Journal of Combinatorial Optimization (2019) 38:808–835
https://doi.org/10.1007/s10878-019-00413-1

An evolutionary approach for the target search problem in uncertain environment

M. Barkaoui[1] · J. Berger[2] · A. Boukhtouta[3]

Published online: 16 May 2019

Abstract

Search path planning is critical to achieve efficient information-gathering tasks in dynamic uncertain environments. Given task complexity, most proposed approaches rely on various strategies to reduce computational complexity, from deliberate simplifications or ad hoc constraint relaxation to fast approximate global search methods utilization often focusing on a single objective. However, problem-solving search techniques designed to compute near-optimal solutions largely remain computationally prohibitive and are not scalable. In this paper, a new information-theoretic evolutionary anytime path planning algorithm is proposed to solve a dynamic search path planning problem in which a fleet of homogeneous unmanned aerial vehicles explores a search area to hierarchically minimize target zone occupancy uncertainty, lateness, and travel/discovery time respectively. Conditioned by new observation outcomes and request events, the evolutionary algorithm episodically solves an augmented static open-loop search path planning model over a receding time horizon incorporating any anticipated information feedback. The proposed approach has shown to outperform alternate myopic and greedy heuristics, significantly improving relative information gain at the expense of modest additional travel cost.

Keywords Evolutionary algorithms · Target search · Unmanned aerial vehicle · Dynamic path planning

✉ M. Barkaoui
mohamed.barkaoui.1@ulaval.ca

[1] Département d’informatique et de génie logiciel, Université Laval, 1065, av. de la Médecine, Quebec, QC G1V 0A6, Canada

[2] Defence Research Development Canada - Valcartier, 2459 Pie-XI Blvd. North, Québec, PQ G3J 1X5, Canada

[3] Defence Research Development, Canada - CORA, 101 Col. By Drive, Ottawa, ON K1A 0K2, Canada

1 Introduction

Target search path planning consists in constructing a path that enables one or many vehicles, robots, or UAVs to successfully search and detect one or multiple targets. Despite recent progress in information technology and analytics (e.g. computational intelligence), the search path planning problem is NP-Hard (Stone 1975). Path planning computational complexity is a major challenge for UAVs evolving in a time-constrained uncertain environment as solution plan must be computed in real time. Should the path planner fail to generate a feasible solution by a predetermined deadline, mission failure would possibly occur.

Common target search optimization problem objectives include probability of detection maximization and expected detection time minimization. Stone (1975) offers significant insight into the progression of an optimal search. Search plans for static single targets with optimal effort allocation have been discussed with the restrictive assumption of exponential detection functions. Lo et al. (2012) and Berger et al. (2016) addressed the static target search path planning problem with homogeneous sensors and heterogeneous sensors respectively, using a mathematical programming framework. Reported results have nonetheless been limited to a small number of sensors/searchers. Bekhti et al. (2016) similarly used an integer linear programming framework to model the path planning of autonomous UAVs where tracking capabilities are provided by terrestrial wireless networks. The problem is formalized as a constrained shortest path problem, where the objective is to minimize the delay for reaching a destination, while ensuring a certain message delivery rate in reporting the drone's positions. Bayesian search focuses on how to estimate the target's position based on probability theory. Search area is generally divided into a finite set of cells, individually characterizing probability of cell occupancy by the target. The goal is to determine the optimal path to find the lost or moving target(s). Lanillos et al. (2012) and Perez-Carabaza et al. (2016) used the cross entropy approach to solve an optimization problem with a discounted probability of detection objective function, aimed at minimizing target search time. The problem of searching a lost target at sea by a single autonomous sensor platform (UAV) is in counterpart discussed in Bourgault et al. (2003). In this case, the target may be static or mobile, but not evading. The paper presents a Bayesian approach to the problem and the feasibility of the method is investigated using a high fidelity UAV simulator. Many contributions on search path planning may alternatively be found in the robotics literature, in particular in the robot motion planning field (see e.g. Botzheim et al. 2012) such as terrain acquisition and coverage path planning (Wong and MacDonald 2003; Svennebring and Koenig 2004; Yang and Luo 2004). Robot motion planning envisioned search path planning primarily focusing on coverage problem instances from a constrained shortest path perspective (Rekleitis et al. 2008; Agmon et al. 2008). A survey reviewing the most relevant work for path planning problem for robotics is presented in (Galceran and Carreras 2013). Robotics path planning heuristic-based algorithms are reported in (Mac et al. 2016) for coverage problems, while reference (Otte 2017) summarizes machine learning approaches. Problem variants emphasizing specific features or settings including collision avoidance, target multiplicity, decentralized and dynamic decision-making have also been proposed and are briefly summarized next.

Search path planning assuming unknown static targets, obstructions and an uncertain environment presents a natural variant of the basic problem. In Lin and Saripalli (2014), a path planning algorithm reinforcing UAV obstacle avoidance based on 3D Dubins curves to avoid static and moving obstacles was developed. An operational UAV platform using a 3D model of a region containing buildings and road structures to autonomously generate collision-free paths is presented in Pettersson and Doherty (2006). Similarly, Hrabar (Hrabar 2008) uses probabilistic roadmaps for path planning and uses stereo vision for detecting obstacles and dynamic path updating. In Chia et al. (2010), the authors developed an ant colony algorithm to compute a search path plan ensuring collision-free mobile robot navigation control toward the target position. However, their ant colony algorithm is limited to single robot systems. Two approaches are further proposed in Sun et al. (2017) to find the minimum path-time path for a UAV flying from fixed starting and ending points with avoidance zones. The first consists in modelling the path planning problem as a constrained quadratic programming problem whereas the second integrates flight kinematic constraints in a sampling-based heuristic search to generate smooth paths. A collision-free path planner exploiting parametrized Euler Spirals for a group of UAVs has also been developed (Dai and Cochran 2009).

A few open literature contributions on search path planning considering multiple targets are worth mentioning. Considering a team of UAVs searching for an unknown number of targets, Wood and Hedrick (2011) use a target density distribution to estimate expected number of targets over a surveillance area. The path planning strategy involves a two-layer structure. A first layer views target partition as tasks and allocates them to the UAV team while the second layer determines optimal agent search paths over partitions characterizing target probability density distribution. More recently, Freundlich et al. (2015) alternatively proposed an optimal path planning approach for multiple target localization using Kalman filtering. Accordingly, a dynamic program aimed at computing optimal trajectories in the joint state-space of uncertain robot positions and target locations is solved for each target.

Expanding from centralized problem settings, cooperative target search planning has been partly investigated as well. Vidal et al. (2002) developed a coordination model for UAV/UGV (unmanned air/ground vehicles) team searching for multiple targets and empirically proved their approach to locate evading targets. A cooperative path planning algorithm for tracking a moving target in urban environments using both UAVs and UGVs is alternatively proposed by Yu et al. (2015). In that case, the target state is updated using a Bayesian filter. The path planning algorithm consists in maximizing the sum of the joint probability of detection over a finite look-ahead horizon for all vehicles. In counterpart, Bourgault et al. (2003) describe a decentralized Bayesian approach for locating a single target by coordinating multiple autonomous agents. In their framework, automated search agents make individual decisions based on local knowledge (prior probability distribution) and information gathered by the different sensing platforms. Information is combined using a fully decentralized Bayesian data fusion technique, and controls are given using a decentralized coordinated control scheme. Tisdale et al. (2009) proposed an online path planning approach for a team of UAVs cooperatively searching for a single stationary target. A Bayesian filter and a receding-horizon with an objective function that captures information gain has been used to build path plans. The work proposed by Yang et al. (2005) consists in cooperatively optimizing dynamic path planning to successfully search and detect multiple stationary targets. It relaxes some of the constraints generally imposed by classical search approaches. These relate to partially observable environment state, explicit information feedback exploitation, imperfect sensors (e.g. false-alarm), limited computational resources, or a mix of them.

Reported work on dynamic target search path planning presents a different problem perspective. Dynamic task allocation and multiple autonomous vehicles path planning are concurrently handled by Yuan et al. (2009). It combines multiple autonomous systems dynamically planning collision-free paths. Peng and Demin (2012) alternatively present an intelligent online path planner (OPP) computing UAV paths, where the problem is modeled as a dynamic multi-objective optimization problem. The Bayesian network and fuzzy logic are used to quantify respective bias introduced by each optimization objective to successfully select the best path from the output of the OPP. The main idea is to use historical information and collected Pareto sets to construct several time series to guide search.

Despite problem diversity, most proposed problem-solving approaches often rely on common computational complexity reduction strategies to mitigate the curse of dimensionality, from ad hoc constraint relaxation to fast or global search methods utilization promoting low-cost computation. Some of these techniques include sampling (Lanillos et al. 2013), coarse-grained environment representation (Lanillos et al. 2012) and limiting action parameter specification (Lanillos et al. 2014). Problem complexity reduction strategies are coupled to traditional search theory methods such as branch and bound (Washburn 1998; Lau and Dissanayake 2005) or A* search techniques and related variants. However, in this case, the main challenge in designing efficient heuristics lies on the discovery and computation of tight bounds which remain largely difficult to achieve in practice (Lau 2007). In counterpart, exact closed-loop modeling and problem-solving techniques such as policy-iteration, value-iteration or related dynamic programing method variants are also exploited to handle partially observable Markov sequential decision problem formulations (Brooks et al. 2006; Seuken and Zilberstein 2008). But, these approaches are prohibitive and do not scale showing serious practical limitations.

In this paper, a new information-theoretic-based evolutionary approach is proposed to solve the dynamic search path planning problem. The effectiveness of the proposed approach is assessed using a modified version of a hybrid genetic algorithm (GA), previously reported in Barkaoui et al. (2014, 2015). The algorithm developed in Barkaoui et al. (2015) was designed for the dynamic vehicle routing problem with time windows to capture customer satisfaction over multiple visits. In Barkaoui et al. (2014), some preliminary results for the dynamic search path planning problem were reported. The new approach defines an information-theoretic evolutionary anytime path planning algorithm to solve a dynamic search path planning problem in which a fleet of homogeneous unmanned aerial vehicles (UAVs) explores a search area to hierarchically minimize target zone occupancy uncertainty (entropy), lateness, and travel time respectively. Conditioned by new observation outcome and exogenous request events, the evolutionary algorithm episodically solves an augmented static open-loop search path planning model over a receding time horizon incorporating any anticipated information feedback. The problem search area decomposed in zones, assumes a prior zone occupancy (target) probability distribution to be known. Entropy objective contribution function separability over local zones, coupled to conditional independence of anticipated observation outcome events, enable efficient prior objective function precomputing, leading to a novel decision problem model formulation. The new decision model which incorporates sensor false alarms and anticipated action outcome feedback naturally lends itself to parallel computing, paving the way to rapidly solve practical size problems. Problems with a large time horizon are dynamically handled by repeatedly solving a revisited open-loop problem instance triggered by a new event over a receding horizon, using real information feedback (observation outcome) from the previous episode. The advocated strategy exploits episodic information feedback to opportunistically improve solution quality over a limited period rather than averaging path solution performance over a distant time horizon for all possible outcome sequences. We use a genetic algorithm to construct a set of potential paths containing existing zone visits as well as possible future revisits. The algorithm refines current zone visit plans while considering optional local revisit requests. A comparative computational experiment has been conducted to show potential gain of the proposed approach. The algorithm outperforms alternate myopic and greedy heuristics, significantly improving relative information gain at the expense of a small additional travel cost.

The remainder of the paper is organized as follows. Problem definition, describing the main characteristics of the dynamic target search path planning problem, is first introduced. Then, the main solution concept is presented, introducing a new information-theoretic-based evolutionary approach to compute a near-optimal solution. The next section reports and discusses computational results comparing the value of the proposed method with alternative techniques. Finally, a conclusion is given in the last section.

2 Target search path planning problem description

The problem involves a fleet of homogeneous UAVs, searching stationary targets in a bounded environment over a given time horizon. It is considered as a targeted closed-loop search centralized path planning problem with a hierarchy of objectives. The first objective is to maximize information gain or equivalently to minimize uncertainty or entropy about target occupancy within a given search region, the second consists in minimizing lateness, and the third aims at minimizing target discovery time. The proposed multi-objective hierarchy is aligned to a lexicographic ordering, for which overall quality of computed solutions are ranked against the respective related objectives, in that order. Remaining responsive to dynamic exogenous events, the evolutionary approach consists in solving an open-loop search path planning with anticipated feedback (action outcomes/observations) problem model over a rolling horizon while gradually incorporating information feedback made available. Performed by a single base control station, a path planning episode takes place over a

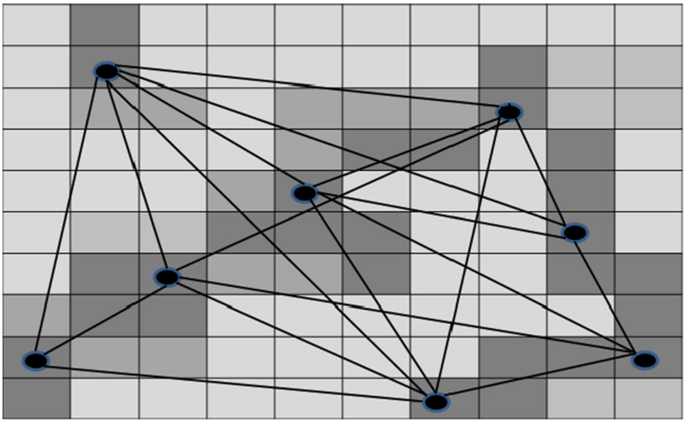

Fig. 1 Graph representation of a search area forming multiple zones. Nodes capture geographical zones or search sub-regions while edges connecting zones reflect feasible UAV moves at each decision cycle

period separating two successive events. The search region is composed of geographically distributed zones, possibly populated by non-cooperative stationary targets. A representation of the search region is given in Fig. 1. The search area is represented as an undirected graph in which a node reflects a geographical zone or search sub-region and an edge defines a feasible zone transition describing a legal UAV move. Targets separately occupy a single zone, and a zone may contain multiple targets. Target cardinality and respective positions are assumed unknown. Derived from domain-dependent situation models reflecting specific conditions as well as terrain and intrinsic target behavioral properties, a probability density distribution defining individual zone occupancy characterizing prior single or multiple targets presence in related location is assumed to be known. Occupancy probability distribution is assumed zone-independent. Zones are visited by a homogeneous fleet of UAVs, initially located at a central base station. Paths are assumed to start and end at the central base station. All zones must be visited within a specified time interval and a specific deadline is set to safely complete all surveillance/reconnaissance tasks. Under centralized control, UAVs (typically surveillance/reconnaissance drones) act as stand-off imperfect sensing agents collecting sensor readings while periodically communicating state and plan information back and forth with the base control station. UAV's speed and zone visit time are assumed constant. Reflecting vehicle's autonomy, UAV flight time should not exceed a predetermined maximum travel time. The team embraces a simple collision avoidance policy in which members fly at different altitudes. A UAV path solution to the search problem aims at minimizing uncertainty (entropy) over zone target occupancy, lateness and target discovery time respectively.

In the current dynamic problem setting, we assume the next UAV destination (intended) to be communicated by the dispatching system at each visit location, or upon zone visit completion. The next destination is determined according to the best computed path plan available. The dispatching system may advise of any new destinations during the problem-solving process as necessary. It should be emphasized that a

UAV traveling to its next destination is fully committed to visit its target. Consequently, aspects such as diversions for a moving UAV have not been considered.

2.1 Probabilistic observation model

Our sensing model is based on the fact that during period t, a UAV visits a zone to determine target occupancy (i.e. presence of at least one target). Modeling partial world state agent observability, the observation model governs agent sensor's perception (Berger et al. 2009). A sensor reading obs_t at time t may then be either positive ($obs_t = 1$) or negative ($obs_t = 0$) as determined through a probabilistic observation model. The uncertainty is modeled using conditional probability of detection and false alarm, given zone target vacancy or occupancy state $x \in X = \{0,1\}$. If we suppose obs_t is the observation of zone occupancy at the end of period t: {positive = 1, negative = 0}, then we have $p_c = p\ (obs_t = 1 | x = 1)$ indicating the probability of correct observation and $p_f = p\ (obs_t = 1 | x = 0)$ the probability of false alarm.

These two probabilities are zone-dependent reflecting specific sensor sensitivity and terrain features and conditions (e.g. landscape, visibility, obstacles, clutter,). It is assumed that the observation model is known by the decision-maker. Agent sensor's range defining visibility or footprint (coverage of observable zones given the current sensor position) is limited to the zone being searched.

The local zone target occupancy beliefs, $p(x = 1)$, can be updated using Bayesian filtering (from a real or anticipated UAV sensor observation):

$$p_t(x|obs_t) = \frac{p(obs_t|x)p_{t-1}(x)}{\sum\limits_{x \in X=\{0,1\}} p(obs_t|x)p_{t-1}(x)} \tag{1}$$

Probabilities p_{t-1} and p_t refer to prior and posterior zone target occupancy probability respectively.

2.2 Entropy function

A centralized decision-making process episodically makes a vehicle's search path planning decision based on vehicle's position. The objective consists in constructing a plan modeled as a sequence of moves to minimize entropy (target occupancy uncertainty) over the entire region. From information theory (Cover and Thomas 2006), the entropy function E is defined as:

$$E = \sum_{x \in \{0,1\}} p(x) \log_2 p(x) \tag{2}$$

where $p(x)$ specifies the current probability/belief of zone target occupancy, and x a binary zone occupancy state. A zone with a zero entropy value means absolute certainty about occupancy or vacancy, whereas a maximum entropy value (1) refers to complete uncertainty. The path planning objective consists in minimizing the entropy function and the decision-making is subject to limited computational resources imposed by episode duration.

We define in our model an entropy threshold E^* to determine a maximum number of zone visits. Consequently, when zone entropy is smaller than E^*, then certainty may be assumed about target occupancy otherwise, a new visit is made. The resort to a maximum number of visits helps prevents excessive zone revisit requests. Visit ordering on a given zone has no impact on zone entropy even if the zone is visited by different UAVs. However, cumulative lateness and total travel time may be affected.

