

k -Survivability: Diversity and Survival of Expendable Robots

Yu-Han Lyu, Yining Chen, and Devin Balkcom

Abstract—We define the k -survivability of a set of n paths as the probability that at least k out of n robots following those paths through a stochastic threat environment reach goals. High k -survivability sets tend to contain short and diverse paths. Finding sets of paths with maximum k -survivability is NP-hard. We design two algorithms: a complete algorithm that finds an optimal list of paths, and a heuristic method that finds paths with high k -survivability. Although computing k -survivability is expensive and this work is still preliminary, we believe that understanding the relationship between diversity and survival will yield new insights into multirobot motion planning.

Index Terms—Path Planning for Multiple Mobile Robots or Agents, Redundant Robots, Motion and Path Planning, Probability and Statistical Methods, Search and Rescue Robots.

I. INTRODUCTION

HOW should a set of robots move through a dangerous environment to accomplish objectives? Is it better for the robots to travel together, or should the robots split up? What is the relationship between survival and diversity of actions?

As an example, consider the following whimsical planning problem: n ants must migrate from one nest to another through a field containing both obstacles and antlions, which make disc-shaped traps. If we assume a uniform distribution of trap locations, which n paths should the ants follow, if the ants must decide their paths before moving and cannot reroute during movement?

One idea might be to maximize the expected number of surviving ants. However, the best strategy for this problem turns out to be uninteresting and unwise: find the safest path for a single ant (for simplicity, assume there is a unique safest path), and have all ants follow that path. This solution is not robust – a single trap could destroy the entire colony. Therefore, we consider a problem that is more suitable if ants are expendable: maximize the probability that at least some k (with $k \leq n$) ants survive. If the number of traps is unknown, the solution may contain up to n unique paths.

Figure 1 shows an example problem for which paths have been selected to achieve high survivability of routes across a college campus. The paths are short, interestingly diverse, and may be of practical interest if there is actual danger, traffic congestion, or surveillance to be avoided.

Manuscript received August 31, 2015; accepted January 10, 2016. Date of publication February 2, 2016; date of current version March 22, 2016. This paper was recommended for publication by Associate Editor H. Kurniawati and Editor N. Amato upon evaluation of the reviewers' comments. This work was supported by the NSF under Grant IIS-0643476.

The authors are with the Department of Computer Science, Dartmouth College, Hanover, NH 03755 USA (e-mail: devin@cs.dartmouth.edu).

Digital Object Identifier 10.1109/LRA.2016.2524067

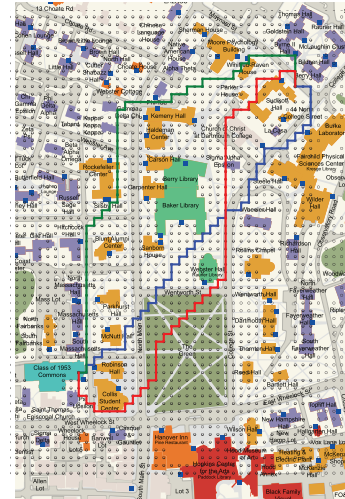


Fig. 1. A high 1-survivability set of three paths from Department of Computer Science (Sudikoff) to food court (1953 Commons) on the Dartmouth campus; gray circles represent the discretization of the environment.²

We believe this to be the first work that explicitly studies the theoretical implications of robot expendability. Path diversity has been explored in several settings, with applications including motion planning [1]–[7], robust routing in computer networks [8], and dissimilar paths in transportation [9]. Approaches to finding diversity typically involve defining an arbitrary distance metric that describes separation of paths, and finding solutions that balance distance between paths against length of paths, using linear or non-linear weights, constrained optimization, or by analyzing the Pareto frontier.

Instead of defining an arbitrary pairwise path diversity metric or choosing arbitrary tradeoffs between path lengths and diversity metrics, our approach proceeds directly from the threat model, since we believe that diversity should be considered as a means rather than an end.

We define k -survivability to measure the quality of paths in a stochastic threat environment. Sets of paths with high k -survivability naturally balance length and diversity. Although choosing sets of paths to maximize k -survivability is NP-hard, we design a complete algorithm. Since the maximization algorithm is computationally infeasible except for $k = 1$ and $n = 2$, we also design a practically faster heuristic method that finds paths with high k -survivability.

