Coordinating a Team of Searchers

Of Ants, Swarms, and Slime Molds

Silja Meyer-Nieberg
Fakultät für Informatik, Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
Neubiberg 85579
silja.meyer-nieberg@unibw.de

ABSTRACT

This paper introduces new hybrid coordination techniques for searching mobile targets in dangerous and dynamic environments. Situations like this arise for example during and after natural or human caused disasters. The approach presented combines algorithmic concepts from ant colony and particle swarm optimization with an adaptive network approach that utilizes principles stemming from slime-molds. The objectives are to achieve a sufficiently wide distribution of the search team in order to cover and explore the area, to send the searchers towards the suspected position, and to safeguard against passing unsafe areas. In addition, the technique proposed can react fast towards environmental changes. Two main algorithms were developed. They are investigated in a series of experiments showing promising first results.

CCS CONCEPTS

• **Computing methodologies → Multi-agent planning**; *Randomized search*;

KEYWORDS

swarm intelligence, search, mobile targets, uncertainty

ACM Reference format:
Silja Meyer-Nieberg. 2017. Coordinating a Team of Searchers. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
https://doi.org/http://dx.doi.org/10.1145/3071178.3071277

1 INTRODUCTION

The search for mobile targets with a team of robots has received more and more attention. With the help of robots or more generally with the help of autonomous vehicles, search tasks can be performed in hazardous environments unsuited for human operators. Applications can be found in the civilian sector as e.g. for search-and-rescue missions and for military or security related operations.

In order to combine information from multiple sources, multi-sensor data fusion techniques are applied [15]. The gathering and merging of dynamic and uncertain information is of considerable importance in order to make reliable predictions regarding e.g. the position of a drifting vessel at sea or of a person in a area affected by an earthquake. But how can the information obtained be used to guide the searchers towards better positions in order to locate the target? How can we coordinate the search itself? This paper addresses the questions above and focuses on a search for a mobile target in dynamic and hazardous environments. We present a novel hybrid approach combining methods stemming from several fields of natural computing. The approach is an extension of the method introduced in [14].

The task described is considered in several research fields, e.g. in the area of swarm robotics [19]. It also belongs to the area of search and pursuit-evasion games which have a long research tradition [3]. To solve the task several techniques have been applied. They range from stochastic heuristics [8] to specialized branch-and-bound optimization approaches [18].

The paper is structured as follows. First, the swarm of searchers and the task are described. Afterwards, we provide details concerning the model and the assumptions used. The next section is concerned with the algorithm and introduces its procedures in detail. The resulting approach is investigated in a series of experiments which aim to provide more insights into its potential advantages and weaknesses.

1.1 Searching with Swarms

A team of K autonomous vehicles $s_1, \ldots, s_K$ or robots (called swarm in the following) is employed for a search task. The objective is to locate a moving target the position of which is uncertain. The search can be divided into several phases depending on the amount of available information concerning the targets and on the location of the searchers w.r.t. the target. If no information concerning potential target positions is obtainable at present, the searchers must cover the area. To this end, terrain covering techniques that stem from the search for stationary targets can be applied. If the searchers are able to obtain measurements, a likelihood function for the target location can be derived and used in turn to guide the search. Regions with sufficiently large likelihood values should be prioritized. If they are in the vicinity of a searcher, they should be inspected first, otherwise, a robot should move towards the search regions with the largest likelihood of containing the target. To do so, the searcher should follow a swift and safe path as it can be obtained by applying the dynamic network approach introduced later.

The swarm members are assumed to be equipped with sensors and communication devices. As it is common in literature on swarm

ACM ISBN 978-1-4503-4920-8/17/07…$15.00
https://doi.org/http://dx.doi.org/10.1145/3071178.3071277

robotics, we assume omni-directional sensors as well as an omni-directional communication. The communication devices may have a limited radius and may also be impaired by environmental conditions. The same holds for the sensors. The communication radius is denoted by $r_c > 0$. In the following, we say that searcher s_k is in the communication neighborhood of searcher s_i, i.e. $s_k \in \mathcal{CN}_i$, if they are able transfer data directly, that is, if the distance of their positions p_{s_k} and p_{s_i} is smaller than the radius, i.e., $\|p_{s_k} - p_{s_i}\| \leq r_c$. When a searcher is inside the communication range of another robot, data is exchanged and measurements are combined. The swarm operates with dynamic neighborhood topologies depending on the cohesion of the swarm with respect to the communication range. For this reason, the neighborhood structure may vary from fully connected topologies to extremely sparse structures.

1.2 Of Likelihoods and Risks

In the following, several definitions are introduced that are required in the remainder of the paper. The environment the searchers operate in is called the *search space* $\mathcal{S}$ with $\mathcal{S} \subset \mathbb{R}^2$. For the algorithm, it is modeled in a first step as a two-dimensional area or map $C = \{c_i | i = 1, \ldots, N\} \subset \mathbb{R}^2$ which is discretized into cells c_i of constant size and regular form. The cell type depends on the characteristics of the robots, i.e., on their speed, on their sensing radius, and on their maneuverability. For each cell c_i the *neighborhood environment* $\mathcal{N}_i$ contains all other cells that can be reached directly from c_i. The target is found, if it occupies the same cell as a searcher. The estimate of the target positions $\mu(\mathbf{x}, t)$ can be aggregated to provide the general likelihood μ_i^t for the cell containing the object of the search at time t, e.g. $\mu_i^t = \max_{\mathbf{x} \in c_i} \mu(\mathbf{x}, t)$. In order to ensure a stable behavior of the algorithm that is introduced later, a minimal value $\epsilon_\mu > 0$ is introduced. The spatial representation of the $\mathcal{M}_\mu^t := (\mu_i^t)_{i=1,\ldots,N}$ will be referred to as a *likelihood map*. If it pertains to a single searcher s_k, it is denoted as $\mathcal{M}_\mu^t(s_k)$. The coordination technique introduced will be based on normalized values, i.e., $\mu_i^t \in [0, 1]$ holds. Since exchanging and combining information requires the original values, the minimal and maximal values used in the transformation must be stored. Similarly, a risk value ρ_i^t for the cell c_i can be derived. Following a risk averse approach, the maximal value inside the cell $\rho_i^t = \max_{\mathbf{x} \in c_i} \rho(\mathbf{x}, t)$ is utilized. Here, $\rho(\mathbf{x}, t)$ denotes the estimate of the risk. Once the values are obtained, they are normalized so that $\rho_i^t \in [0, 1]$. The associated *spatial risk map* is denoted as $\mathcal{R}^t$ with $\mathcal{R}^t(s_k)$ standing for the distinct set of searcher s_k. In contrast, the *safety of passage* $\eta_i^t := 1 - \rho_i^t$ indicates the preference of a cell due to its low risk.

Using the measures introduced above, the concept of a neighborhood is refined by considering the *admissible neighborhood* of a cell c_i which is denoted as all directly reachable cells that a searcher may pass in safety, i.e., where the safety exceeds a predefined threshold $\eta_{\min} > 0$ as $\mathcal{AN}_i^t = \{c_j | c_j \in \mathcal{N}_i \wedge \eta_i^t \geq \eta_{\min}\}$. Areas with high risk values are termed *unsafe regions* $\mathcal{X}^t = \{c_i | \eta_i^t < \eta_{\min}\}$. The minimal values $\eta_{\min}$ and ϵ_μ are also used to initialize the cell values when no information is available. In order for the swarm to cover the search region, the current area of interest must be divided among the swarm members. Several coordination methods can be applied for this task including e.g. auctioning. This paper presents an approach based on particle repulsion.

1.3 From Maps to Graphs

For the path planning, the search space $\mathcal{S}$, or more correctly, the map C, is represented by a graph $G = (V, E)$, with $V = \{v_1, \ldots, v_N\}$ denoting the set of nodes and E the set of edges. The cell structure can be transformed into graph form in several manners, here, we let the nodes v_i stand for the centers of the cells c_i whereas the edges of the network result from the neighborhood structure in other words $e_{ij} \in E \Leftrightarrow c_j \in \mathcal{N}_i$. The edges are undirected and weighted with costs $d_{ij} := \|v_i - v_j\| > 0$ standing for the Euclidean distance between the centers v_i and v_j. In a slight misuse of the notation we say that v_j is in the neighborhood of v_i and denote it by $v_j \in \mathcal{N}_i$ if $c_j \in \mathcal{N}_i$. The same holds for $v_j \in \mathcal{AN}_i^t$. The positions of the searchers $s_1, \ldots, s_K$ are denoted by $p_{s_k}^t$ and defined for $p_{s_k}^t \in V$, i.e., we do not update the position on the edges. Concerning the target and its estimated location, we define $\mathcal{M}_{\max}^t = \{c_j | j \in \arg\max_i \mu_i^t\}$ as the cells and $\mathcal{V}_{\max}^t = \{v_j | j \in \arg\max_i \mu_i^t\}$ as the nodes with the largest values or in other words the best guesses for the current target location. Again, each searcher may have its distinct parameter set. Let $v \in V$ be a node in the graph. The closest maximal point is then given by $p_{Mk}^t(v) \in \mathcal{M}_{\max}^t := \arg\min_{u \in \mathcal{M}_{\max}^t} \|u - v\|$. For a searcher we have the same definition, i.e., $p_{Mk}^t(s_k) \in \mathcal{M}_{\max}^t := \arg\min_{u \in \mathcal{M}_{\max}^t} \|u - p_{s_k}^t\|$ denotes the closest best guess for the searcher s_k. Without loss of generality, we assume that there exists only a single point. Otherwise, the tie would be broken by selecting it at random.

2 COMBINING SWARMS AND SLIME-MOLDS

This section introduces the hybrid swarm coordination which is realized as an online approach. It uses an adaptive network based on slime-molds to derive fast and safe paths towards the assumed target position. The resulting path lengths is then considered in the movement planning for the searchers. The decisions in each iteration are based on several factors. They are described in detail in this section.

2.1 Path Planning in Hazardous Environments

Path planning for autonomous vehicles or robots has a long research tradition. In general, offline and online techniques are distinguished. While the first class uses information available beforehand to generate a path, the second takes place when the searcher is already in motion and can adapt to situational changes making it more suitable for dynamic and uncertain environments. Since this equals the application scenario considered, we adopt also an online approach. In the following, two examples are briefly described. Wen et al. [22] consider for example online planning for unmanned arial vehicles operating in uncertain and dangerous environments. They apply several methods resulting in a rapidly exploring random tree based planning algorithm. Ok et al. [16] also used a combination of approaches or more exactly planners in their Voronoi Uncertainty Fields. The higher level planner computes Voronoi diagrams which it uses to determine a connected path. The path is then corrected based on uncertainty estimations with the help of artificial potential fields. The techniques proposed, also applies a combination of several techniques or a two-stage process. We use a path finding algorithm in order to identify suitable pathways to the current target position. The information gained then enters the second phase of the actual movement planning. The path planning considers several objectives simultaneously: While short paths to the target are important, their safety carries also considerable weight. Both have to be taken into account. The approach presented is based on adaptive networks or on slime mold optimization. *Slime mold optimization* (SMO) represents a relatively new optimization technique based on models of the tube dynamics of real slime molds [21]. It can be applied to network flow and linear programming problems and has been used among others for routing and transportation, see e.g. [10]. Typically, the slime-mold covers the complete graph. In the case of large graphs, problems may appear: SMO approaches consider an artificial nutrient flow from sources to targets and adapt their structure accordingly by shrinking connections that are not profitable with respect to the nutrient transport. For this reason, [12] introduced a dynamic, adaptive network with two main phases: growth or re-growth and tube dynamics. The first ensures that start and target node for the path planning are connected and provides a sufficiently large network structure so that appropriate parts of the graph are covered. Once the subgoal is established, the second phase, the tube dynamics, is started. During this process, the adaptive network withdraws from edges that impair the solution quality w.r.t. an optimal path to the target until the network reaches a stable steady state.

As stated, instead of covering the complete graph, the adaptive network approach grows from start (or sink) to target (source). While this may result in non-optimal solutions, the smaller size of the resulting structure improves the efficiency of the tube dynamics. The process, however, requires information in order to bias the growth towards the region and the point of interest. In [12] several indicator functions were introduced. They provide a kind of potential field or –a nutrient gradient– followed by the slime mold. Here, we consider an indicator function that depends on the Euclidean distance $d_e(v_i, s)$ to the source s and on the immediate safety of passage at the node $\mathcal{H}(v_i) = (\eta_i^{\kappa})/((1 + d_e(v_i, s))^{\delta})$ with the two parameters δ and κ controlling the influence of the objectives [12]. The topology of the initial network that results from the growth process depends on the choice of nodes from which the adaptive network continues to grow and on the nodes selected that will be newly covered. Concerning the selection of the growth nodes, two strategies were introduced in [12]: full and partial. In the case of the former, the network grows from all outer nodes, i.e., from all network nodes that have at least one neighbor not already covered. A sparser structure is assumed when only a subset may be used as it is the case for the partial strategy. Currently, the subset is selected deterministically taking the indicator function values into account. Once the growth nodes have been selected, it has to be determined which of the free neighbors will be covered by the adaptive network. Again, two strategies have been implemented which influence the topology of the initial structure: a greedy approach where a connection is only established with nodes where the indicator function attains maximal values and a so-called above average method which takes all vertices into account where the indicator function assumes values larger than or equal to the neighborhood average [12]. Once sink and source are connected, the tube dynamic phase replaces the growth process and the flow of the artificial nutrients begins. In SMO, the process is modelled with the help of Poiseulle flows where the flow Q_{ij} along the edge e_{ij} is proportional to the pressure difference $p_i - p_j$ at the two nodes v_i and v_j connected. More information can be found e.g. in [12]. The resulting network represents either a single path solution or a multiple path variant. Which is obtained depends on the control parameter setting of the SMO, see the discussion in [12]. In the present context, the focus lies on single path solutions since the path length is used for further decision making. Therefore, we also decided on using the partial approach concerning the growth node selection combined with the greedy approach for finding the newly covered vertices. It should be mentioned that the adaptive network approach reacts to significant situational changes. If for example the risk values for a covered edge increases above the limit allowed, the slime-mold withdraws from it. In the case that the connectivity between source and sink is lost, a re-growth phase is initiated until it is reestablished. For more information, see [12].

2.2 Balancing Risk and Local Likelihood

Once the paths have been determined, the online movement planning can be started. It is based on stochastic decision making. Each searcher takes information stemming from itself and from other robots into account in order to select a new node from the feasible neighbors: Consider the situation that a searcher has reached a node v_i with $\mu_l > \epsilon_\mu$ for at least some $v_l \in \mathcal{AN}_i^t$. We assume that in general the likelihood map provides useful information although it may be distorted by measurement errors. Cells with large likelihood values therefore have a high probability of containing the target on average. Therefore, a robot should take the likelihood values in the neighborhood into account. But the information may also be misleading and the movement may not contribute to the ultimate goal. For this reason, stochastic effects are introduced. The basic equation for the selection probability of a node v_j of the admissible neighbors $\mathcal{AN}_i^t$ reads

$$p(v_j, s_k, t) \quad = \quad \frac{(\mu_j^t)^\alpha (\eta_j^t)^\beta}{\sum_{v_k \in \mathcal{AN}_i^t} (\mu_k^t)^\alpha (\eta_k^t)^\beta} \tag{1}$$

with parameters $\alpha, \beta \geq 0$. With their help, the influence of safety and the most promising neighboring field can be balanced. If the admissible nodes do not differ in their likelihood values, (1) leads to a stochastic risk aversion. Equation (1) represents a variant of the central probabilistic decision rule of ant colony optimization (ACO), a well known stochastic metaheuristics [5]. Ant colony optimization is often used in combinatorial optimization, to tackle e.g. routing or scheduling tasks. It takes its inspiration from mathematical models of search behavior of ant colonies. Instead of using pheromone traces as it would be the case in ACO, the approach presented is based on problem specific values since the focus lies on an online mechanism which plans one step ahead and no complete path is constructed for the swarm in the planning stage.

2.3 From Local to Global Information

Equation (1) is a variant of a stochastic greedy search heuristic. Since it considers local information, it will tend to guide the searchers towards the best local areas. This represents a myopic approach and a potential problem. The algorithm may be unable to identify preferable regions in the search space. Therefore, global information as obtained by the sensor measurements is utilized and the searcher's decision is biased with respect to the current best solution which represents the current best estimate for the target's position. The approach presented is based on particle swarm optimization where a particle has a certain tendency to move into the direction of the best member of the swarm. Here, the searchers do not move towards the robot that resides in a cell or in a node with the largest function value but towards the best guess of the target location. This best guess may differ if swarm members are outside the communication area. Particle swarm optimization (PSO) [6] has been introduced in the 1990s. Today, it represents an established metaheuristic for continuous optimization. It operates with a swarm of particles. The velocity vector of the particles is the central element of the search. It is continuously adjusted during a run by taking the search history into account. Here, we strive to increase the selection probability of promising directions. These directions are defined by their distance to the current best estimates for the target positions. They do not need to coincide with any actual searcher position. Instead they represent the cells or the nodes with the largest likelihood, i.e., $p^t_{Mk}(v)$ for a node v in the admissible neighborhood of searcher s_k. Furthermore, let $d^t_{SMO}(v)$ denote the distance as obtained by the adaptive network approach. The shorter the safe path is that begins at the node, the more preferable is a selection. This needs to be respected by a suitable transformation of the distance into $[0, 1]$. A potential choice reads $f(x) = 1/(1 + x)$ with $x \geq 0$. Augmenting the stochastic rule (1) with $f^t_j := f(d^t_{SMO}(v_j))$ leads to

$$p(v_j, s_k, t) \quad = \quad \frac{(\mu^t_j)^\alpha (\eta^t_j)^\beta (f^t_j)^\gamma}{\sum_{v_k \in \mathcal{AN}^t_i} (\mu^t_k)^\alpha (\eta^t_k)^\beta (f^t_k)^\gamma} \tag{2}$$

with parameters α, β, and $\gamma \geq 0$. So far, the presence of multiple searchers had only indirect influence. If parts of the group are able to communicate they operate with the same situational awareness and therefore with shared likelihood and risk maps. No mechanism has been introduced, however, which can be used to achieve an appropriate spreading of the swarm in the search space. If the swarm distribution collapses and all searchers follow the same path the potential benefits that arise with multiple searchers diminish. While it is easier to replace units that malfunction or are destroyed if the replacements are nearby, in general resources are wasted if the swarm is concentrated in only small parts of the whole space. This concerns especially a dynamic search with uncertain information. Therefore, the next section introduces an approach to distribute the searchers in the space.

2.4 Spreading the Search Team

The estimation function results from measurements of the target position. Therefore, the region or the cells with the highest values can be assumed to be the current best estimate. However, in addition to noisy disturbances, the function values are subject to change since the target is mobile. The swarm must be able to track the changes and must be kept from collapsing into a single point. To this end, the concept of particle repulsion is integrated. Particle repulsion stems from charged swarms [1], a particle swarm optimization technique developed for dynamic optimization. Charged swarms maintain or even increase the population diversity by allowing parts of the swarm to repel another. This serves to counteract the natural tendency of the swarm to converge into a single point in the search space. In the case of PSO, the extend of the repulsion depends on the load carried by the particles and the distance (vector) of the particles to each other. In the case that the distance of two particles is smaller than a predefined radius, referred here as $d_{i_0} > 0$, the particles are repelled. They move into opposite directions along the line defined by the distance vector. The extend of the repulsion is proportional to the charge and reciprocally proportional to the square of the distance. To safeguard against vast and sudden position changes which are usually detrimental, the repulsion magnitude is bounded by introducing a minimal distance. Here, the approach is extended: First of all, let $\mathcal{I}_k := \{s_l \| \|p_{s_l} - p_{s_k}\| \leq d_{i_0}\}$ be the subset of searchers that are too close to the searcher s_k. They (and s_k) will be "repelled". In contrast to particle swarm optimization which considers continuous search spaces and operates with distance vectors, the coordination mechanism presented here is based on graph representations. Therefore, the searchers are not repelled in the strict sense. In each iteration, the searcher may only choose one of the nodes in the neighborhood. Therefore, we implement a stochastic bias – in other words, the searcher evades the others as far it is possible. The question of how close is too close is addressed dynamically. To this end, let us define the expected degree of target approach of a searcher s_k as

$$d^a(s_k, t) := \frac{d(s_k, t)}{\text{path}(s_k, t) + d(s_k, t)} \tag{3}$$

with $\text{path}(s_k, t)$ denoting the distance traveled so far and $d(s_k, t)$ the current safe distance to the most likely target position. The parameter is introduced in order to control the particle repulsion. The intent is that the swarm spreads out and covers a wide search region especially in the beginning of the search when only a few measurements have been taken, the information available is highly uncertain, and the units are potentially far away from the target position. The more time passes and the closer the searchers approach the target, the more and more the information can be trusted. For this reason, the threshold of the onset of the repulsion mechanism decreases with expected degree of target approach. Furthermore, it takes into account how close the searcher is to its particular best guest relative to the respective distances of the other searchers in the neighborhood

$$d_I(s_k, t) = d_{i_0}\,(d^a(s_k, t)) \left(\frac{\sum_{l \in \mathcal{I}_k} d(s_k, t)}{\max\left(d_{\min}, d(s_k, t)\right)} \right). \tag{4}$$

The bias against nodes that bring the searcher too close to others depends on the respective distances. For a given node $v_j \in \mathcal{AN}^t_i$ and a searcher $s_l \in \mathcal{I}_k$, they are provided by

$$\tilde{d}(v_j, p_{s_l}, t) = \begin{cases} \infty & d(v_j, p_{s_l}) > d_I(s_k, t) \\ d(v_j, p_{s_l}) & d_{\min} \leq d(v_j, p_{s_l}) \leq d_I(s_k, t) \\ d_{\min} & d_{\min} \geq d(v_j, p_{s_l}) \end{cases} . \tag{5}$$

To determine the overall repulsion for the node, the distances are first transformed so that smaller distances result in larger contributions than larger. Here, similar expressions as before can be used. The aggregated transformed distance for a node v_j and a searcher

s_k is obtained as

$$d(v_j, s_k, t) = \sum_{s_j \in \mathcal{I}_k : s_l \neq s_k} t(\tilde{d}(v_j, s_l)). \quad (6)$$

The larger the repulsion (6) of a node, the smaller should the selection probability be. The final repulsion factor can then be derived for example as $r_j^t = 1/(1 + \tilde{d}(v_j, p_{s_l}, t))$ and represents the last factor for the probabilistic decision rule that is applied

$$p(v_j, s_k, t) = \frac{(\mu_j^t)^\alpha (\eta_j^t)^\beta (f_j^t)^\gamma (r_j^t)^\delta}{\sum_{v_k \in \mathcal{AN}_i^t} (\mu_k^t)^\alpha (\eta_k^t)^\beta (f_k^t)^\gamma (r_k^t)^\delta} \quad (7)$$

with four control parameters α, β, γ, and $\delta \geq 0$. The algorithm using (7) is called *simple probabilistic search* (SPS). Suitable settings for the control parameters need to be determined experimentally. The question of robust, optimal, or critical parameter settings will be addressed in Section 3 with the help of data farming experiments.

2.5 Exploiting the Information

Additional algorithms are based on an efficient ant colony optimization variant: ant colony system (ACS)[4]. Ant colony system represents one of the best ACO-methods with respect to solution quality and efficiency [5]. Most of the other ACO algorithms make use of a similar stochastic rule to Eq. (7). Ant colony system employs a different principle: Whenever a decision is to be made, ACS first determines whether to use the random rule (7) or whether to proceed deterministically. In the case of the latter, the next node is selected as the best choice in the neighborhood. For the swarm search considered, this equals taking

$$v_h = \arg \max_{v_j \in \mathcal{AN}_i^t} p(v_j, s_k, t) \quad (8)$$

as the next node to move to. The switch between random and deterministic rule is made at random following a uniform distribution. The probability $p_{SPS} > 0$ for choosing (7) instead of (8) does not need to remain constant. Instead it may move from exploration with (7) towards exploitation with (8). For this, we make use of (3) which lowers the potential for exploration with the expected approach towards the target with

$$p_{ASPS}(s_k, t) = d^a(s_k, t) p_{SPS}. \quad (9)$$

Equation (4) does not necessarily represent a monotonously decreasing function with time. The resulting algorithm is termed *adaptive exploiting probabilistic search* (AEPS) whereas the static version will be denoted as *exploiting probabilistic search* (EPS).

3 DATA FARMING EXPERIMENTS

This section describes the experiments that were conducted to analyze the performance of the algorithms. Since we aim at a proof of concept, the paper addresses the question whether AEPS or EPS lead to promising results that merits further investigations and comparisons with other online approaches. We focus on the latter two since preliminary experiments showed that their performance is better than that of SPS. For results concerning the performance of the first variants of EPS and SPS on square grids, the reader is referred to [14] where also a comparison with random search was conducted.

First, details concerning the methodological approach, data farming, are provided, before the performance measures used are introduced. Afterwards, the experimental set-up, that is, the implementation choices made and the parameter settings is given. The section concludes with a discussion of the results of the experiments for hexagonal grids.

Data Farming [2] represents an iterative process which strives to derive more information concerning potential model or system behaviors especially w.r.t. the impact of control parameters. In its basic form, it is strongly related to the design and analysis of computer experiments (DACE) [17] and the design and analysis of simulation experiments (DASE) [11]. All approaches introduce methods by which the parameter space of the model can be explored to gain more insights and information w.r.t. the model response. The parameter space is screened with the help of experimental designs which provide the value combinations for the simulation. The design type depends on the particular analysis and its focus. Up to now, a multitude of variants has been introduced [17]. Here, a space-filling design is applied since the focus lies on an investigative screening of the control parameter space in order to identify critical combinations which may also indicate weaknesses of the method. The particular design used is a nearly orthogonal Latin hypercube (NOLH) design which represents a compromise between a space-filling latin hypercube and the additional criterion of low correlations between the variables [9].

3.1 Experimental Set-Up

The methods, AEPS and EPS were implemented in Java and integrated into MASON [13], an open source multi-agent discrete event simulator, which is developed by the George Mason University. Additionally, GeoMason [20], a geo-spatial extension which allows to include coverages and grids has been used. In the following, the model properties and the assumptions made are discussed. The searchers are spawned randomly and start from the same node. They have the same velocity as the target which is set to $v = 1$. In the experiments, the target follows a Markovian motion model. The next position of the center c_{target} is chosen at random using a uniform distribution around the old location in both dimensions. Once the new position in the plane has been determined, it is mapped to the nearest node. The positional uncertainty is modeled as an ellipsoid around the center. The experiments consider dynamically changing hazardous areas as they appear e.g. during wildfires, floods, or earthquakes. The area is continuous and represented by an ellipsoid. The length, width, and the rotation angle are adjustable but do not vary during a simulation run. The movement of the centroid c_{risk} follows again a Markov process with independent uniform random distributions in both dimensions. The risk area is spawned close to the target uncertainty area so that high risk situations with the risk area overlapping the target position occur with a high probability. Several performance measures are used. We are interested in the first hitting time (FHT) of a run, the number of target hits (NHT), the number of incidents (NI), and of course in the duration of the incidents (DI). The experimental analysis considers the following environmental and control parameters: the number of searchers N, the probability p_{SPS}, the minimal distance $d_{\min}$, the height, width, and angle of the target area, the height, width, and angle of the unsafe area, the frequency of the risk area changes,the control parameters α, β, γ, δ, the scaling factor d_{i_0} of the influential distance, and the minimal distance $d_{\min}$. Their impact on the performance is assessed for two strategies EPS and AEPS. The searchers start at the same randomly initialized position whereas the target field center and the center of the disturbance start in the same points with the target center being located in the middle of the lower quarter of the simulation area with the disturbance field blocking its upper left area. The simulation is carried out for hexagonal grids. The grid size is set to $K = 900$ nodes. As stated, the disturbance field is modeled as a Markov process allowing independent uniform movements in the plane. For the experiments, the maximal movement concerning the x-coordinate is set to [-10,10] w.r.t. to the simulation area size whereas it reads [-20,20] for the y coordinate. The hazardous area is thus able to shift suddenly and rather widely over the search region in order to model a hazardous search scenario which may arise e.g. during wildfires. The NOLH design chosen results in 63 data points, each with 30 repeats resulting in 1890 runs for the statistical analysis.

3.2 Results for Hexagonal Grids

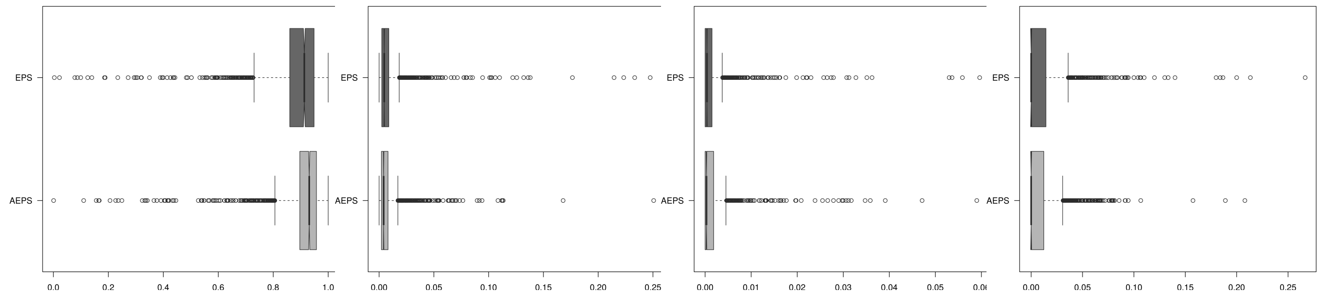
The strategies adaptive exploiting probabilistic search (AEPS) and exploiting probabilistic search (EPS) lead to a similar performance on hexagonal grids. The total success rate is one for both, the incident rate of AEPS is 0.562, whereas it reads 0.617 for EPS. This would indicate a slight performance benefit for AEPS. However, the more detailed measures have to be considered. First of all, we need to note that not all results enter the statistical analysis. In the case of AEPS 40 runs had to be discarded due to no data probably because the searchers where initialized inside the hazardous area. In the case of EPS 44 runs were not used. To summarize, if a statistical test, i.e., the Wilcoxon-Mann-Whitney test, see e.g. [7], is performed it indicates significant statistical differences at the 0.05 level. Only in the case of the number of incidents, the zero hypothesis cannot be discarded. However, it should be noted that the number of runs is quite large.

Our detailed investigation starts with the box-plots of the four measures. Figure 1 provides the results for the relative performance measures normalized with the number of searchers and the simulation time. Concerning the number of target hits, the adaptive exploiting search seems to have advantages which also transfers to the time until the target is found for the first time. In the latter case, several outliers are present. The swarm finds the target quite early in the majority of the experiments. The distribution of the outliers covers a very large range, however. This is underlined by the histograms in Figures 2 and 3 which hint at skewed, heavy-tail distributions. The decision trees suggest an influence of the disturbance field. As the figures indicate the number of searchers appears as an important factor for the time that the target is first found. Apparently, the swarm sizes larger than four allow to locate the target early. Therefore, very small swarm sizes are discouraged. Additional factors concern the characteristics of the disturbance field at least in the case of EPS. These affect only small swarms with less than five members, however. Since the number of data points for these parts of the decision tree is low, the number of experiments should be increased in order to gain more insights concerning swarm sizes of three and four. Small swarms, however, are extremely sensitive to a loss or failure of a sensor platform. For this reason, we do not investigate their characteristics further. Our current hypothesis is that small, cohesive swarms may be susceptible to disturbances by either a large or by a frequently changing disturbance field.

The number of target hits varies strongly, see Figs 2 and 3. The histograms in Figs. 2 and 3 show heavy-tailed distributions. Here, differences between AEPS and EPS arise. The data for adaptive exploiting probabilistic search is more centered as for exploiting probabilistic search. Differences can also be found in the decision trees. Here, the decision trees for AEPS and EPS are split for factors stemming from the hazardous area or more correctly, its size leading to the conjecture, that the performance may be affected by too large disturbance fields. It should be noted, however, the worst median performance measure in the tree remains close to the best value signaling a quite robust behavior for the input parameter space considered. In the case of AEPS, the parameter α which controls the influence of the local information appears to be additionally important whereas in the case of EPS, the parameter p_{SPS} appears which determines the probability to follow either the stochastic decision rule or to exploit the information deterministically. Apparently, if the proportion of the deterministic decision making is too large, the exploration capability of the swarm is affected resulting in a worse performance. A reinforcing factor in this case is the exponent of the safety of passage. While the number of data points is too low to derive significant conclusions, the findings indicate that exploiting probabilistic search may require a careful balancing of risk aversion and target finding if the search is dominantly deterministic. Similarly, if AEPS focuses too strongly on local information, it may take longer to find the target.

Concerning the number of incidents, both strategies lead to good results. The box-plots in Fig. 1 as well as the histograms in Figs. 2, and 3 reveal that the number of incidents remain low in the majority of the experiments but that outliers are present and that the distributions are heavy-tailed. The deciding factor appears to be the movement frequency of the disturbance field. Both strategies are strongly affected if the field relocates extremely often. This may be aggravated additionally by its potentially large movement speed. When the location of the field changes less frequently, in the experiments, around every 8th time step (EPS) or 17th (AEPS), the median number of incidents drops considerably. Concerning the split, there may be an advantage for the exploiting probabilistic search. While EPS may result in a slightly larger percentage of runs, where incidents occur, it may be more robust w.r.t. the number of affected searchers.

The duration of incidents remains low in most experiments. Again outliers can be observed. In the case of AEPS, the decision tree in Fig. 2 traces the most aggravating factor back to the change frequency of the disturbance field. A larger number of relocations impairs the performance leading to larger values of the incident duration. Additionally, the number of searchers appear to be important. This may be due to stronger repulsion which may cause a wider distribution in the search space and thus a more robust behavior. However, this needs to be investigated further since number of corresponding data points is rather low. The same but even more



a) number of target hits b) time until first target hit c) number of incidents c) duration of incidents

Figure 1: Box-plots for the cases of EPS and AEPS. The measures are relative to the number of searchers.

strongly holds for the left part of the tree. Concerning the decision tree for EPS in Fig. 3, the later data splits indicate advantages for larger swarms in some cases. This could mean that the risk for a single swarm member would be lower. This may be interesting since the findings may be caused by the stronger exploitation component. More experiments must be performed, however.

4 CONCLUSIONS

This paper addressed moving target search in hazardous environments and focused on the search coordination of a team of mobile sensor platforms or robots. Each robot is equipped with sensors which can be used to infer the target's position and to gain information concerning the state of the environment. We assumed a dynamically changing environment where hazardous areas cover a significant part of the search space and searchers may be trapped by rapidly changing conditions. Concepts from particle swarm optimization and ant colony optimization together with a safe-path finding procedure stemming from artificial slime-molds [12] were applied. Two promising strategies, the adaptive exploiting probabilistic search and the exploiting probabilistic search were investigated in a series of data farming experiments. The situation considered assumed risk areas with a potentially large cohesive region which are able to relocated fast and frequently. The experiments revealed that considering the performance measures chosen which record the occurrences of incidents and target hits, both strategies lead to good and robust performances. Concerning the task of finding the target, the adaptive variant seems to offer a slight advantage which may be due to its differing exploration-exploitation balancing procedure. The stronger exploitation of the second variant may be helpful in avoiding the risk areas of the search environment. Additionally, the experiments hinted at that the swarm size should be chosen with care with disadvantages appearing for too small and too large swarms.

REFERENCES

[1] T. M. Blackwell and Peter J. Bentley. 2002. Dynamic Search With Charged Swarms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '02)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 19–26. http://dl.acm.org/citation.cfm?id=646205.682961

[2] A. G. Brandstein and G. E. Horne. 1998. *Data Farming: A Meta-technique for Research in the 21st Century.* Maneuver Warfare Science 1988. US Marine Corps Combat Development Command Publication.

[3] T. Chung, G. Hollinger, and V. Isler. 2011. Search and pursuit-evasion in mobile robotics. *Autonomous Robots* 31 (2011), 299 – 316. Issue 4. http://dx.doi.org/10.1007/s10514-011-9241-4

[4] M. Dorigo and L.M. Gambardella. 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on* 1, 1 (1997), 53 – 66. https://doi.org/10.1109/4235.585892

[5] M. Dorigo and T. Stützle. 2004. *Ant Colony Optimization.* MIT Press.

[6] A. P. Engelbrecht. 2005. *Fundamentals of Computational Swarm Intelligence.* Wiley.

[7] S. García, D. Molina, M. Lozano, and F. Herrera. 2009. A Study on the Use of Non-parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics* 15, 6 (Dec. 2009), 617–644. https://doi.org/10.1007/s10732-008-9080-4

[8] B. P. Gerkey, S. Thrun, and G. Gordon. 2005. Parallel stochastic hill-climbing with small teams. In *Multi-Robot Systems: From Swarms to Intelligent Automata*, L. E. Parker and others (Eds.), Vol. III. Springer, 65–77.

[9] A. S. Hernandez, T. W. Lucas, and M. Carlyle. 2012. Constructing nearly orthogonal latin hypercubes for any nonsaturated run-variable combination. *ACM Transactions on Modeling and Computer Simulation* 22, 4 (2012), 1–17. https://doi.org/10.1145/2379810.2379813

[10] A. Johannson and J. Zou. 2012. A Slime Mold Solver for Linear Programming Problems. In *How the World Computes*, S. B. Cooper, A. Dawar, and B. Löwe (Eds.). Lecture Notes in Computer Science, Vol. 7318. Springer Berlin Heidelberg, 344–354. https://doi.org/10.1007/978-3-642-30870-3_35

[11] J. Kleijnen. 2008. *Design and Analysis of Simulation Experiments.* Springer.

[12] E. Kropat and S. Meyer-Nieberg. 2016. A Multi-layered Adaptive Network Approach for Shortest Path Planning During Critical Operations in Dynamically Changing and Uncertain Environments. In *Proceedings of the 2016 49th Hawaii International Conference on System Sciences (HICSS) (HICSS '16)*. IEEE Computer Society, Washington, DC, USA, 1369–1378. https://doi.org/10.1109/HICSS.2016.173

[13] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan. 2005. MASON: A Multiagent Simulation Environment. *Simulation* 81, 7 (July 2005), 517–527. https://doi.org/10.1177/0037549705058073

[14] S. Meyer-Nieberg, E. Kropat, S. Pickl, and A. B. Bordetsky. 2013. Intercepting a Target with Sensor Swarms. In *HICSS-46*. IEEE, 1222–1230.

[15] H. B. Mitchel. 2012. *Data Fusion: Concepts and Ideas* (2nd ed.). Springer-Verlag Heidelberg, Berlin.

[16] K. Ok, S. Ansari, B. Gallagher, W. Sica, F. Dellaert, and M. Stilman. 2013. Path planning with uncertainty: Voronoi Uncertainty Fields. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* 4596–4601. https://doi.org/10.1109/ICRA.2013.6631230

[17] T. J. Santner, B. J. Williams, and W. I. Notz. 2003. *The Design and Analysis of Computer Experiments.* Springer. 283 pages.

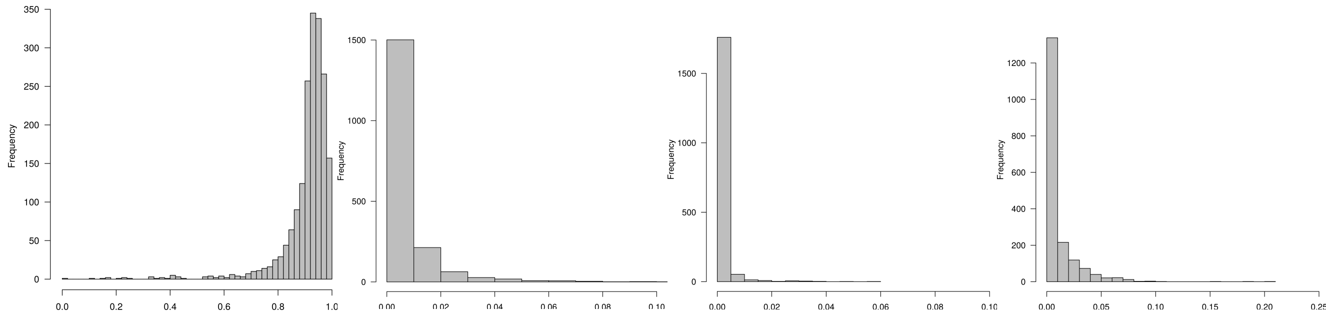
[18] H. Sato and J. O. Royset. 2010. Path optimization for the resource-constrained searcher. *Naval Research Logistics (NRL)* 57, 5 (2010), 422 – 440. https://doi.org/10.1002/nav.20411

[19] M. Senanayake, I. Senthooran, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed. 2016. Search and tracking algorithms for swarms of robots: A survey. *Robotics and Autonomous Systems* 75, Part B (2016), 422 – 434. https://doi.org/10.1016/j.robot.2015.08.010

[20] K. Sullivan, M. Coletti, and S. Luke. 2010. *GeoMason: GeoSpatial Support for MASON.* Technical Report GMU-CS-TR-2010-16. George Mason University Technical Reports. http://cs.gmu.edu/~tr-admin/papers/GMU-CS-TR-2010-16.pdf

[21] A. Tero, R. Kobayashi, and T. Nakagaki. 2007. A mathematical model for adaptive transport network in path finding by true slime mold. *Journal of Theoretical Biology* 244, 4 (2007), 553 – 564. https://doi.org/10.1016/j.jtbi.2006.07.015

[22] N. Wen, X. Su, P. Ma, L. Zhao, and Y. Zhang. 2015. Online UAV path planning in uncertain and hostile environments. *International Journal of Machine Learning and Cybernetics* (2015), 1–19. https://doi.org/10.1007/s13042-015-0339-4

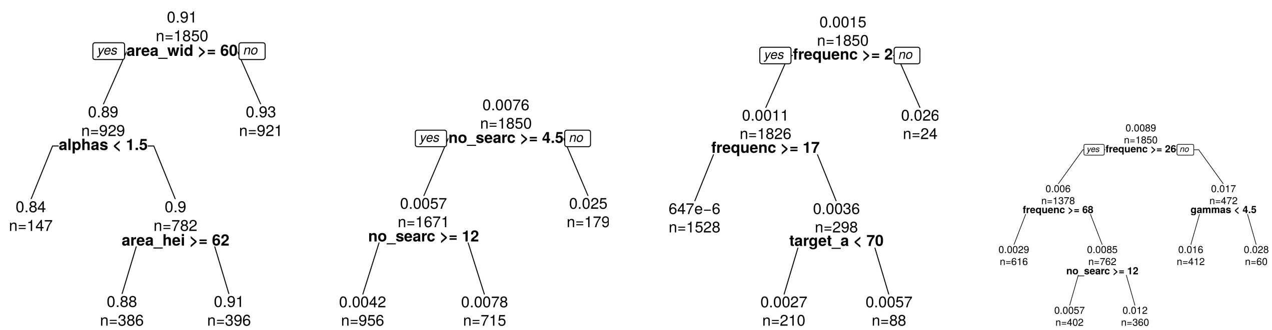


a) no of target hits histogram

b) time of first target hit histogram

c) no of incidents histogram

d) time of of first target hit histogram



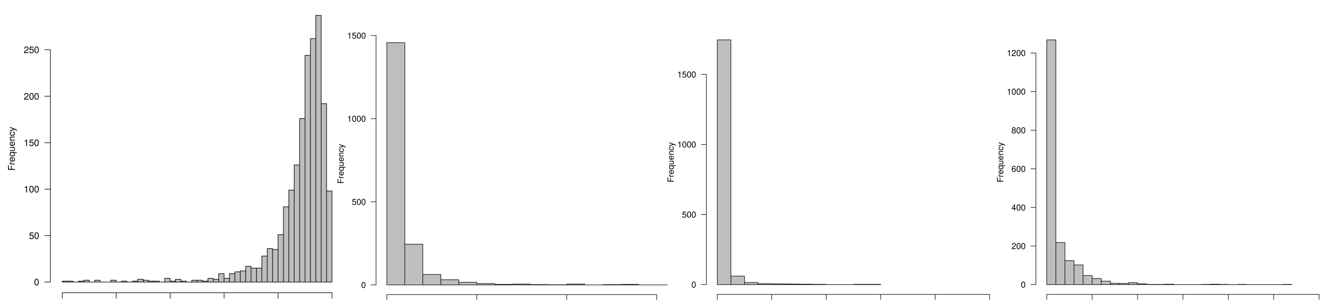
e) no of target hits decision tree

f) time of first target hit decision tree

g) no of incidents decision tree

h) duration of incidents decision tree

Figure 2: Results for the adaptive exploiting probabilistic search (AEPS) on the hexagonal grid. The measures are relative to the number of searchers.

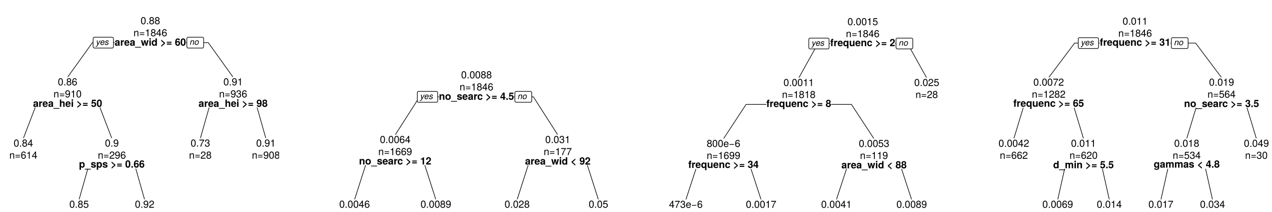


a) no of target hits histogram

b) time of first target hit histogram

c) no of incidents histogram

d) time of of first target hit histogram



e) no of target hits decision tree

f) time of first target hit decision tree

g) no of incidents decision tree

h) duration of incidents decision tree

Figure 3: Results for the exploiting probabilistic search (EPS) on the hexagonal grid. The measures are relative to the number of searchers.