3 Genetic-based target search heuristic approach

A hierarchical approach provides a way to specify relative objectives importance to the decision maker who ranks them accordingly. It is deemed preferable to a weighted multi-objective approach for the target search path planning problem presenting a high-level hierarchy of preferences over natural solutions ordering. The information-theoretic target search framework proposed in this paper is inspired from the genetic algorithm paradigm. Target search ultimately results from the simultaneous evolution of two populations of “plan” individuals describing a sequence of zone visits over a given horizon. An individual’s score is computed and ranked according to the hierarchical objective structure following lexicographic ordering, that is: minimize uncertainty or expected system entropy about target occupancy within a given search region, minimizing lateness, and minimizing travel discovery time.

3.1 Objective function

The information-theoretic based approach captures uncertainty related to target occupancy, projecting expected entropy resulting from anticipated path plan execution. As target occupancy may be assumed independent with respect to the zones of interest, and observation outcome mainly rely on current UAV position, the objective function is composed of separate contributions partitioning subsets of decision variables. As zone entropy essentially depends on effective local visits, the global expected entropy objective function is separable. Objective separability enables expected zone entropy $\hat{E}_{z,l}$ precomputation, where z is the zone (node in the graph) and l is number of visits (multiplicity). Expected entropy is determined over possible future observation outcome scenarios involving imperfect UAV sensors which characterize partial environment observability and ultimately anticipated information feedback resulting from path plan execution. Simulating a given path plan, expected zone entropy (uncertainty) is computed over all possible courses of observation outcomes. As zone visit ordering is ultimately invariant with respect to expected zone entropy, symmetry on sequence of observation outcomes may be further exploited. Hence, for a given number s of success (anticipated positive observations) and l-s failure (anticipated negative observation outcomes) events, an l-visit scenario is governed by a binomial distribution over possible observation outcomes. Accordingly, using Eq. (2), local expected entropy $\hat{E}_{z,l}$ resulting from a simulated path plan which assumes l visits over zone z is defined as follows:

$$\hat{E}_{z,l} = \sum\nolimits_{s=0}^{l} p_z(s|l)\ E(\{p_z(x|s,l)\}) \quad l \geq 1 \tag{3}$$

where

$$p_z(x=1|s,l) = \frac{1}{1 + \alpha_z^s \beta_z^{l-s}\left(\frac{1}{p_{z,0}(x=1)} - 1\right)}; \quad p_z(x=0|s,l) = 1 - p_z(x=1|s,l) \tag{4}$$

and

$$p_z(s|l) = \sum\nolimits_{x \in X} p_z(s|l,x) p_{z,0}(x) = \ p_z(s|l,x=1) p_{z,0}(x=1) + p_z(s|l,x=0)\big(1 - p_{z,0}(x=1)\big) \tag{5}$$

with

$$p_z(s|l,x=1) = \frac{l!}{s!\,(l-s)!}\left(p_z^c\right)^s \left(1 - p_z^c\right)^{l-s}$$

and

$$p_z(s|l,x=0) = \frac{l!}{s!\,(l-s)!}\left(p_z^f\right)^s \left(1 - p_z^f\right)^{l-s}$$

and

$$\alpha_z = \frac{p_z^f}{p_z^c}; \quad \beta_z = \frac{1 - p_z^f}{1 - p_z^c} \tag{6}$$

$$\hat{E}_{z,0} = E_{z,0} \tag{7}$$

Equation (3) reflects average entropy over possible scenario combinations characterizing s successful observations out of l visits in zone z. Conditional ($s|l$) scenario entropy contribution is weighted by the probability $p(s|l)$ of such a scenario (s anticipated positive observations given l visits). Posterior zone z target occupancy belief for a s-success l visit scenario is represented by Eq. (4). It naturally emerges from expression (1) after multiple observations, where $p_{z,0}$ stands for the initial target occupancy belief in zone z. The expression $p_z(s|l,x)$ is a binomial probability distribution of positive observations, giving the probability to obtain s success out of l visits on zone z, conditional on occupancy state x. The probability of correct observation in zone z is described by p_z^c and the probability of false alarm (not correct observation) in zone z is described by p_z^f, as introduced earlier. $E_{z,0}$ reflects current zone z entropy. The use of homogeneous vehicles and the exploitation of symmetry over equivalent sequence of success/failure events reduce complexity to a linear number of scenarios to be examined, as specific scenario event order does not matter.

3.2 Information-theoretic-based evolutionary approach

The objective function consists in minimizing uncertainty or entropy about target occupancy within a given region. Therefore, anticipatory information about zones that may require subsequent visits (revisits) in the near future is explicitly considered. Consequently, the approach consists in continuously generating plans that are consistent with past decisions while anticipating future requests by considering both existing and potential future visit requests during plan generation. More specifically, the strategy introduces "dummy" revisits representing likely revisit request projections in UAV paths. The expected number of such visits *vis* for a given zone *z*, which may require service in the near future, is conditionally determined by the following formulas:

$$\left\{ \frac{1}{1+\beta_z^{vis}\left(\frac{1}{p_{z,0}(x=1)}-1\right)} \leq p_1^* \qquad if\ p_{z,0}(x=1) \leq 0.5 \right. \tag{8}$$

$$\left\{ \frac{1}{1+\alpha_z^{vis}\left(\frac{1}{p_{z,0}(x=1)}-1\right)} \geq p_2^* \qquad otherwise \right. \tag{9}$$

where *vis* represents the minimum number of zone *z* visits assumed necessary to confirm target occupancy state. User-defined target occupancy probability thresholds p_1^* and p_2^* are used to confirm target vacancy and occupancy respectively.

A genetic algorithm is used to construct a set of potential paths containing existing zone visits as well as possible future revisits. Consequently, the algorithm captures path construction look ahead without violating temporal constraints. The advantage of that is a larger number of zone visits can then be considered in building a solution, leading possibly to some quality improvement. The strategy is based on the assumption that better solutions may emerge when taking advantage of a possibly larger number of zone visits due to additional requests that may occur in the near future. It is therefore assumed that better opportunities generated by considering possible future requests compensate the cost of myopic scheduling opportunities, ultimately resulting in solution improvement. The maximum number of revisits (user-defined parameter) represents the number of steps to plan ahead.

Individuals represent expandable solutions capturing currently planned visits, as well as previously serviced zones, while dynamically accommodating incoming zone visit requests. Individuals are represented as chromosomes encoding for a given time step, a feasible path plan expressed as a sequence of intended zone visit actions (physical moves) to be executed over a specific time horizon T. Each UAV is connected with one path. Zones may require several visits before confirming target occupancy state (zone entropy reaches some threshold value E^*). Zone entropy is updated after each visit using Eq. (2) and (4). When a zone is visited, outdated planned visits from all solution individuals are then delayed accordingly and updated using a large penalty cost for lateness. Feasible solutions for initial populations are first generated using a sequential insertion heuristic in which zone visits are arbitrarily inserted in random insertion positions within paths. This strategy is fast and simple while ensuring unbiased solution generation.

The processing of new zone visit request is primarily entropy-driven. A new zone service request automatically occurs when zone entropy resulting from the latest visit is still significantly large. If multiple visits are candidates for insertion in planned paths, the first visit to insert will be the one minimizing expected zone entropy defined by Eq. (3). The insertion strategy related to zone visit requests is described as follows:

- A new visit request is inserted to minimize solution expected entropy. Visit requests and potential future revisit requests are calculated using Eqs. (8) and (9), and considered for insertion.
- A new zone visit request along with its potential future revisit requests using Eqs. (8) and (9) are first considered for insertion in the current plan solution.
- New revisit request on a given zone replaces one of its anticipated revisit, if any; otherwise it will be inserted in one of the planned paths of the current solution, assuming an admissible insertion position. Should such admissible insertion position not exist, a beneficial exchange with planned revisits would be explored. Otherwise the zone revisit request will be stored and reconsidered for future insertion.
- A zone visit can be removed (for exchange purposes) from planned path to insert another visit (visit with the maximum expected information gain) which could minimize entropy. Removed visits can be subsequently reinserted in the planned paths.
- Anticipated zone revisits that do not occur are ultimately removed from planned paths.
- Zone visit insertion is biased toward entropy minimization first, and then conditionally on at least one feasible insertion position, the resulting solution is further refined to minimize lateness and travel time. Accordingly, once a zone visit can be inserted in a solution path plan, alternate solutions may be explored for lateness and travel time as entropy is order-independent in terms of zone visits. Alternatively, should feasible insertion of a new zone request fail in the first place, an exchange procedure would be used to search alternate high-quality solutions.

3.2.1 Information-theoretic-based evolutionary algorithm

The new information-theoretic-based evolutionary approach is performed within a previously reported hybrid genetic algorithm heuristic (Barkaoui et al. 2015). Zone entropy and zone revisits are included for consideration in the overall problem-solving process. With this modification, the overall problem-solving process algorithm can be outlined as follows:

```
Initialize populations with currently known zones

Repeat
    If (no external event occurs) Then
        Search dynamically optimizing paths using a genetic algorithm
        Schedule (visit commitment) next visit(s) if required
        Update best computed solution
    Else -- event occurrence
        Stop the genetic algorithm
        If (end of visit at zone location) Then
            Calculate zone entropy using Eq. (2) and (4)
            If (zone entropy > $E^*$) Then
                Generate new zone visit request (should the maximum number of visits is not
                reached)
            Else – target occupancy state confirmed
                Remove anticipated visits of the zone, if any, from planned paths
                Determine UAV's next planned visit from best computed solution
            If (planned next visit involves a waiting time) Then
                Delay UAV's departure for that period
                Update population solutions incorporating the next planned UAV's visit (next
                planned destination)
        Else -- new zone visit request
            If (it is an anticipated visit) Then
             Replace the anticipated visit by the new visit
            Else – not an expected visit
                Explore admissible visit insertion in population solutions – using the insertion
                procedure, or should it fail, the exchange procedure described below.
                If (at least one admissible insertion position or one beneficial visit exchange
                  exists) Then
                    Update population members and best-computed solution accordingly
                Else -- no admissible solution
                    Request will be reconsidered for future insertion
    Reactivate the genetic algorithm (the GA continues from the point when it was stopped)
Until (end of the horizon period)
```

3.2.2 Genetic algorithm

General description The algorithm mainly relies on the basic principles of genetic algorithms, disregarding explicit solution encoding issues for problem representation. During a pre-optimization phase, solutions are first generated as a set of paths, covering known zones. These solutions are expanded to handle incoming visit requests while properly accounting for previously visited zones or committed visits. Zone cardinality is identical across both populations and the number of paths is bounded by the size of the vehicle fleet. Key events dynamically changing solution individuals include "visit commitment" and "new visit request". Visit commitment modifies the active root (latest scheduled zone) of the developing path while maintaining state consistency across solution individuals. The occurrence of a new visit request expands the length of the solution individual incorporating an additional zone visit. Feasible zone insertions are explored to minimize lateness and travel time. If all possible insertions across populations ultimately violate the deadline associated with the problem time horizon or the path time constraint, the zone visit will be reconsidered for future insertion in the planned paths. When a new zone visit is successfully inserted in at least one solution, the insertion is also included in alternate population members, even at the cost of reconstructing some solution individuals. This requirement ensures solution homogeneity (same set of visits) and a maximum number of serviced zones. Since visit ordering on a given zone has no impact on zone entropy, only travel time and lateness are considered in fitness and objective functions to evaluate solution individuals. An exchange procedure exploring alternate path visit insertion swaps from a subset of pending requests to maximize information gain is then executed. A post-processing procedure aimed at reordering zone visits is further applied to improve the current best solution. The evolutionary process is repeated until a predefined stopping condition is met.

A steady-state genetic algorithm involving two overlapping populations is proposed. At first, new individuals are generated and added to the current population $\mathrm{Pop_p}$. The process continues until the overlapping population outnumbers the initial population by n_p. Then, the n_p worst individuals are eliminated to maintain population size using the evaluation function. Individual evaluation ($Eval_i$) is computed as follows:

$$Eval_i = \varepsilon_i + CV_i, \tag{10}$$

where

$$\varepsilon_i = r_i + \gamma t_i, \tag{11}$$

$$CV_i = \sum_{z=0}^{n} \mu_z \max\{0, b_z^i - l_z\} + \delta \; Viol_i \tag{12}$$

ε_i : basic evaluation of individual (solution) i,
r_i: number of UAV paths in individual i,
γ: relative user-defined weight parameter controlling the time contribution. The value of γ is selected to privilege solutions having a smaller number of paths (e.g., $\gamma = 1/t_m$, where t_m refers to the maximum travel time over the individuals forming the initial population),
CV_i: time period constraint violation associated to individual i,
$t_{i:}$ total travel time related to individual i,
n: number of zones (z = 0 refers to the central base station),
μ_z : penalty associated with time period constraint violation z,

b_z^i: scheduled time to visit zone z in individual i,
l_z : latest time (upper bounds of time period) to visit zone z,
δ : penalty associated with the number of violated time period constraints,
$Viol_i$: number of time period constraints violated in individual i.

The evaluation function indicates that better individuals generally (but not necessarily) include fewer paths (UAVs), and smaller total travel time, while satisfying temporal constraints. It is consistent with the three objectives mentioned earlier, namely the minimization of entropy and, total lateness and then target discovery time.

A cycle of the proposed genetic algorithm is specified as follows:

```
p=1
Repeat   {evolve population $Pop_p$ - new generation}
  For $j$ =1..$n_p$ do
    Select two parents from $Pop_p$
    Generate a new solution $S_j$ using crossover and mutation operators associated with $Pop_p$
    Add $S_j$ to $Pop_p$
  end for
  Remove from $Pop_p$ the $n_p$ worst individuals using the evaluation function (Eq. 10)
  $p$=$p$+1
Until (all populations $Pop_p$ have been visited)
Apply the post-processing procedure on the best solution {visit reordering}
Population migration {local best solutions exchange across populations}
```

Fitness The used genetic algorithm relies upon concurrent population evolution and partial temporal constraint relaxation (Barkaoui et al. 2015). For a matter of convenience, two populations *Pop₁* and *Pop₂* are chosen to emphasize solution diversification. Individual fitness is computed as follows:

$$Pop_1 : fitness_i = t_i + \sum_{z=0}^{n} \mu_z \max\{0, b_z^i - l_z\} + \delta \; Viol_i \tag{13}$$

$$Pop_2 : fitness_i = \sum_{z=0}^{n} \mu_z \max\{0, b_z^i - l_z\} + \delta \; Viol_i \tag{14}$$

The notations are the same as in Eq. (10)–(12). Better individuals generally (but not necessarily) tend to include short total travel time in Pop_1 and satisfy as many temporal constraints as possible in Pop_2.

Genetic operators The proposed genetic operators mostly rely on two basic principles. First, for a given number of paths, an attempt is made to construct feasible solutions with as many zone visits as possible. Second, remaining zones are inserted within existing paths, violating temporal constraints to keep the total number of paths allowed to a fixed value. The proposed genetic operators incorporate some key features of the best heuristic routing techniques such as Solomon's insertion heuristic I1

(Solomon 1987), large neighborhood search (Shaw 1998), and the neighborhood-based two-stage metaheuristic (Liu and Shen 1999). Two crossover operators are developed, namely IB_X and LNSB_X (large neighborhood search-based), and a suite of five mutation operators is proposed, namely EE_M, RS_M, IEE_M, RSP_M and RZ_M. Each operator is briefly described next.

Selection A fitness proportional scheme is used to select parent candidates. Accordingly, an individual is selected with a probability proportional to its fitness value. Fitness functions given by Eq. (13), (14) are used for Pop_1 and for Pop_2 respectively.

Crossover The insertion-based IB_X(k) crossover operator creates an offspring by combining, one at a time, k paths (R1) of parent solution P1 with a subset of zones, formed by nearest-neighbour paths (R2) in parent solution P2. The neighbourhood R2 includes the paths of P2 whose centroid is located within a certain range of r1 $\in$ R1 (centroid). This range corresponds to the average distance separating r1 from the paths defining P2. The paths of R1 are selected either randomly, with a probability proportional to the number of zones characterizing a path or based on average distance separating consecutive zones over a path. A stochastic removal procedure is first carried out to remove from r1, zones' subject to be migrated to alternate paths. Targeted zones are either selected according to waiting times, distance separating them from their immediate neighbours, or randomly. Then, using a modified insertion heuristic inspired from Solomon (Solomon 1987) a feasible child path is constructed, expanding the altered path r1 by inserting zone visit candidates derived from the nearest-neighbour paths R2 defined earlier. The proposed insertion technique consists in adding a stochastic feature to the standard insertion heuristic I1 (Solomon 1987) by selecting randomly the next zone visit over the three best candidates with a bias toward the best. Once the construction of the child path is completed, and reinsertion is no longer possible, a new path construction cycle is initiated. The overall process is repeated for the k paths of R1. Finally, the child inherits the remaining "diminished" paths (if any) of P1. If unrouted zones still remain, additional paths are built using a nearest-neighbour procedure. The whole process is then iterated once more to generate a second child by interchanging the roles of P1 and P2. The operator is briefly sketched in Fig. 2.

The LNSB_X crossover operator relies on the concept of the Large Neighborhood Search (LNS) method proposed by Shaw (Shaw 1998) to create an offspring. LNS consists in exploring the search space by repeatedly removing a subset of zones (R) from parent solution P1 and reinserting them using constraint-based tree search (constraint programming). The subset R includes related zones in parent solution P2 and zones violating temporal constraints. Zone relatedness defines a relationship linking two zones based upon specific properties (e.g., proximity and/or identical path membership), such that when both zones are considered simultaneously for a visit, they can compete with each other for reinsertion creating new opportunities for solution improvement. Therefore, zones close to one another naturally offer interchange opportunities to improve solution quality. Similarly, the number of paths in a solution is more likely to decrease when zones sharing path membership are removed all together.