A. Model

We focus on the discrete problem in which the environment is represented by a point set D and the free space is represented by

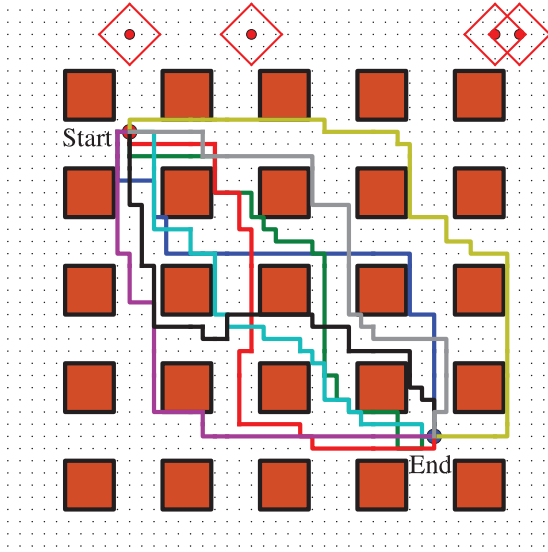


Fig. 2. Eight paths with high 1-survivability. Small gray circles are vertices of G (4-connected). Squares are obstacles; diamonds are example fixed traps.

a graph $G = (V, E)$, where $V \subseteq D$. Unknown stochastic dangers are called *traps*. Several models of traps will be discussed in the next section.

Our problem is to find paths for n point robots such that the i -th path connects the designated start vertex $s_i \in V$ and the designated goal vertex $g_i \in V$. See Figure 2. Robots cannot communicate, do not have sensors, and cannot reroute; both obstacles and traps are time-independent.

We define k -survivability to be the probability that at least k paths successfully connect their (perhaps different) starts to goals. The k -survivability problem (k SP) is formalized as:

Input = $(G, M, \{(s_i, g_i)\}_{i=1}^n, k)$, where

- 1) $G = (V, E)$ denotes the free space.
- 2) M is a trap model (see next section).
- 3) n point robots have start locations $s_i \in V$ and goal locations $g_i \in V$ for all $1 \leq i \leq n$.
- 4) survivability parameter k , with $1 \leq k \leq n$.

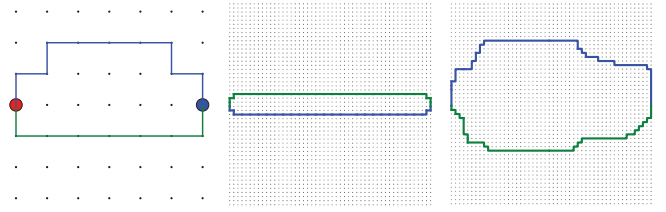
Output = \mathcal{P} , a list of n paths maximizing k -survivability such that for all $1 \leq i \leq n$, $P_i \in \mathcal{P}$ connects s_i and g_i .

We now discuss two trap models: *fixed traps*, which have known shapes, and *variable traps*, for which the shape is unknown but drawn from some known distribution. Even under the fixed trap model, a different trap shape (or even multiple traps) may be placed at each vertex.

A *fixed trap* F is a subset of D . When a fixed trap F is in effect, all paths passing through F are blocked. A *fixed trap model* $M = \{(F_i, p_i)\}_{i=1}^{|M|}$ is a collection of fixed traps and their corresponding, independent probabilities. If all probabilities are equal, then the model is a *uniform fixed trap model*.

For example, under a *uniform fixed r -disc trap model*, each vertex has equal and independent probability to be the center of a disc trap of radius r . Figures 3a and 3b show examples.

²The campus map is from the Dartmouth College website and the street data is from OpenStreetMap. Since data from OpenStreetMap is inconsistent with the campus map, some vertices are slightly inside buildings.



(a) Paths with the highest 1-survivability under the fixed 1-disc trap model. (b) Paths with high 1-survivability under the fixed 3-disc trap model. (c) Paths with high 1-survivability under the variable 5-disc trap model.

Fig. 3. Example paths for two robots in different environments and parameters. In Figure 3a, since $r = 1$, the optimal solution has parallel subpaths with distance two to avoid being destroyed by one 1-disc easily.

A *variable trap* A is represented by a distribution over a set of fixed traps. A *variable trap model* M is represented by a collection of variable traps and corresponding probabilities: $M = \{(A_i, p_i)\}_{i=1}^{|M|}$.

For example, under a *variable r -disc trap model*, each vertex has identical and independent probability to be the center of a disc trap, whose radius follows a geometric distribution with mean r . Two paths with high 1-survivability under the variable 5-disc trap model are shown in Figure 3c.

II. RELATED WORK

Diversity has been studied in location theory, motion planning, graph theory, computer networks, and transportation.

Location theory: In location theory, the maximum diversity problem is to find m points maximizing diversity among given points in a metric space. Although location theory focuses on finding diverse points, methods can be adapted to find diverse paths as long as a metric space on paths can be defined. Formulations include [10]:

- 1) *remote-edge problem*: find a set of points maximizing the minimum mutual distance (also called the p -dispersion problem [11]).
- 2) *remote-pseudoforest problem*: find a set of points maximizing the sum of the distance to the nearest neighbors (also called the p -defense problem [12]).
- 3) *remote-clique problem*: find a set of points maximizing the sum of mutual distances (also called the *max-avg facility dispersion* problem [13], or the *maximum dispersion* problem [11]).

Diverse trajectories in motion planning: Increasing the diversity of trajectories in motion planning has been studied by several researchers [1]–[7]. Our work is most related to Erickson's and LaValle's work [14]. They propose a definition of *survivability* that measures the correlation of damage on paths when a random disc obstacle is placed on a path. Whereas survivability favors separated paths, k -survivability is a direct probabilistic measure of survival that in some cases can be maximized by allowing robots to follow overlapping short paths.

Finding trajectories in a threat environment has been studied for aircrafts [15], UAVs [16], vehicles [17], and ships [18]. Our

work differs in that the threat model is probabilistic, and in the search for multiple trajectories.

In the Euclidean plane, finding a path connecting two points among polygonal obstacles can be solved efficiently [19]. One possible definition for the diversity of paths in the Euclidean plane is the number of distinct homotopy classes of paths [20]. Eriksson-Bique *et al.* studied the problem of finding k shortest paths with distinct homotopy classes [21].

Path diversity on graphs: The problem of finding k -shortest paths on a graph has been studied since the '70s [22], as the problem of finding vertex or edge disjoint paths [23]. When each vertex/edge is associated with a failure probability, short and reliable paths are desirable. Finding short paths subject to reliability constraints can be considered as resource-constrained shortest-path problems [24]. These models only disfavor paths sharing edges or vertices, while k -survivability disfavors paths passing through the same traps, which is more general.

Robust routing in computer network: One way to improve the robustness of a network is to increase the path diversity between end-points [25]. Diverse routing problems have been studied for more than a decade using graph theory methods [8]. Rohrer *et al.* define the diversity of paths based on the distance on graphs and geographic distances [26], which is similar to the idea of path space [5].

Dissimilar paths in transportation: The problem of finding dissimilar paths has been studied in transportation, since dissimilar paths avoid bottlenecks and are beneficial for *e.g.* hazardous waste transportation [9], [27].

III. COMPUTING k -SURVIVABILITY

Since k -survivability is independent of the order of vertices along paths, paths are represented as sets of vertices.

A. Computing k -Survivability Under the Fixed Trap Model

Given a fixed trap model $M = \{(F_i, p_i)\}_{i=1}^{|M|}$ and a path P on a graph G , the *forbidden index set* of P is $\text{Forbid}(P) = \{i | P \cap F_i \neq \emptyset\}$. The probability that P is not blocked equals $\Pr(P) = \prod_{i \in \text{Forbid}(P)} (1 - p_i)$. Similarly, for a set of paths $\mathcal{P} = \{P_1, \dots, P_h\}$, the *forbidden index set* of \mathcal{P} is $\text{Forbid}(\mathcal{P}) = \cup_{P \in \mathcal{P}} \text{Forbid}(P)$. The probability that all paths in \mathcal{P} are not blocked equals $\Pr(\mathcal{P}) = \prod_{i \in \text{Forbid}(\mathcal{P})} (1 - p_i)$.

Given a set of paths \mathcal{P} , 1-survivability can be computed by using the inclusion-exclusion principle as follows:

$$\text{Survive}_1(\mathcal{P}) = \sum_{S \subseteq \mathcal{P}, S \neq \emptyset} (-1)^{|S|+1} \Pr(S)$$

The computation of k -survivability is similar but more time-consuming. For a set of paths \mathcal{P} and a positive integer k , we define $\text{comb}(\mathcal{P}, k) = \{S | S \subseteq \mathcal{P}, |S| = k\}$; that is, $\text{comb}(\mathcal{P}, k)$ contains all k -subsets of \mathcal{P} . Given a set of paths \mathcal{P} , k -survivability can be computed by using the inclusion-exclusion principle:

$$\text{Survive}_k(\mathcal{P}) = \sum_{R \subseteq \text{comb}(\mathcal{P}, k), R \neq \emptyset} (-1)^{|R|+1} \Pr(\cup_{S \in R} S)$$

B. Computing k -Survivability Under the Variable Trap Model

Let $M = \{(A_i, p_i)\}_{i=1}^{|M|}$ be a variable trap model. Each variable trap A_i can be represented as a collection of fixed traps and their probabilities: $A_i = \{(F_{i,j}, p_{i,j})\}_{j=1}^{|A_i|}$. Given a path P on a graph G , the *forbidden index set* of P with respect to the i -th variable trap is $\text{Forbid}(P, i) = \{j | P \cap F_{i,j} \neq \emptyset\}$. The probability that P is not blocked equals $\Pr(P) = \prod_i (1 - \sum_{j \in \text{Forbid}(P, i)} p_{i,j})$. The calculation of k -survivability can be derived in the same manner as under the fixed trap model.

IV. THEORETICAL RESULTS

In this section, we show that k SP is NP-hard. We also show that maximizing k -survivability leads to diversity in the sense that at most k robots will follow the same path, if other paths are available.

A. NP-Hardness of k SP

We show that k SP under the uniform fixed trap model is NP-hard with $n = k = 1$. Since the fixed trap model is a special case of the variable trap model, k SP under the variable trap model is also NP-hard.

Our proof, similar to the NP-completeness proof of the minimum color path problem [28], is a reduction from the *minimum set cover problem* (MSCP), defined below, to k SP.

Input = (S, \mathcal{C}) , where

1) $S = \{1, \dots, a\}$ is a set of positive integers from 1 to a .

2) $\mathcal{C} = \{C_1, \dots, C_b\}$ is a collection of subsets of S .

Output = $\mathcal{C}' \subseteq \mathcal{C}$ a minimum cardinality collection of subsets whose union is S .

Theorem 1: k SP under the uniform fixed trap model with $n = k = 1$ is NP-hard.

Proof: Let (S, \mathcal{C}) be an instance of MSCP. We construct an instance $(G, M, \{(s, g)\}, 1)$ of k SP in polynomial time such that an optimal solution in k SP can be transformed into a minimum set cover of (S, \mathcal{C}) and vice versa. ■

We construct $G = (V, E)$, which is a subgraph of a grid graph, as follows. First, for each element $i \in S$, we create three vertices $v_i^{0,0} = (4i, 0)$, $v_i^{1,0} = (4i + 1, 0)$, and $v_i^{3,0} = (4i + 3, 0)$. We create edges $(v_i^{0,0}, v_i^{1,0})$ for all $1 \leq i \leq a$ and edges $(v_i^{3,0}, v_{i+1}^{0,0})$ for all $1 \leq i < a$. Our idea is to design a gadget for each element $i \in S$ such that i can be covered by a set in \mathcal{C} if and only if a solution of k SP, which is a path, passes through $v_i^{0,0}$ and $v_i^{3,0}$.

Second, for each $j \in \{1, \dots, b\}$, we create vertices $v_i^{1,j} = (4i + 1, j)$ and $v_i^{3,j} = (4i + 3, j)$ for all $1 \leq i \leq a$. Then, we create edges $(v_i^{1,j}, v_{i+1}^{1,j+1})$ and $(v_i^{3,j}, v_{i+1}^{3,j+1})$ for all $0 \leq j < b$. Intuitively, the j -th row represent the j -th set in \mathcal{C} .

Finally, we create vertices $v_i^{2,j} = (4i + 2, j)$, edges $(v_i^{1,j}, v_i^{2,j})$, and edges $(v_i^{2,j}, v_i^{3,j})$ for each $i \in C_j$. We use these vertices to model the constraint that every element in S is covered by a set in \mathcal{C} . See Figure 4.

The uniform fixed trap model is $M = \{(F_j, p)\}_{j=1}^b$ for an arbitrary choice $p \in (0, 1)$, where $F_j = \{v_i^{2,j} | i \in C_j\}$. The start vertex is $v_1^{0,0}$ and the goal vertex is $v_a^{3,0}$.

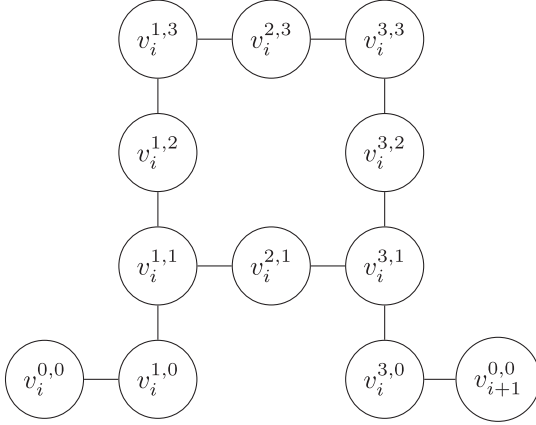


Fig. 4. Gadget used in the proof of Theorem 1.

Let P be an optimal solution of the instance $(G, M, \{(v_1^{0,0}, v_a^{3,0})\}, 1)$ of k SP. By the construction of the graph, P passes every vertex $v_i^{3,0}$ for all $1 \leq i \leq a$. Moreover, for each $1 \leq i \leq a$, the only way to get $v_i^{3,0}$ is to pass through a vertex $v_i^{2,j}$ for some $1 \leq j \leq b$. Since $v_i^{2,j}$ exists if and only if $i \in C_j$, the set $C' = \{C_j | \exists 1 \leq i \leq a, P \text{ passes } v_i^{2,j}\}$ is a set cover of S .

Moreover, since all traps have the same probability, maximizing 1-survivability is the same as minimizing $\{j | \exists 1 \leq i \leq a, P \text{ passes } v_i^{2,j}\}$. Hence, C' is also an optimal solution of MSCP.

Transforming an optimal solution of MSCP to an optimal solution of k SP can be done similarly.

Note that this reduction relies on the fact that individual fixed traps might be formed from disconnected sets of vertices. However, even if we restrict individual fixed traps to be contiguous, the problem still appears to be hard, since k SP under the uniform fixed trap model with $n = k = 1$ can be used to solve the *barrier resilience problem* [29]. The complexity of the barrier resilience problem is still open and currently no polynomial time algorithm exists.

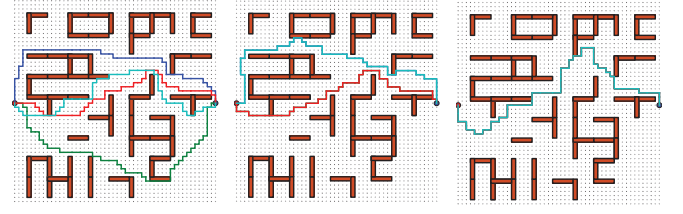
Even approximating an optimal solution is hard:

Theorem 2: No polynomial time algorithm with constant approximation ratio for k SP with $n = k = 1$ under the uniform fixed trap model exists unless $P = NP$.

Proof: We show that if a r -approximation polynomial time algorithm for k SP exists for some constant r , then we can solve MSCP in polynomial time. ■

Let (S, C) be an instance of MSCP and c be the size of a minimum set cover. By using the same reduction as in the proof of Theorem 1, we obtain an instance $(G, M, \{(s, g)\}, 1)$ of k SP. Since c is the size of a minimum set cover, the optimal solution of $(G, M, \{(s, g)\}, 1)$ has value $(1 - p)^c$.

Suppose that a r -approximation algorithm for k SP exists, $0 < r < 1$, and this approximation algorithm is guaranteed to find a solution with 1-survivability at least $r(1 - p)^c$. Since the choice of probability p in the reduction is arbitrary, we set p to be a value satisfying $r > (1 - p)$. Because $r(1 - p)^c > (1 - p)^{c+1}$ and a path can only pass through an integral number of fixed traps, the approximation algorithm must return a solution with value $(1 - p)^c$, which is an optimal solution



(a) $k = 1$. All robots take different paths. (b) $k = 2$. Robots form two groups. (c) $k = 4$. All robots take the same path.

Fig. 5. Example paths with high k -survivability for different values of k under the fixed 5-disc trap model.

of $(G, M, \{(s, g)\}, 1)$ and can be transformed into an optimal solution of (S, C) in polynomial time.

B. