decision model involves *n* agents (searchers) with imperfect sensing capability (but false alarm -free) searching an area (grid) to maximize cumulative probability of success in detecting *m* independent heterogeneous (dissimilar) targets, given a time horizon and prior cell occupancy probability distribution. The model takes advantage of anticipated feedback information resulting from observations outcomes along the path to update target occupancy beliefs and make better decisions. A network flow representation significantly reduces modeling complexity (e.g. constraint specification) as well as implementation and computational costs. The new decision model relies on an abstract network representation, coupled to a parallel computing capability (e.g. using the CPLEX solver [24]) to gain additional speed-up. The novelty lies in a new linear model, and the fast computation of near optimal solutions of practical size problems, providing a tight upper bound on solution quality through Lagrangean programming relaxation. The computable upper bound constitutes an objective measure to fairly estimate and compare performance gap against various techniques. Computational results prove the proposed approach very efficient. Small computational run-time naturally enables open-loop model (with anticipated feedback) extension to a closed-loop formulation in which action outcomes from the previous episode may be explicitly incorporated in real-time to update target occupancy belief distribution. As a result, an updated solution can be dynamically computed, by periodically solving new problem instances taking advantage of feedback information (from real observation outcomes), over short rolling horizons. The idea is to readily exploit episodic feedback information whenever available. In that case, computational run-time required to generate a solution corresponds to the duration of an episode. This way to embrace constructive dynamic planning in real time through inexpensive computational effort is largely preferable to dynamic programming techniques aimed at computing an exhaustive optimal policy, mapping suitable actions to any possible posterior states at a prohibitive computational cost. The proposed approach rather determines the best sequence of moves given the current state while updating the path solution resulting from partial path execution by repeatedly solving a new problem instance characterizing the follow-on state. Similarly, large time horizon problems can be solved efficiently, optimizing multiple problem instances over receding horizons.

The structure of the paper is organized as follows. Section II first introduces problem definition, describing the main characteristics of the open-loop search path planning problem with anticipated feedback. Then the main solution concept for the problem is presented in Section III. It describes a new mixed-integer linear programming network flow formulation combined with network representation to efficiently compute a near-optimal solution. The proposed CPLEX-based problem-solving technique and some implementation issues are then briefly presented in Section IV. Section V reports and discusses computational results depicting the value of the proposed method. Finally, a conclusion is given in Section VI.

II. Problem Definition

A. General Description

The discrete centralized search and rescue path planning problem involves a team of *n* homogeneous stand-off sensor agents searching *m* stationary and independent heterogeneous targets in a bounded environment over a given time horizon. From a search and rescue mission perspective, the goal consists in maximizing a weighted cumulative probability of success in detecting multiple heterogeneous targets within a given region. Weights capture predetermined relative target values. Represented through a grid, the search region characterizes an area defined as a set of cells *N*, describing possible target locations. Presumably occupying a single cell, the precise location of a given target is assumed unknown. A prior target location probability density distribution for which cell occupancy probabilities sum up to one can be derived from domain knowledge. It reflects possible individual cell occupancy, defining a grid cognitive map or uncertainty grid. A cell may be occupied by multiple targets. Should a target be located outside the search areas of interest, a special inaccessible, and invisible virtual cell would simply be added to the basic problem description depicted. A target cognitive map constitutes a knowledge base describing a particular world state, including variables such as target occupancy belief distribution, time, agent positions and orientations. An example of a cognitive map for a given target is illustrated in Fig. 1 at a specific point in time.

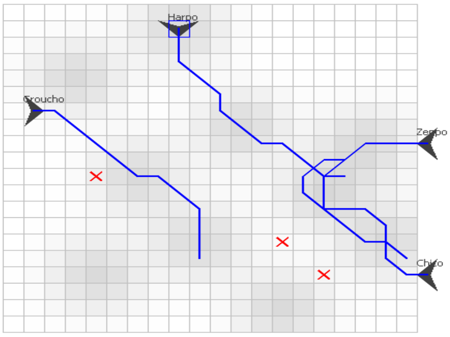

Fig. 1 Uncertainty grid /cognitive map at time step *t* for a given target. The 4-agent team beliefs are displayed through multi-level shaded cell areas. Projected agent plans are represented as possible paths

The duration of a cell visit or service time is assumed constant, specifying the period of each episode. Vehicles are assumed to visit different cell locations at the same time, and fly at slightly different altitudes to avoid colliding with one other. A search path solution consists in constructing an agent path plan selecting base-level control action to maximize target detection.

B. Agent Path Planning

Episodic agent search path planning decision is based on agent's position (cell location), specific orientation {*N, S, E, W, NE, SE, SW, NW*} and speed determining possible legal moves to adjacent cell locations. For example, the 3-move agent investigated in [23] is limited to three possible moving directions with respect to its current heading, namely, *ahead, right* or *left* as depicted in Fig. 2.

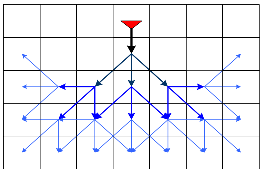

Fig. 2 Agent's region of interest displayed as forward move projection span (possible paths), for a 3-move agent over a 3–step time horizon

In this work, agent movement or manoeuvring capability is generalized to all degrees of freedom, permitting free motion along any possible directions to explore its neighborhood. An agent can therefore legally move toward its neighbouring cells offering eight alternate possible directions at each time step. This additional capability expands an agent path solution space by a factor $(8/3)^T$ over a 3-move planning agent for a given time horizon *T*, significantly increasing computational complexity.

The primary goal consists in planning base-level control action moves to maximize probability of success (target detection) over the entire grid.

C. Cumulative Probability of Success

In the proposed open-loop SAR model, the probability to successfully detect multiple targets resulting from *n* agent path solution executions on the grid is defined as the sum over cells of the product of the probability of detection reflected from cell visits and target cell occupancy belief dictated by cognitive maps (grids) [5], [25], [26]. A weighted cumulative probability of success (*CPOS*) for team path solutions (sequence of cell visits) over a time horizon *T* can then be expressed as follows:

$$CPOS = \sum_{c\in N}\sum_{t}\sum_{\tau} w_\tau pos_{ct\tau} = \sum_{c\in N}\sum_{t}\sum_{\tau} w_\tau p_{cc\tau}\, p_{ct\tau} \quad (1)$$

where $pos_{ct\tau}$ represents the probability of successfully detecting target τ while visiting cell *c* over period *t* given it has not been detected during earlier visits. w_τ reflects a user-defined weight capturing relative target value. $p_{ct\tau}$ refers to the probability/belief of cell occupancy by target τ during time interval *t* which incorporates "anticipated" information feedback that would result from past visits. As for $p_{cc\tau}$, it is a conditional probability on a specific agent visit to 'correctly' detect target τ in cell *c* given that τ is present in *c*. Target heterogeneity implies conditional probability of correct detection diversity. An agent sensor is assumed to be false-alarm free, meaning that a visit to a vacant cell induces a negative observation by the sensing agent. Conversely, based on this assumption, a positive observation confirms that a target is found. In the current setting, sensor range defining visibility or footprint (coverage of observable cells given the current sensor position) is limited to the cell being searched.

III. Mixed-Integer Linear Programming Model Formulation

A. Network Representation

A network representation is used to simplify modeling and constraint specification as well as problem-solving, as it eliminates the need to explicitly capture all constraints. These include maximum path length or deadline, admissible/legal move, and disconnected subtours elimination which may significantly impact run-time when handled explicitly.

Let $\mathcal{G}_k=(\mathcal{V}_k, \mathcal{A}_k)$be the grid network, a directed acyclic graph associated with agent $k\in \eta =\{1,...,n\}$, where $\mathcal{V}_k=\bigcup_{t\in T}\mathcal{V}_{kt}$ is the set of vertices associated to agent states (i.e. position and orientation state variables during a given episode $t \in T=\{0,1,2,..,|T|\text{-}1\}$), and $\mathcal{A}_k$ the set of arcs (i,j) where $i,j\in\mathcal{V}_k$, reflecting possible agent state transition between consecutive episodes over the grid, corresponding to a legal move *m* selected from the action set A={*left, ahead, right*}. $N_{kt} = N$ is the set of possible cell locations $\{1,...,|N|\}$ over the grid during episode *t* whereas $O_{kt} = O$ refers to the set of possible agent orientations/headings $\{E,NE,N,NW,W,SW,S,SE\}$ during episode *t*. As a result, $\mathcal{V}_k=\bigcup_{t\in T}\mathcal{V}_{kt} = \bigcup_{t\in T}(N_{kt}\times O_{kt})$. The nodes *o* and *d*are additional fictitious origin and destination location vertices defining legal path ends in graph. An excerpt from the abstracted representation for the agent network over two consecutive episodes is given in Fig. 3. An integer binary flow decision variable x_{ijk} is associated to each arc $(i,j)\in\mathcal{A}_k$. Agent *k* path solution include arcs $(i,j)\in\mathcal{A}_k$ for which $x_{ijk} = 1$. Given an initial agent state $i_0(k)$, path may be defined over the grid network traveling along arcs connecting *o* to *d* instantiating flow decision variables to build feasible paths and then, consequently, assigning visit decision variables involved in the objective function. Agent state vertex duplication over |*T*| episodes is aimed at eliminating disjoint solution subtours otherwise difficult to handle explicitly, and provides a directed acyclic graph to represent a legal solution through binary integer flow decision variables including a multi-cycle path (possible occurrence of many visits on the same cell). Duplication implicitly satisfies path length constraint as well. The significant gain obtained through duplication clearly exceeds the cost incurred by slightly degraded model readability due to the utilization of more complex notations. The agent network includes |O| |*N*| |*T*| nodes and |O| |*N*| |*T*| |A| arcs. It is assumed that a cell *c* can be visited at most V_c times.

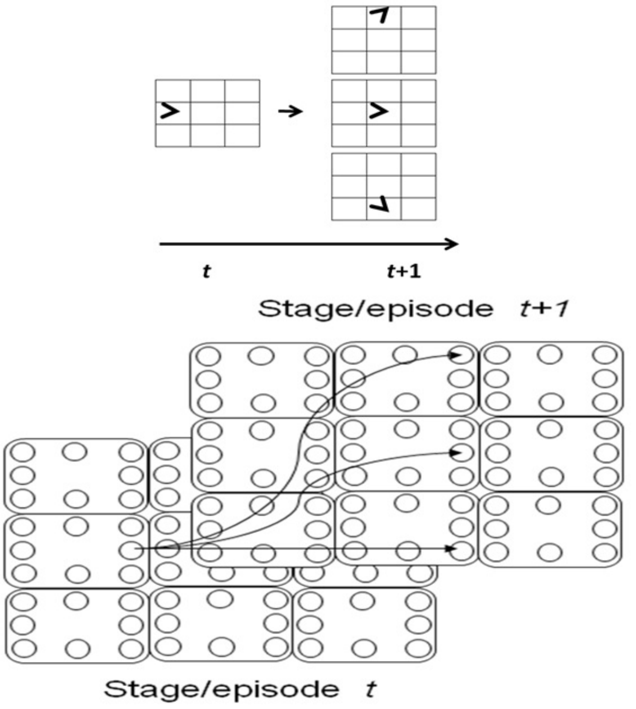

Fig. 3 Agent grid network (directed acyclic graph) excerpt, over consecutive episodes *t* and *t*+1 for a 3x3 -cell grid. Nodes depict agent state (position, orientation, episode) whereas arcs capture node transition between episodes defined by possible legal moves. Squares refer to grid cells enclosing 8 possible agent orientations. A |*T*|-move path may be constructed by moving along arcs from stage 0 to stage |*T*|-1

B. Mathematical Modeling

A mathematical mixed-integer linear programming (MIP) formulation is proposed for the discrete stationary multi-target search and rescue (SAR) path planning problem. It extends the single agent model [23] to a multi-agent multi-target setting while incorporating any possible agent action moves.