As stated in Shaw (1998), a set of related zones is first removed. Then, zones violating temporal constraints are taken out as well, before proceeding to the reinsertion

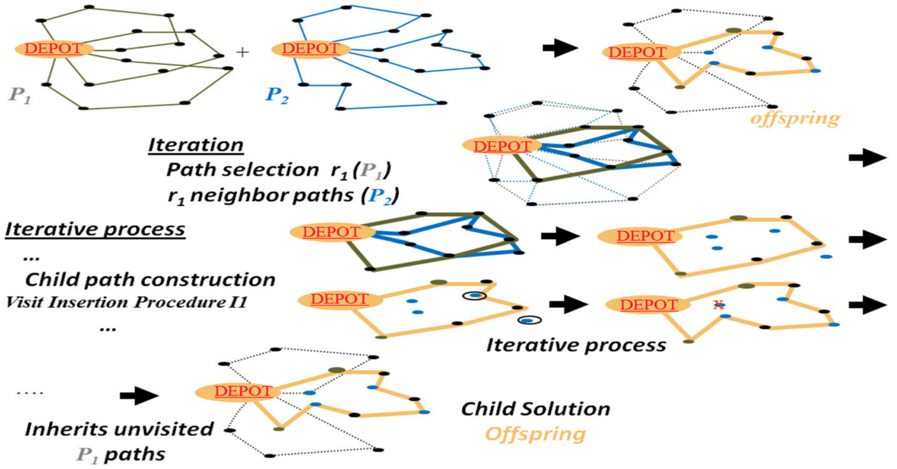

Fig. 2 Insertion-based (IB_X) crossover creates an offspring by combining iteratively various paths of a parent solution P1 with a subset of visits, formed by nearest-neighbor paths from parent solution P2. Exhaustive parent solution path representations have been omitted for clarity purposes

phase. The proposed zone reinsertion method differs from the procedure proposed by Shaw (1998) in two respects, namely, the insertion cost function used, and the order in which zones are visited for insertion through the search process. Insertion cost is defined by the sum of key contributions referring respectively to increased traveled distance, and delayed service time, as prescribed in Solomon's procedure I1, as well as constraint violation (see Eq. (12)). As for zone ordering, zones ({c}) are sorted (*CustOrd*) according to a composite ranking, departing from the myopic scheme originally proposed by Shaw. The ranking is defined as an additive combination of two separate rankings, previously achieved over best insertion costs ($Rank_{Cost}$(c)) on the one hand, and number of feasible insertion positions ($Rank_{|Pos|}$(c)) on the other hand:

$$CustOrd \leftarrow Sort(Rank_{Cost}(c) + Rank_{|Pos|}(c)) \tag{15}$$

The smaller the insertion cost (short total distance, traveled time) and the number of positions (opportunities), the better the ranking. The next zone to be visited within the search process is selected according to the following expression:

$$zone \leftarrow CustOrd\left[INTEGER\left(L\,rand^{D}\right)\right] \tag{16}$$

in which L = current number of zones to be inserted, *rand* = real number over the interval [0,1] (uniform random number generator), D = parameter controlling determinism. If D = 1 then selection is purely random (default: D = 15).

Once a zone is selected, search is carried out over its different insertion positions, exploiting limited discrepancy search as specified in Shaw (1998). However, search tree expansion is achieved using a non-constant discrepancy factor d, selected randomly (uniform probability distribution) over the set {1, 2, 3}. Despite the implicit contribution of constraint violation in the insertion method, all legal zone visits are first successfully explored. Unvisited zones (if any) are then routed relaxing temporal constraints.

Mutation The EE_M (edge exchange) and RS_M (repair solution) mutators focus on inter-path improvement. EE_M attempts to shift zones to alternate paths as well as to exchange sets of zones between two paths. It is inspired from the λ-interchange mechanism in Osman (1993), performing reinsertions of zone sets over two neighboring paths. In the proposed mutation procedure, each zone is explored for reinsertion in its surrounding path neighborhood made up of two paths. Paths are being selected such that the distance separating their centroid from zone location is minimal. Zone exchanges occur as soon as the solution improves, i.e., we use a "first admissible" improving solution strategy. Assuming the notation (x, y) to describe the different sizes of customer sets to be exchanged over two routes, the current operator explores values running over the range (x = 1, y = 0, 1, 2). As for the RS_M mutation operator, it focuses on exchanges involving one illegal zone visit. In that scheme, each illegal visit in a path is exchanged with an alternate legal one or two-zone visit sequence in order to generate a new set of zones with either violated or non-violated temporal constraints. The IEE_M (intra-path edge exchange) mutation operator is similar to EE_M except that zone migration is restricted to the same path.

The RSP_M (reinsert shortest path) mutation operator eliminates the shortest path of the solution, reducing by one the total number of paths. Zones from the shortest path are first removed. Then, following an iterative process, various solutions are explored, in which unvisited zones are reinserted within existing paths using the insertion heuristic. The entire iterative process is repeated over I different sets (e.g. I = 20) of randomly generated parameter values, and returns the best solution computed. The RZ_M (reorder zones) mutation operator is an intensification procedure intended to reduce total traveled distance of feasible solutions by reordering zones within a path. The procedure consists in repeatedly reconstructing a new path using the sequential insertion heuristic I1 over I different sets (e.g. I = 2) of randomly generated parameter values, returning the best solution generated shall an improved one emerge.

3.2.3 Exchange procedure

The exchange procedure focuses on local path improvement. It attempts to exchange sets of zone visits involving pending visits for insertion. In that scheme, each pending visit is exchanged with alternate planned zone visits in order to generate better path plans. Each current and pending visit pair is explored for swapping using the insertion procedure. Using a "first admissible" improving solution strategy, zone visit exchanges occur as soon as solution quality increases. The exchange procedure may be summarized as follows:

- For each possible "planned visit—pending visit" insertion swap, calculate related information gain.
- Identify visits to eliminate from current planned paths. Visits to be eliminated are those showing smaller gain than prospective gain from pending visits. Elimination does respect visit order.

- Eliminated visits and pending visits are sorted over information gain, in descending order.
- The insertion procedure favors visits with highest information gain first. Eliminated visits remain eligible for insertion in future iterations.

3.2.4 Time complexity

Several proofs have been developed that describe the expected convergence time and provide worst-case and average-case convergence time of genetic algorithms (Ankenbrandt 1991; Rylander and Foster 2001; Rylander et al. 2001). Ankenbrandt (1991) demonstrated that, with proportional selection, genetic algorithms have average and worst case time complexity in, $O\left(O(Evaluate(X)) * m\frac{log(m)}{log(r)}\right)$ where m is the population size, r is the fitness ratio (the average fitness of a chromosome over the average fitness function of all other chromosomes in the generation), and "Evaluate" represents the combined domain-dependent evaluation and decoding functions of the chromosome X (complexity of the fitness, mutation and crossover functions). The $O(Evaluate(X))$ is the order of the combined functions computed in every step. For the fitness and objective functions, it is in the order of $O(n*m)$, where n is the chromosome length. The crossover operator is also in the order of $O(c_p*n*m)$, where c_p is the crossover probability. The mutation complexity is in the order of $O(m_p*n*m)$, where m_p is the mutation probability. Finally, the exchange procedure is in the order of $O(n*m)$. The same bounds are assumed to be valid in our proposed approach to the dynamic search path-planning problem since basic genetic operators have been augmented, exploiting problem structure to better explore the solution space. Algorithm execution is alternately interrupted at the end of the time horizon (900 s, as defined in Sect. 4), as stochastic observation outcomes describe various dynamic evolutionary behaviours prone to a variety of performance results over a given run-time.

4 Numerical Experiments

Numerical experiments have been conducted to show the value of the proposed approach for dynamic target search problem. The experiment aims at comparing performance of the proposed evolutionary approach to alternate heuristics. Computed solutions from respective methods are reported against differential relative information gain. Differential relative information gain $RIG(T)$ between two heuristics (1 and 2) shown at the end of time horizon T is defined as follows:

$$RIG(T) = \frac{|g_T^1 - g_T^2|}{\max\left(g_T^1, g_T^2\right)} = \frac{\left|\left(E_0 - \hat{E}_T^1\right) - \left(E_0 - \hat{E}_T^2\right)\right|}{E_0 - \min\left(\hat{E}_T^1, \hat{E}_T^2\right)}$$
$$= \frac{\left|\hat{E}_T^2 - \hat{E}_T^1\right|}{E_0 - \min\left(\hat{E}_T^1, \hat{E}_T^2\right)} \tag{17}$$

where E_0 and $\hat{E}_T^i$ are the initial system entropy and the expected entropy computed by the heuristic i at the end of the time horizon T respectively. The larger the final entropy gap, the better the relative performance.

To our knowledge there is no well-known benchmark data available on target search path planning. Computer simulations were conducted for two problem instances under the following conditions. Each problem instance involves 100 zones (nodes), randomly distributed over a geographical area. The travel time separating two zones corresponds to their relative Euclidean distance. Zone locations for a problem instance are either generated randomly using a uniform distribution (problem R1) or combining randomly distributed and clustered zones (problem RC1). Problem set R1 contains 12 problem instances (r101–r112) and problem set RC1 contains 8 problem instances (rc101–rc108). Scenarios for both problem instances R1 and RC1 convey different levels of difficulty. The proportion of unknown zone requests lies between 10 and 50% (degree of dynamism). Real-time requests are generated over specific simulation time horizons (900 s). Such a scenario involves approximately three requests per minute. The total number of UAVs is fixed in advance for each scenario. The initial zone entropy value is generated from an exponential distribution.

4.1 Myopic algorithms

To the best of our knowledge no known detailed approach implementation is available for our problem instances. The performance of the proposed genetic algorithm (GA) has been compared with a myopic version of our approach (GA_Myopic), in which potential future visit requests associated with unconfirmed target occupancy state are explicitly ignored during plan generation. Also, a greedy one-step limited look ahead method (Greedy) is used for comparison purposes. The greedy procedure consists in myopically planning moves one step ahead of time, progressively visiting the zone with highest expected information gain. After a zone visit, zone entropy is updated accordingly and the next destination is the one maximizing expected information gain. The procedure is then reiterated for each episode over time horizon T.

4.2 Genetic algorithm settings

The proposed algorithm is borrowed from a previously reported hybrid genetic algorithm heuristic (Barkaoui et al. 2008, 2015) in which parameter values were empirically determined. In Barkaoui et al. (2008, 2015) parameter selection is mainly driven by good performance shown from empirical experiments trading-off run-time and solution quality. A finite set of parameter assignments is explored, discretizing the range of possible parameter values.

GA and GA_myopic both use crossover and mutation rates of 0.8 and 0.6 respectively. The migration parameter refers to the number of (best) solutions mutually exchanged between populations after each generation (Migration parameter = 2). The population size is fixed to 10 individuals and an elitist strategy is employed for both approaches. The elitist strategy consists to preserve the best two individuals to the next generation (Population overlap per generation = 2). Past experience has proved these

Table 1 Parameter values

Parameter	μ_z	δ	p_z^c	p_z^f	E^*	p_1^*	p_2^*
Value	100	100	0.8	0.1	$E\,(0.96)$	0.04	0.96

parameters to be fairly acceptable for a variety of problems, and therefore, turn out to be a natural choice for both approaches. Further information on parameter values and operator combinations for GA and GA_Myopic can be found in Barkaoui et al. (2008, 2015). Parameters in fitness and evaluation functions as well as Eq. (3)–(9) are given in Table 1.

The maximum number of visits per zone is bounded to four. We assume that all zones have the same values of μ_z,p_z^c and p_z^f.

As the purpose of the experiments is to show the feasibility and the high level of competitiveness of the proposed approach, extensive investigation of best parameter selection strategies for genetic algorithm have been deliberately overlooked.

The proposed algorithms have been implemented in C++, using the GAlib genetic algorithm library.[1] The experiment consisted in running four simulation runs and reporting average performance for each possible scenario occurrence uniformly distributed over R1 and RC1 problem instances, leading to a sample larger than thirty scenarios in each case, permitting statistical performance analysis to compare examined methods.

4.3 Results

Solutions are analyzed based on hierarchical ranking considering information gain first, total lateness second, and then, total travel time. The first analysis of the data focuses on the entropy, since this is the first objective. Since hierarchically the entropy is the first element to minimize, total lateness becomes relevant only if approaches obtain a final solution displaying the same entropy. Travel time is considered to break ties over entropy and total lateness.

Results obtained for the two problems R1 and RC1 over respective possible scenarios (i.e. 12/R1, 8/RC1) are summarized in Tables 2 and 3. The first column presents scenario identifiers. Measures of performance are the entropy (Entropy), total lateness (Lateness) and total travel time (Travel Time).

Our GA approach outperforms the greedy procedure as it clearly shows better results for all scenarios. The overall average comparative performance gap for the examined problem instances is more than 74% as shown in Table 2. Some scenarios even exhibit an 81% gain. The differential relative entropy clearly demonstrates the value of predictive/advance planning over a limited look ahead myopic attitude. Furthermore, results of the greedy approach may be explained by the ignorance of travel time in selecting next UAV destination. This often leads to long UAV paths to visit remote zones at the expense of other zones due to time constraints. In GA_Myopic and GA, visits are inserted into path plan positions minimizing lateness and total travel time. Genetic operators are naturally inclined to generate path plans with the maximum number

[1] http://lancet.mit.edu/ga/.

Table 2 Performance comparison and relative information gain of GA over Greedy

Problem instances	Greedy			GA			RIG(T)
	Entropy	Lateness	Travel Time	Entropy	Lateness	Travel Time	
r101	44.85	925.27	5986.15	29.80	90.13	5817.45	0.65
r102	44.11	954.05	5936.79	29.51	136.77	5807.75	0.63
r103	45.95	519.96	5961.96	28.28	108.07	5898.52	0.72
r104	46.00	595.82	5917.94	26.01	26.64	5927.26	0.75
r105	47.30	1859.47	5906.28	30.70	20.97	5930.67	0.75
r106	46.22	1284.71	5331.65	29.52	20.44	5381.64	0.72
r107	46.31	843.91	5363.37	29.68	58.68	5275.37	0.72
r108	45.84	95.24	5049.39	31.45	7.73	5065.40	0.67
r109	47.60	2159.29	5688.93	28.36	0.00	5631.07	0.79
r110	47.66	544.99	5363.69	29.29	0.00	5384.36	0.78
r111	47.08	392.78	5964.29	26.25	0.00	5947.78	0.79
r112	47.29	247.85	5961.35	25.34	0.00	5938.34	0.80
rc101	47.47	1235.72	6200.75	30.29	45.73	6154.25	0.76
rc102	46.96	1093.63	6177.72	28.98	59.78	6195.72	0.75
rc103	48.04	480.76	5547.41	30.71	3.62	5621.49	0.78
rc104	45.74	197.09	5918.94	26.69	5.22	5948.12	0.73
rc105	48.01	1032.15	6177.83	28.04	37.12	6195.65	0.81
rc106	46.79	826.79	5643.98	31.80	0.00	5639.69	0.71
rc107	47.10	632.48	5563.92	29.32	0.00	5579.09	0.76
rc108	49.01	64.06	4959.16	32.34	0.00	4961.11	0.81
Average	46.77	799.30	5731.08	29.12	31.04	5715.04	0.74

of visits. GA is shown to significantly improve information gain over GA-Myopic. Table 3 indicates that on average, GA decreases uncertainty by 12%, a substantial benefit. Solutions generated by GA further exhibit slightly better comparative average performance in terms of lateness and total travel time, while servicing more zones. This observation certainly reflects the value of considering potential future visit requests for a dynamic environment during plan generation.

Statistical analysis has been conducted to compare GA with greedy and Myopic_GA approaches respectively. Assuming "entropy" (first/primary objective) for a given problem instance to be a random variable, and a sufficiently large simulation sample (e.g. > 30) to approximately match a normal distribution by virtue of the central limit theorem, an analysis based on the standard student *t* test can be readily achieved. The *t* distribution can conservatively approximate a normal distribution as sample size gets larger. Accordingly, the null hypothesis H_0 asserts similar average entropy for both approaches; both algorithms in each comparison (GA vs Myopic_GA and GA vs Greedy) do not exhibit significant difference between average entropy. A bilateral student *t*-test to test the hypothesis H_0 with a 5% level of significance was then realized. Assuming the occurrence of scenarios composing a given problem (population) to be the same, a computational experiment was carried out performing 4 simulation runs

Table 3 Performance comparison and relative information gain of GA over GA_Myopic

Problem instances	GA_Myopic			GA			RIG(T)
	Entropy	Lateness	Travel Time	Entropy	Lateness	Travel Time	
r101	33.21	109..56	6006.03	29.80	90.13	5817.45	0.15
r102	32.53	335.27	6000.91	29.51	136.77	5807.75	0.13
r103	31.03	50.39	5989.47	28.28	108.07	5898.52	0.11
r104	28.54	0.00	5900.44	26.01	26.64	5927.26	0.09
r105	31.03	22.23	5949.60	30.70	20.97	5930.67	0.01
r106	33.72	12.38	5393.72	29.52	20.44	5381.64	0.18
r107	32.98	14.29	5367.23	29.68	58.68	5275.37	0.14
r108	32.82	15.18	5048.16	31.45	7.73	5065.40	0.06
r109	30.70	0.00	5688.70	28.36	0.00	5631.07	0.10
r110	34.02	0.00	5365.57	29.29	0.00	5384.36	0.20
r111	30.08	0.00	5892.45	26.25	0.00	5947.78	0.14
r112	28.20	0.00	5954.56	25.34	0.00	5938.34	0.10
rc101	35.14	1.07	6198.55	30.29	45.73	6154.25	0.22
rc102	31.34	5.36	6228.70	28.98	59.78	6195.72	0.10
rc103	32.36	2.18	5592.80	30.71	3.62	5621.49	0.08
rc104	30.68	10.70	5836.01	26.69	5.22	5948.12	0.15
rc105	33.35	5.09	6227.40	28.04	37.12	6195.65	0.21
rc106	32.32	36.73	5603.45	31.80	0.00	5639.69	0.03
rc107	31.60	30.75	5621.87	29.32	0.00	5579.09	0.10
rc108	34.01	4.71	4950.53	32.34	0.00	4961.11	0.08
Average	31.98	32.79	5740.81	29.12	31.04	5715.04	0.12

Table 4 Results of student's *t*-test for the validation of the solution approach comparison

	Problem	Sample size	Degrees of freedom	Critical *t*-value	*T*-value
GA versus GA_Myopic	R1	48	94	1.986	4.858
	RC1	32	62	1.999	4.046
GA versus greedy	R1	48	94	1.986	48.829
	RC1	32	62	1.999	36.590

per scenario. The results presented in Table 4 show the *t*-values to exceed the critical *t*-values over the two problem instances for the two comparisons. Thus, H_0 is rejected over all problem instances at a 95%-significance level. Consequently, statistically dissimilar performances expectedly demonstrate a clear dominance of the proposed GA over its myopic counterpart as well as the examined greedy approach for the targeted entropy minimization primary objective.

Differential relative information gains reported in Tables 2 and 3 comparing the proposed GA with the two examined methods are graphically represented in Fig. 3.

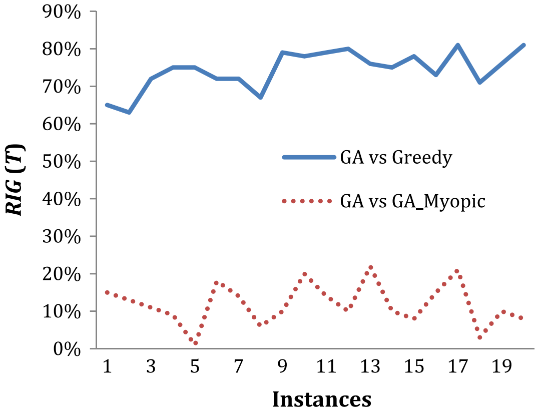

Fig. 3 Differential relative information gain *RIG*(*T*) between two heuristics (Greedy and GA_Myopic) and the proposed genetic algorithm (GA) shown at the end of time horizon *T* for the 20 problem instances reported in Tables 2 and 3

4.3.1 Results for different degrees of dynamism

A measure to describe system dynamics is very helpful when examining the performance of a specific algorithm under varying conditions. Such suitable basic measure in a path-planning context may be defined by the ratio of the number of visit requests generated dynamically relative to the total number of visit requests characterizing the problem. Accordingly, the notion of degree of dynamism was introduced as follows (Larsen et al. 2002):

$$\text{degree of dynamism} = \frac{\text{number of visit requests generated dynamically}}{\text{total number of visit requests}}$$

This measure defines values running over the interval [0, 1], with higher values indicating increasingly dynamic problems. Tables 5 and 6 present the result for each problem (R1 and RC1) and degree of dynamism (Larsen et al. 2002). Examined degree of dynamism oscillates between 10 and 90% with uniform increments. Reported measures of performance for each problem instance and selected degrees of dynamism include respectively entropy, total lateness and total travel time in that order, averaged over 10 simulation runs.

From Tables 5 and 6 it can be observed that GA outperforms competing algorithms by a large margin of more than 31% with respect to entropy, the first objective. However, this comes at the expense of an increase in travel time less than 3.5% in comparison to the alternate methods. This simply reflects the strong bias of the GA toward entropy minimization in exploring the solution space, and indicates entropy to negatively correlate with travel time, as significant reduction in entropy generally translates in servicing more requests which tend to augment travel time. In contrast, results in Tables 5 and 6 show mitigated impact of dynamism as performance for a given algorithm barely change. However, entropy for a given genetic-based method slightly escalates with uncertainty conditioned by initially known visit request, suggesting that entropy is still inclined to increase with degree of dynamism. In counterpart, the proposed GA seems to adapt with growing dynamism to ultimately repair solutions and nearly match the quality observed in the static case. This is evidenced through

Table 5 Performance comparison and relative information gain of GA over Greedy for different degrees of dynamism

Degree of dynamism	Problem	Greedy			GA			RIG(T)
		Entropy	Lateness	Travel time	Entropy	Lateness	Travel time	
10%	R1	35.26	90.03	5548.03	23.03	4.72	5680.07	0.41
	RC1	37.02	54.81	5636.62	23.83	3.10	5776.07	0.46
30%	R1	35.47	9.08	5521.88	23.49	3.47	5670.68	0.41
	RC1	36.56	0.84	5565.72	23.93	2.48	5768.20	0.44
50%	R1	35.56	6.52	5478.45	23.54	6.55	5657.11	0.41
	RC1	35.89	2.04	5585.03	24.37	4.71	5765.83	0.41
70%	R1	35.40	5.29	5488.57	24.03	4.68	5658.19	0.40
	RC1	36.29	1.88	5576.27	24.84	7.23	5762.03	0.41
90%	R1	35.39	10.64	5466.61	24.17	12.66	5658.14	0.39
	RC1	36.60	4.18	5553.88	24.61	5.14	5763.61	0.43
Average		35.94	18.53	5542.11	23.98	5.47	5715.99	0.42

Table 6 Performance comparison and relative information gain of GA over GA_Myopic for different degrees of dynamism

Degree of dynamism	Problem	GA_Myopic			GA			RIG(T)
		Entropy	Lateness	Travel time	Entropy	Lateness	Travel time	
10%	R1	32.29	2.14	5594.58	23.03	4.72	5680.07	0.31
	RC1	33.56	1.26	5698.95	23.83	3.10	5776.07	0.34
30%	R1	32.90	0.86	5540.06	23.49	3.47	5670.68	0.32
	RC1	33.82	1.47	5655.43	23.93	2.48	5768.20	0.34
50%	R1	32.96	2.18	5502.70	23.54	6.55	5657.11	0.32
	RC1	33.73	1.72	5640.80	24.37	4.71	5765.83	0.33
70%	R1	33.13	2.61	5458.83	24.03	4.68	5658.19	0.32
	RC1	34.05	1.41	5650.81	24.84	7.23	5762.03	0.33
90%	R1	33.48	9.28	5477.92	24.17	12.66	5658.14	0.33
	RC1	34.32	8.54	5634.36	24.61	5.14	5763.61	0.34
Average		33.42	3.15	5585.44	23.98	5.47	5715.99	0.33

small RIG measure of performance variations. Exhibited fluctuations may be due to average performance compilation over problem instances. In other respect, recorded lateness is naturally coupled with uncertainty, as increasing rate of real-time visit request occurrence tend to degrade overall solution quality and further delay visits. It is also worth noticing that GA under maximum uncertainty (90%) still outperforms the greedy and GA_Myopic methods under minimum uncertainty (10%), reflecting the gain provided by forward-looking planning over myopic approaches. In fact, ignoring potential future visit requests associated with an unconfirmed target as promoted

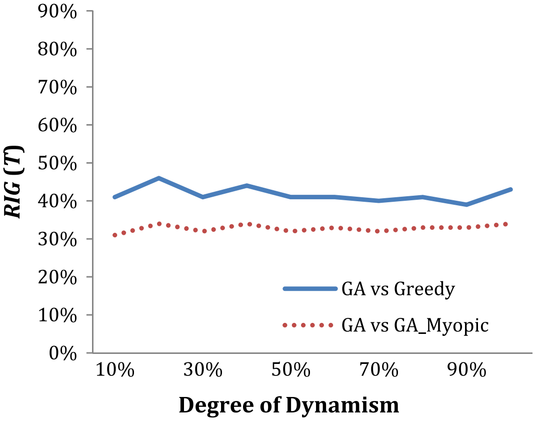

Fig. 4 Differential relative information gain $RIG(T)$ between two heuristics (Greedy and GA_Myopic) and the proposed genetic algorithm (GA) at the end of time horizon T for different degrees of dynamism over R1 and RC1 problem instances

by myopic approaches may lead to the impossibility of accepting certain zone visits which might have been accepted otherwise.

Differential relative information gains reported in Tables 5 and 6 comparing the proposed GA with the two examined methods over various degrees of dynamism (dynamic request distribution) are graphically represented in Fig. 4.

5 Conclusion

The proposed work consists in optimizing dynamic path planning to successfully search and detect multiple stationary targets. The developed approach relies on a new information-theoretic evolutionary algorithm to solve target search path planning. Problems having large time horizons may be adapted to a dynamic setting by repeatedly solving new static problem instances over a rolling horizon, incorporating observation outcomes from the last episode. The proposed approach takes advantage of objective function separability and conditional observation probability independence to hierarchically minimize expected system entropy, lateness and travel/discovery time respectively. A computational experiment has been carried out to conduct comparative performance analysis. The proposed GA approach turns out to be very efficient and competitive when compared to alternate heuristics such as myopic genetic algorithms and a greedy method as it clearly shows better results for all conducted simulations. Comparative performance over various degree of dynamism reflecting the proportion of requests dynamically generated for a given scenario, further indicates GA to outperform competing algorithms by more than 31% with respect to entropy, the first objective. This gain comes at the expense of less than a 3.5% travel time increase in comparison to the alternate methods.

Future work aims at naturally extending the current decision model to capture heterogeneous UAVs, and investigate its practical limitations. Alternate directions consist in adapting the approach to different search objectives such as proportion of target discovery or expected detection time minimization. Other challenges lie in modeling search problem variants incorporating more complex observation models and various target occupancy dependency and domain constraints, possibly infringing separability and symmetry assumptions. Multi-dimensional search problems involving complex domain knowledge modeled as belief networks represent another challenge as well.

References

Agmon N, Hazon N, Kaminka GA (2008) The giving tree: constructing trees for efficient offline and online multi-robot coverage. Ann Math Artif Intell 52:143–168

Ankenbrandt C (1991) An extension to the theory of convergence and a proof of the time complexity of genetic algorithms. In: Foundations of genetic algorithms. Morgan Kaufman, pp 53–68

Barkaoui M, Berger J, Boukhtouta A (2008) A hybrid genetic approach for the dynamic vehicle routing problem with time windows. Am J Math Manag Sci 28(1–2):131–154

Barkaoui M, Berger J, Boukhtouta A (2014) An information-theoretic-based evolutionary approach for the dynamic search path planning problem. In: International conference on advanced logistics and transport (ICALT), Hammamet, Tunisia

Barkaoui M, Berger J, Boukhtouta A (2015) Customer satisfaction in dynamic vehicle routing problem with time windows. J Appl Soft Comput 35:423–432

Bekhti M, Abdennebi M, Achir N, Boussetta K (2016) Path planning of unmanned aerial vehicles with terrestrial wireless network tracking. Wireless days, Toulouse

Berger J, Happe J, Gagne C, Lau M (2009) Co-evolutionary information gathering for a cooperative unmanned aerial vehicle team. In: 12th International conference on information fusion, FUSION ‘09

Berger J, Lo N, Barkaoui M (2016) Static target search path planning optimization with heterogeneous agents. Ann Oper Res 244(2):295–312

Botzheim J, Toda Y, Kubota N (2012) Bacterial memetic algorithm for offline path planning of mobile robots. Memetic Computing 4(1):73–86

Bourgault F, Furukawa T, Durrant-Whyte HF (2003) Optimal search for a lost target in a bayesian world. In: Proceedings of the 4th international conference on field and service robotics (FSR’03), 24, Lake Yamanaka, Japan, pp 209–222

Bourgault F, Furukawa T, Durrant-Whyte HF (2003b) Coordinated decentralized search for a lost target in a Bayesian world. IEEE/RSJ Int Conf Intell Robots Syst 1:48–53

Brooks A, Makarenko A, Williams S, Durrant-Whyte H (2006) Parametric POMDPs for planning in continuous state spaces. Robot Auton Syst 54(11):887–897

Chia SH, Su KL, Guo JH, Chung CY (2010) Ant colony system based mobile robot path planning. In: 4th International conference on genetic and evolutionary computing, Shenzhen, China

Cover T, Thomas J (2006) Elements of information-theory, 2nd edn. Wiley, Hoboken

Dai R, Cochran JE (2009) Path planning for multiple unmanned aerial vehicles by parameterized cornu-spirals. In: Conference paper in proceedings of the American control conference, pp 2391–2396

Freundlich C, Mordohai P, Zavlanos MM (2015) Optimal path planning and resource allocation for active target localization. In: American control conference, Chicago, IL, USA, 2015, pp 3088–3093

Galceran Enric, Carreras Marc (2013) A survey on coverage path planning for robotics. Robot Auton Syst 61(12):1258–1276

Hrabar S (2008) 3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs. In: IEEE/RSJ international conference on intelligent robots and systems, nice, France, 22–26 Sept 2008, pp 807–814

Lanillos P, Besada-Portas E, Pajares G, Ruz JJ (2012) Minimum time search for lost targets using cross entropy optimization. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 602–609

Lanillos P, Zuluaga JY, Ruz J, Besada-Portas E (2013) A Bayesian approach for constrained multi-agent minimum time search in uncertain dynamic domains. In: Proceedings of the 15th annual conference on genetic and evolutionary computation. ACM, pp 391–398

Lanillos P, Gan SK, Besada-Portas E, Pajares G, Sukkarieh S (2014) Multi-UAV target search using decentralized gradient-based negotiation with expected observation. In: Information sciences, 282, 2014, pp 92–110

Larsen A, Madsen O, Solomon M (2002) Partially dynamic vehicle routing-models and algorithms. J Oper Res Soc 53:637–646

Lau H (2007) Optimal search in structured environments, PhD thesis, University of Technology, Sydney

Lau H, Dissanayake G (2005) Optimal search for multiple targets in a built environment. In: Proceedings of the IEEE/RSJ international conference intelligent robots and systems

Lin Y, Saripalli S (2014) Path planning using 3D dubins curve for unmanned aerial vehicles. In: Proceedings of the IEEE international conference on unmanned aircraft systems (ICUAS), Orlando, FL, USA, 27–30 May 2014, pp 296–304

Liu FH, Shen SY (1999) A route-neighborhood-based metaheuristic for vehicle routing problem with time windows. Eur J Oper Res 118:485–504

Lo N, Berger J, Noel M (2012) Toward optimizing static target search path planning. In: IEEE symposium on computational intelligence for security and defence applications, Ottawa, Canada, pp 1–7

Mac TT, Copot C, Tran DT, De Keyser R (2016) Heuristic approaches in robot path planning: a survey. Robot Auton Syst 86(12):13–28

Osman IH (1993) Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Ann Oper Res 41:421–451

Otte MW (2017) A survey of machine learning approaches to robotic path-planning. University of Colorado at Boulder, Report, August 2017, p 93. https://www.cs.colorado.edu/~mozer/Teaching/Computational%20Modeling%20Prelim/Otte.pdf

Peng Xingguang, Demin Xu (2012) Intelligent online path planning for UAVs in adversarial environments. Int J Adv Robot Syst 9:1–12

Perez-Carabaza S, Besada-Portas E, Lopez-Orozco JA, de la Cruz JM (2016) A real world multi-uav evolutionary planner for minimum time target detection. In: Proceedings of the genetic and evolutionary computation conference. ACM, pp 981–988

Pettersson PO, Doherty P (2006) Probabilistic roadmap based path planning for an autonomous unmanned helicopter. J Intell Fuzzy Syst 17:395–405

Rekleitis I, New AP, Rankin ES, Choset H (2008) Efficient boustrophedon multi-robot coverage: an algorithmic approach. Ann Math Artif Intell 52:109–142

Rylander B, Foster J (2001) Computational complexity and genetic algorithms. In: Proceedings of the world science and engineering society's conference on soft computing, advances in fuzzy systems and evolutionary computation. World Science and Engineering Society Press, pp 248–253

Rylander B, Soule T, Foster J (2001) Computational complexity, genetic programming, and implications. In: Proceedings of the European genetic programming conference

Seuken S, Zilberstein S (2008) Formal models and algorithms for decentralized decision making under uncertainty. Auton Agent Multi-Agent Syst 17(2):190–250

Shaw P (1998) Using constraint programming and local search methods to solve vehicle routing problems. In: Maher M, Puget J-F (eds) Principles and practice of constraint programming. Lecture Notes in Computer Science. Springer, New York, pp 417–431

Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res 35:254–265

Stone LD (1975) Theory of optimal search. Academic Press, New York

Sun Chuangchuang, Liu Yen-Chen, Dai Ran, Grymin David (2017) Two approaches for path planning of unmanned aerial vehicles with avoidance zones. J Guid Control Dyn 40(8):2076–2083

Svennebring J, Koenig S (2004) Building terrain-covering ant robots: a feasibility study. Auton Robots 16(3):313–332

Tisdale J, Zuwhan K, Hedrick JK (2009) Autonomous UAV path planning and Estimation: an online path planning framework for cooperative search and localization. IEEE Robot Autom Mag 16(2):35–42

Vidal R, Shakernia O, Kim HJ, Shim DH, Sastry S (2002) Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. IEEE Trans Robot Autom 18(5):662–669

Washburn AR (1998) Branch and bound methods for a search problem. Nav Res Logist 45:243–257

Wong S, MacDonald B (2003) A topological coverage algorithm for mobile robots. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, Las Vegas, 2003, pp 1685–1690

Wood J, Hedrick K (2011) Multi-agent path planning for an unknown number of targets over dynamic space partitions. In: 50th IEEE conference on decision and control and european control conference, Orlando, FL, USA, 2011, pp 564–569

Yang S, Luo C (2004) A neural network approach to complete coverage path planning. IEEE Trans Syst Man Cybern B Cybern 34(1):718–724

Yang Y, Minai A, Polycarpou M (2005) Evidential map building approaches for multi-UAV cooperative search. In: Proceedings of the American control conference

Yu H, Meier K, Argyle M, Beard RW (2015) Cooperative path planning for target tracking in Urban environments using unmanned air and ground vehicles. IEEE/ASME Trans Mechatron 20(2):541–552

Yuan S, Lau H, Liu DK, Huang SD, Dissanayake G, Pagac D, Pratley T (2009) Simultaneous dynamic scheduling and collision-free path planning for multiple autonomous vehicles. In: International conference on information and automation, Macau, China, 2009