The open-loop decision model captures explicitly ahead of time anticipated information feedback resulting from projected action execution to update target cell occupancy probability (belief). Occupancy belief are assumed to be independent of past search visits as reported for the optimal searcher path problem [5], [25]-[28]. Accordingly, based on the completion of a projected visit in cell *c* during time interval *t*, the posterior probability of cell containment for a given target $\tau p_{c'\ t+1\tau}$ for any cell *c'* is related to its prior belief $p_{c'\,t\tau}$ by:

$$p_{c't+1\tau} = \left(1 - p_{cc\tau}\delta_{cc'}\right)p_{c'\,t\,\tau} \qquad (2)$$

where $\delta_{cc'}$= 1 if *c'*=*c* and 0 otherwise. $p_{c't\tau}$ refers to the probability/belief of cell *c'* occupancy by target τ during time interval *t* which incorporates "anticipated" information feedback that would result from past visits.

It should be noted that even if probability update normalization does not change relative probability over cells *c* for a given path, it nonetheless introduces nonlinearity making problem-solving much harder. However, it is assumed that normalization alone does not generally induce substantial differences to significantly impact the quality of the final path solution for this problem. Therefore, similarly to [5], [25]-[28], we do deliberately ignore normalization in updating occupancy beliefs, proposing a near-optimal solution to an approximate problem model.

The variables and parameters defining the decision model are given as follows:

η : set of homogeneous agents {1,2,…,*n*}

N: set of cells defining the grid search area {1,2,..,|*N*|}

T: set of time intervals defining the time horizon {0,1,...,|*T*|-1}

T: set of targets {1,2,…,*m*}

w_τ: relative value given to target $\tau(\sum_{\tau\in\mathrm{T}} w_\tau = 1)$

V_c: maximum number of visits on cell *c*

$p_{cc\tau}$: conditional probability of 'correct' target τ detection on a visit in cell *c* given that the target is located in *c*.

$\beta_{c\tau}$: 1/(1-$p_{cc\tau}$)

$p_{ct\tau}$: belief of cell *c* occupancy by target τ during time interval *t*. {p_{c0}} refers to the initial belief distribution of target occupancy over the grid.

$pos_{ct\tau}$: probability of success (finding the target τ) resulting from the observation of cell *c* at the end of time interval *t*

CPOS: objective function defining weighted cumulative probability of success.

v_{clt}: binary decision variable corresponding to cumulative number of visits *l* on cell *c* at the end of time interval *t*– v_{clt}=1 (otherwise 0)

y_{ct}: binary decision variable reflecting agent position in episode *t*. It indicates that cell *c*is visited during time interval *t* - $y_{c\,t}$ =1 (otherwise 0)

x_{ijk}: state transition binary variable. x_{ijk} = 1 reflects agent *k* network state transition from state *i* to *j* between consecutive episodes. Agent *k* path solution includes arcs (*i*,*j*) ∈ $\mathscr{A}_k$ for which x_{ijk} = 1

The MIP decision model may be formulated as follows:

$$\max CPOS = \max_{\{pos_{ct\tau}\}} \sum_{c\in N}\sum_{t\in T}\sum_{\tau\in\mathrm{T}} w_\tau\; pos_{ct\tau} \qquad (3)$$

Subject to the linear convex constraint set:
Cell visits:

$$\sum_{0\le l\le V_c} v_{clt} = 1 \qquad \forall c\in N, \forall t\in T \qquad (4)$$

$$\sum_{0\le l\le V_c} l v_{clt} = \sum_{0\le t'\le t} y_{ct'} \qquad \forall c\in N, \forall t\in T \tag{5}$$

Belief update:

$$p_{c\,t+1\,\tau} = \sum_{0\le l\le V_c} \frac{p_{c0\tau}}{\beta^{l}_{c\tau}} v_{clt} \quad \forall c\in N, \forall t\in T, \tau\in \mathrm{T} \tag{6}$$

Probability of success:

$$pos_{ct\tau} - p_{cc\tau} p_{ct\tau} \le 1 - y_{ct} \quad \forall c\in N, \forall t\in T, \tau\in \mathrm{T} \tag{7}$$

$$pos_{ct\tau} \le y_{ct} \quad \forall c\in N, \forall t\in T, \tau\in \mathrm{T} \tag{8}$$

Initial probability:

$$p_{c0\tau} = p_{c\tau}(t=0) \quad \forall c\in N, \tau\in \mathrm{T} \tag{9}$$

Network coupling:

$$y_{ct} = \sum_{k\in\eta} \sum_{i_t(c)\in \mathcal{V}_k} \sum_{j_{t+1}\in \mathcal{V}_k} x_{i_t(c)\, j_{t+1}\, k} \quad c\in N, t\in T, \qquad (i_t(c), j_{t+1})\in \mathcal{A}_k \tag{10}$$

Initial agent position:

$$x_{o\, i_0(k)\, k} = 1 \qquad \forall k\in\eta, i_0(k)\in \mathcal{V}_k \tag{11}$$

$$y_{c0} = \sum_{k\in\eta} \delta_{c\, y_0(k)} \quad \forall c\in N, \forall k\in\eta \tag{12}$$

Initial/final path condition:

$$\sum_{i\in\mathcal{V}_k} x_{oik} = 1 \qquad \forall k\in\eta \tag{13}$$

$$\sum_{i\in\mathcal{V}_k} x_{idk} = 1 \qquad \forall k\in\eta \tag{14}$$

Flow conservation:

$$\sum_{i\in\mathcal{V}_k \cup\{o\}} x_{ijk} - \sum_{i\in\mathcal{V}_k \cup\{d\}} x_{jik} = 0 \quad \forall k\in\eta, \forall j\in \mathcal{V}_k, (i,j)\in \mathcal{A}_k \tag{15}$$

Maximum path length:

$$\sum_{i\in\mathcal{V}_k} \sum_{j\in\mathcal{V}_k/\{i\}} x_{ijk} = T \qquad \forall k\in\eta, (i,j)\in \mathcal{A}_k \tag{16}$$

Decision variables

$$pos_{ct\tau}, p_{ct\tau} \in [0,1] \quad c\in N, t\in T, \tau\in \mathrm{T}$$
$$y_{ct}, v_{clt} \in \{0,1\} \qquad c\in N, t\in T, \forall l\in\{0, V_c\}$$
$$x_{ijk} \in \{0,1\} \quad \forall k\in\eta, (i,j)\in \mathcal{A}_k \tag{17}$$

The objective function shown in (3) defines weighted cumulative probability of success over an agent path solution and time horizon $|T|$. Constraints are governed through (4)-(17). For a given path solution, constraints (4) represent the cumulative number of visits paid on site c by the end of time interval t. Constraints (5) simply link that number to past visits on c so far. It should be noticed that simultaneous visits by multiple agents on a specific cell over a given time interval is implicitly prevented and reinforced by the fact that $y_{ct} \le 1$, limiting to at most one, the number of visits a cell can receive during an episode. For cell coverage purposes, we assume a maximum number of visits V_c to be performed on site c. The bound V_c can be pre-computed or selected arbitrarily large. Target occupancy probability update is governed by constraint set (6). It is the explicit form of (2) relating belief and number of conducted visits. Constraint sets (7) and (8) determine probability of success contributions. Both in equations mutually reflect a visit requirement to a cell to ensure a feasible observation and an admissible success contribution aligned with the objective function. Initial probability distributions are specified in (9). Constraint sets (10)-(16) reflect model and network coupling as well as flow constraints imposed on/by the agent network. Constraints (10) link cell visits to the agent path network, connecting outgoing arcs from network nodes (states) on stage t to the cell c being visited during episode t. Accordingly, arcs $(i_t(c),j_{t+1})$ relate to any agent state transition starting from position $c$at stage t. Agent k initial state $i_0(k)$ and position $y_0(k)$ as well as its related network connection are captured in constraints (11)-(12). Constraints (13)-(14) guarantee path solution departure and final arrival points to be uniquely defined. Flow conservation governed by constraints (15) aims at balancing the number of incoming and outgoing arcs respectively for a given node. Constraints (16) guarantee a$|T|$-move path solution for an agent, but turn out to be unnecessary as solution constraints are implicitly satisfied by agent network construction. Binary and continuous domain variables are then defined in (17).

Should eventually a targetbe positively detected (found target) during plan execution, a new problem instance would simply be reformulated from the current state. It would consist in taking away the detected target in the revisited model and then continue on solving the new problem instance over the next decision cycle.

C. Single Team Network Simplification

Given agent homogeneity, a single 'team' (n agent)T-stage network $\mathcal{G}$=($\mathcal{V}$,$\mathcal{A}$) representing possible team paths may alternatively be used, requiring minor network adjustments to concurrently incorporate agent action multiplicity subject to non-simultaneous visits on a same cell. The resort to a single team network rather than multiple network-agent mapping provides additional speed-up, number of decision variable reduction and significant computer savings (by a factor n). The resulting team directed acyclic graph $\mathcal{G}$=($\mathcal{V}$,$\mathcal{A}$) captures agent multiplicity substituting x_{ijk} integer flow decision variables for x_{ij}, slightly modifying some key flow constraints:

$$\sum_{i\in\mathbb{V}} x_{ij} - \sum_{l=0}^{V_c} l v_{cl} = 0 \quad \forall c \in N, (i, j(c)) \in \mathcal{A}$$

$$x_{oi} = \sum\nolimits_{k\in\eta} \delta_{i\,i_0(k)} \; \forall i \in \mathbb{V}$$

$$\sum\nolimits_{i\in\mathbb{V}} x_{oi} = n \,, \sum\nolimits_{i\in\mathbb{V}} x_{id} = n$$

$$\sum_{i\in\mathbb{V}\bigcup\{o\}} x_{ij} - \sum_{i\in\mathbb{V}\bigcup\{d\}} x_{ji} = 0 \quad \forall j \in \mathbb{V}, (i,j) \in \mathcal{A}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in \mathcal{A}$$

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

The expected computational gain comes at the low cost expense of reconstructing individual agent paths from the computed agent-free decision variables of the team network solution. The agent path reconstruction procedure is described next.

1) Agent Path Reconstruction

A particular agent path is reconstructed using the team network and its instantiated integer flow decision variables x_{uv}. A legal T-move agent k path is simply generated by moving along the computed team solution arcs from its departure state node $i_0(k)$(combining initial cell and orientation) in stage 1 adding the related cell to the evolving path, up to stage T, before finally converging to the destination node d. Decision variables are progressively decremented as the path expands. The agent path reconstruction algorithm is straightforward and fast ($O(nT)$), as summarized below:

```
For  $k = 1..n$  do   -- cycle over agents
   $u = i_0(k);\ path_k = \phi;\ t = 1$
   While $(t \le T)$
      select state transition $(u,v) \in \mathcal{A}$ such that $x_{uv} > 0$
      $path_k.cell(t) = cell_u$
      $t = t + 1$
      $x_{uv} = x_{uv} - 1,\ u = v$
   end While    -- T - move path constructi on
end For         -- agent$_k$ path solution
```

The path solution $path_k$ in the above procedure is composed of a sequence of T cell visits. The path element $path_k.cell(t)$ refers to the specific cell ($cell_u$) visited by agent k in period t.

D. Dynamic Planning and Time Horizon

Dynamic problem solution can be computed constructively over receding horizons by repeatedly exploiting real information feedback as it becomes available and a new optimization to progressively improve solution quality. Aside the explicit inclusion of real information feedback, large time horizon problems are similarly solved through repeated fast subproblem optimizations over receding horizons as pictured in Fig. 4. Time horizon is divided in time intervals and corresponding subproblems sequentially solved over respective episodes of period ΔT. Accordingly, a subproblem solution periodically expands the overall current partial path solution progressively incorporating a small fraction of its solution moves (subperiod δT), while updating the objective function with new path contributions. Limited move insertions define overlapping episodes, mitigating the effects of myopic path planning. A new subproblem is then periodically solved subject to the revisited objective function updated from the previous episode accounting for the partial solution being progressively built. The process is then reiterated until the time horizon has been covered. The strategy consists in taking advantage of the fast computation of reasonable time horizon subproblems over a limited number of episodes to quickly compute a near optimal solution to the original problem.

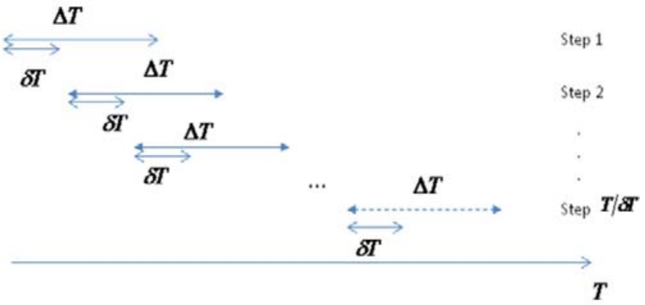

Fig. 4 A large time horizon T is defined over T/δT receding horizons of period ΔT. Moves computed in subperiods δ T form the final path solution to the original problem

E. Discussion

The proposed formulation confers many advantages over alternate modeling procedures, as the linear model allows to efficiently compute a bound on the optimal solution quality through Lagrangean programming relaxation. This provides a comparative measure to carry out performance gap analysis over alternate solutions, as well as the ability to trade-off solution quality and run-time for heuristic methods operating under tight temporal constraints. Problem-solving may be naturally achieved using well-known efficient techniques.

IV. MIP Algorithm - CPLEX solver

The IBM ILOG CPLEX parallel Optimizer version 12.2.0.0 [24] was used, essentially exploiting various optimized problem-solving techniques for large size problems. CPLEX solves the (exact) mixed integer programming (MIP) problem model implicitly computing an upper bound on solution quality through integrality constraint relaxation referred as Lagrangean programming relaxation (LP).

Additional speed-up can be contemplated for implementation efficiency purposes. Accordingly, as probability update for a given cell is assumed strictly dependent upon the number of local visits, the proposed problem model may be solved resorting to an alternate equivalent implementation inspired from [29], exploiting objective contributions pre-calculation in maximizing a linear function over binary integer variables v_{cl} (number of visits l on cell c):

$$\max_{\{v_{cl}\}} \sum_{c\in N}\sum_{l} c_{cl} v_{cl}, \quad c_{cl} = \sum_{\tau} w_\tau p_{c0\tau}\left(1-\left(1-p_{cc\tau}\right)^{l}\right)$$

As a result, the implementation approach significantly reduces search space complexity reporting substantial gain in run-time.

V. Computational Experiment

A computational experiment has been conducted to test the approach for a variety of scenarios. The value of the proposed MIP approach is assessed in terms of optimality gap and run-time. Computed solutions are reported against the relative cumulative probability of success optimality gap shown at the end of horizon $|T|$:

$$Opt\ gap = \frac{CPOS^{*} - CPOS_a}{CPOS^{*}} \qquad (18)$$

where $CPOS^*$ is the optimal cumulative probability of success defined in (1) or a tight upper bound (LP solution), and $CPOS_a$ the performance of our approach for a given scenario. The closer (smaller) the optimality gap the better the performance.

A. Simulations

Computer simulations were conducted under the following conditions:

- Number of targets m, ranging over {1,2,3,4,5} and respective prior target cell occupancy belief distributions; identical value/weight: $w_\tau = 1/m$
- Grid size N = 15x15, 10x10
- Homogeneous sensor agents:
  Team cardinality: n ranging over {1,2,3,4}
  Actions: 8 moves
  V_c=5 for all cells c
  Sensor parameters: p_c running into [0.7,0.95] for all cells
- Hardware Platform:
  Intel (R) Xeon (R) CPU X5670
  Shared-memory multi-processing: 8 processors, 2.93 GHz
  Random Access Memory: 16 Go, 64 bits binary representation (double precision)

It should be noted that as target cell occupancy probability sum up to one, performance analysis for large grid turns out to be less attractive. Accordingly, the larger the grid in general, the smaller (arbitrarily negligible) the related target cell occupancy belief, inevitably conducting either to significant visit payoffs for a limited number of prominently noticeable cells sparsely distributed over a large area, or alternatively in near similar cell visit rewards, for which any sub-optimal algorithms would likely demonstrate highly competitive (near similar) performance behavior. In both cases, this would result in a large and costly fraction of the total effort and time dedicated to the planning and construction of long and unimportant subpath segments, leading ultimately to marginal or insignificant gains. Consequently, grid instances larger than 10x10 – 15x15 should be further downsized and aggregated to embrace minimal belief coverage, to ensure substantial analysis and solution performance evaluation. This is why this study limited its investigation to the exploration of grid instances with a 15x15 maximum in size.

B. Results

A sample of random simulation results is reported in Table I for a few 15x15 grid 8-move multi-agent scenarios over horizon T. Each entry corresponds to a separate problem instance. Target multiplicity, agent team cardinality and time horizon are specified in second and third column respectively. A sample of performance results in terms of cumulative probability of success (*CPOS*) and optimality gap using the optimal CPLEX MIP solver is reported in the fourth column for path solutions subject to converge within a 2% optimality gap threshold. Run-time expressed in seconds is depicted in the last column. Data sets are organized in 5 blocks. Blocks 1-4 include instances 1-11 corresponding to 4-target and 2-8 – agent scenarios grouped by team cardinality, whereas block 5 refers to 12-target and 4-6 -agent data sets defining instances 12-14.

TABLE I
Performance of CPLEX Solver (MIP) for a Sample of Multi-Target, 8-Move 2-8–Agent Data set (15x15 grid)

Instance	***m*** **Targets** ***m***	***n*** **Agents** ***n***	**Time Horizon |*T*|**	**CPLEX Solver - MIP** ***CPOS***	***Opt gap %***	**CPLEX Solver Run-time (*s*)**
^{1}m4n2T20	4	2	20	0.5499	1.95	55
^{2}m4n2T22	4	2	22	0.5649	0.56	63
^{3}m4n4T18^{A}	4	4	18	0.6479	1.78	59
^{4}m4n4T16^{B}	4	4	16	0.6225	1.35	21
^{5}m4n4T16^{C}	4	4	16	0.6238	1.74	12
^{6}m4n4T16^{D}	4	4	16	0.6235	0.56	11
^{7}m4n4T16^{E}	4	4	16	0.6279	0.64	26
^{8}m4n4T18^{F}	4	4	18	0.6427	1.23	26
^{9}m4n6T15	4	6	15	0.6909	1.03	33
^{10}m4n6T17	4	6	17	0.7186	1.30	55
^{11}m4n8T14	4	8	14	0.7399	1.01	47
^{12}m12n4T18	12	4	18	0.6107	0.17	94
^{13}m12n6T17	12	5	17	0.6527	0	114
^{14}m12n8T14	12	6	14	0.6533	0	27

Computational results show that near-optimal solutions for 15x15 grid instances can be computed approximately on a minute timescale, imposing an optimality gap within less than 2%. These findings prove computation of tight upper bounds on solution quality to be quite feasible and realistic in practice. Bound tightness suggests that relaxed solution might be further exploited while decreasing optimality gap threshold to improve solution quality and run-time performance. Near optimal multi-agent solution may in counterpart be computed on a second timescale for 10x10 grids. Generally, the larger the time horizon the larger the run-time, as solution space increases exponentially with T. However, we might sometime observe marginal gain in performance as indicated for n=2,4,6 highlighting the value of considering a limited time horizon outweighing potential benefits anticipated from longer term projection, in particular when path plans can be fully revisited after each visit/episode. In other words, even though larger time horizon for the studied scenarios might turn out to be computationally prohibitive, possibly demanding several minutes to ensure solution optimality convergence, specified time horizons in Table I are sufficient to dynamically build a path plan one step at a time, as the grid remains entirely visible to the planner during planning. The largest 8-move multi-agent problem investigated still delivering near-optimal solution within an approximately 1-minute run-time, involves 8 agents over a 14 unit -time horizon, as exemplified by instance 11. In that case, the reported path solutions cover a significant portion of interesting cells as illustrated by *CPOS* performance results. In effect, the total number of moves nT to be executed by the team already corresponds to half grid size N/2, and therefore leads to high quality path solutions. Upgrading computational power technology through faster hardware and augmented parallel processing might ultimately extend computable T.

It is worth noticing that run-time is not only impacted by combined team cardinality/planning time horizon Nt (path solution space: 8^{nT}) and relative initial agent positions but also by target multiplicity creating additional visit contention among agents, and therefore increasing problem-solving complexity. In effect, target multiplicity generally tends to naturally reduce objective function contribution gaps among grid cells, making more cells equally attractive, and ultimately resulting in higher path solution quality contention. Agent contention is further exacerbated by team cardinality (e.g. n=8), in which more agents are exposed to a larger number of competing candidate cell visits, increasing path plan combinatorics. This is shown in Table I for data sets 12-14 in which run-time for 12-target instances may reach up to 2 minutes. Differential run-time performance between data sets 12 and 13 is due to additional contention induced by team agent proximity when starting the search. It is observed that time horizon larger than grid dimension ($N^{1/2}$) makes problem increasingly complex with target multiplicity. 12-target problem scenarios could be solved within 2 minutes for a team involving up to 6 agents. As mentioned earlier, bound tightness characterizing Lagrangean relaxation path solution might be further exploited (local search) to speed-up problem-solving and possibly handle larger agent teams.

A separate limited experiment alternatively confirms that weight parameters, which translate relative target importance in the objective function (3), mainly operate as agent attractors. They represent a main driver in magnifying high-payoff locations (attractors), expectedly imposing a strong bias in directing more efforts (paid visits) toward most valued targets in priority, whenever necessary.

Simulation results prove the proposed approach very efficient. Providing best or near optimal solution and a comparative measure of efficiency (upper bound obtained from Lagrangean integrality constraint relaxation) for practical size problems, the approach could be repeatedly reused in dynamic settings exploiting intermediate sensor readings, given its small run-time.

A typical path solution is shown in Fig. 5. It should be mentioned that cells presenting large beliefs may sometime be visited less frequently than others depicting smaller belief magnitudes. The reason either relates on the gap between conditional probability of detection characterizing some targets which decrease marginal return differently after a visit, or when cells bridge promising areas, as illustrated in Fig. 5 whereby agent 4 visit cell ‘0.23’ (at the bottom) only once, whereas cell ‘0.2’ is alternatively visited by agent 2 and 3.

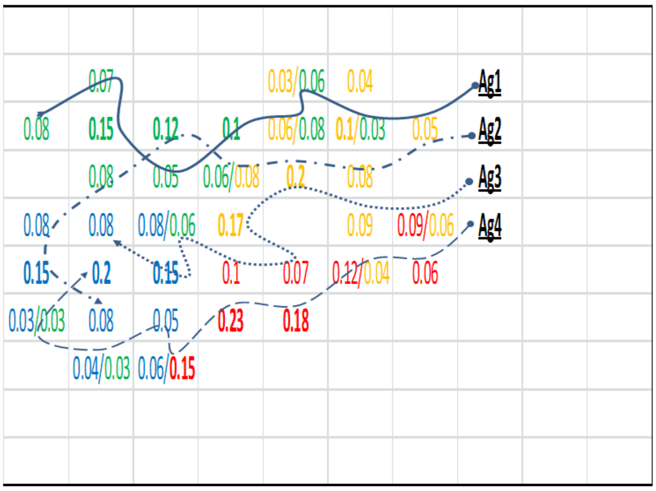

Fig. 5 4-agent team path solutions for 4 superimposed target belief distributions in a 10x10 grid, for T=10

Based upon computational results, an analysis shows that search team behavior is ultimately conditioned by the multi-target belief landscape. Initial peaks and valleys magnitudes and spatial distributions determine a full spectrum of agent path solution configurations, from a complex web of entangled agent trajectories overlapping one another (e.g. a near flat landscape), toward a spatially partitioned agent route pattern resulting from a highly well-structured and contrasted belief landscape with few distinct and distant pikes. Initial belief magnitude distribution is primarily determined by the variance over probability of cell containment for every target. As probability sums up to one, high-value beliefs commonly indicate a low occurrence number of peaks (high variance) asking for fewer searching agents and visits, whereas quasi similar (flat) beliefs (low variance) corresponds to a small number of high payoff locations, therefore requiring contributions from many agents and more visits to cover target belief distribution, while making path construction computationally more challenging. In other respect, belief spatial distribution has shown to significantly influence team behavior and individual agent path solutions. In this regard, large magnitude beliefs in conjunction with a small number of aggregated peaks for all targets, combined to mutually distant targets, eventually occurring at some stage of the search, induces a propensity to partition targets between team members. Accordingly, multi-target beliefs distributed in localized clusters per target and distant from one another as graphically exemplified in Fig. 6(for a 5x5 grid for the sake of the argument) are quite evocative in naturally delivering path solutions partitioning the overall grid, splitting the team one agent per target. This partition remains nonetheless modulated by the various conditional probability of detection characterizing targets and locations as well as belief distribution variance (for a given target) which may incur some final path configuration variability. Incidentally, cells depicting high conditional probability of detection are likely to be more attractive to agents as they naturally tend to further increase the objective function. However, in the general case, as separate target belief clusters either get closer and overlap with one another or progressively widen and expand across the grid while decreasing in magnitude, results show that agent paths rather share a mix of sites from different targets indiscriminately, fully devoted at maximizing the objective function or average coverage.

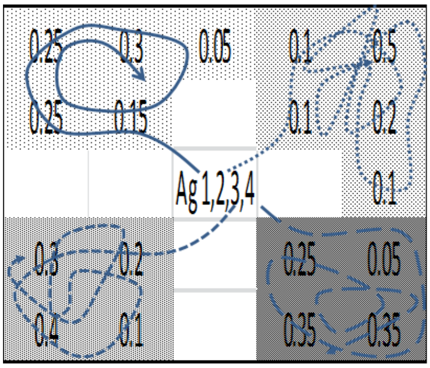

Fig. 6 4-agent team path solutions for 4 non-overlapping target clustered belief distributions illustrated by different shades/patterns on a 5x5 grid. The belief space is optimally partitioned, assigning one agent per target.

VI. Conclusion

An innovative mixed-integer linear programming (MIP) approach has been proposed to solve a probabilistic open-loop multi-target multi-agent search and rescue path planning problem with anticipated feedback, in which agent actions are subject to any neighbouring move directions. The novelty of the approach lies in a revisited combination of an extended problem formulation, an original network representation, and a refined problem-solving procedure based on linear programming CPLEX technology to efficiently compute near-optimal solution for practical size problems, usually handled through heuristic methods. For the first time, an upper bound estimate on the optimal solution naturally derived from the approach may be used for convergence or performance comparison analysis purposes, and/or trading-off solution quality and execution time. Experimental results demonstrate the value of the proposed approach, proving problem-solving to be feasible in reasonable time. Small computational cost naturally allows dynamic planning through a closed-loop environment settings where real information feedback resulting from past sensor agent observations is exploited to compute a revisited solution over a rolling horizon.

Future research directions will consist in considering generalized sensor footprint, and increasingly complex observation models (e.g. false-alarm) while extending search to moving targets. Alternate research work will explore search problem modeling variants involving heterogeneous sensing agents.

References

[1] K.E.Trummel and J.RWeisinger, “The complexity of the optimal searcher path problem”,*Operations Research*, 34 (2), 1986, pp.324-327.

[2] S. Benkoski, M. Monticino, and J. Weisinger, “A survey of the search theory literature”, *Naval Research Logistics* 38, 1991, 469–494.

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

[4] R. Hohzaki, and K. Iida, “Optimal search plan for a moving target when a search path is given”, *Math. Japonica*, 41(1), 1995, pp. 175-184.

[5] H. Lau, *Optimal Search in Structured Environments*, PhD Thesis, University of Technology, Sydney 2007.

[6] H. Lau, and G. Dissanayake, “Probabilistic search for a moving target in an indoor environment”, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2006, pp. 3393-3398.

[7] H. Lau, and G. Dissanayake, “Optimal search for multiple targets in a built environment”, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Edmonton, Alberta, Canada, 2005, pp. 228-233.

[8] J.-C. Latombe, Robot Motion Planning, ser. *International Series in Engineering and Computer Science; Robotics: Vision, Manipulation and Sensors*, vol. 124, Boston, Kluwer Academic Publishers, 1991.

[9] C. Choo, J. Smith, and N. Nasrabadi, “An efficient terrain acquisition algorithm for a mobile robot”, in *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, 1991, pp. 306–311.

[10] A. Sankaranarayanan, and I. Masuda, “Sensor based terrain acquisition: A new, hierarchical algorithm and a basic theory”, in *Proc. IEEE/RSJInt. Conf. Intelligent Robots and Systems*, Raleigh, 1992, pp. 1515–1523.

[11] J. Svennebring, and S. Koenig, “Building terrain-covering ant robots: A feasibility study”, *Auton. Robots*, 16 (3), 2004, pp. 313–332.

[12] S. Wong, and B. MacDonald, “A topological coverage algorithm for mobile robots”, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, 2003, pp. 1685–1690.

[13] S. Yang, and C. Luo, “A neural network approach to complete coverage path planning”, *IEEE Trans. Syst.,Man, Cybern. B, Cybern.*, 34(1), 2004, pp. 718–724.

[14] I. Rekleitis, et al., “Efficient boustrophedon multi-robot coverage:an algorithmic approach”, *Ann Math Artif. Intell.* 52, 2008, pp. 109–142.

[15] W. Wagner, et al., “Distributed covering by ant-robots using evaporating traces”, *IEEE Trans. Robot. Autom.*, 15(5), 1999,pp. 918-933.

[16] J.O. Royset and H. Sato, “Route Optimization for Multiple Searchers”, *Naval Research Logistics*, 57 (8), 2010, pp. 701-717.

[17] T. H. Chung, J. Burdick, “Analysis of search decision making using probabilistic search strategies”, *IEEE Transactions on Robotics*, 28 (10), 2012, pp. 132-144.

[18] Y. Jin, Y. Liao, A. Minai, and M. Polycarpou, “Balancing search and target response in cooperative unmanned aerial vehicle (UAV) Teams”, *IEEE Trans on Sys Man and Cybern. Part B,* 36(3), 2006, pp. 571-587.

[19] G.A. Hollinger, *Search in the Physical World.* CMU-RI-TR-10-20, Robotics Institute, PhD thesis, Carnegie Mellon University, 2010.

[20] T. Chung, “On Probabilistic Search Decisions under Searcher Motion Constraints”, *Workshop on Algorithmic Foundation of Robotics VIII,* Guanajuato, Mexico. 2009, 501-516.

[21] G.A. Hollinger, and S. Singh, “GSST: Anytime Guaranteed Search with Spanning Trees”, *Autonomous Robots*, 29(1), 2010, pp. 99-118.

[22] A. R.Washburn, “Branch and Bound Methods for a Search Problem”, *Naval Research Logistics*,45, 1998, 243-257.

[23] J. Berger, N. Lo, and M. Noel,Exact Solution for Search-and-Rescue Path Planning, Proceedings of the *2nd International Conference on Information Computer Applications*, Rome, Italy, February 2013.

[24] IBMILOG CPLEX Optimizer, Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/cplex-optimizer/cplex-performance/ and IBM ILOG CPLEX

V12.1, 2009. Available: ftp://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/ps_usrmancplex.pdf

[25] M. Morin et al. “The Optimal Searcher Path problem with a Visibility Criterion in Discrete Time and Space”, *Int. conference on Information Fusion*, Seatle, USA, 2009.

[26] M. Morin, A.P. Papillon, F. Laviolette, I. Abi-Zeid, and C.G. Quimper, “Constraint Programming for Path Planning with Uncertainty: Solving the Optimal Search Path problem,” in *Proceedings of the 18th Conference on Principles and Practice of Constraint Programming*, Québec, Qc, Canada, 2012, pp. 988-1003.

[27] J. N Eagle., and J. R Yee., “An optimal branch and bound procedure for the constrained path, moving target search problem”, *Operations Research*, vol. 38, no. 1, 1990, pp. 110-114.

[28] R. F Dell, J. N Eagle, G. H. Martins and A. G. Santos, “Using multiple searchers in constrained-path, moving-target search problems”, *Naval Research Logistics*, vol. 43, 1996, 463-480.

[29] N. Lo, J. Bergerand M. Noel, Toward Optimizing Static Target Search Path Planning, *Proceedings of the 5th IEEE Symposium on Computational Intelligence for Security and Defence Applications*, Ottawa, Canada, July 2012.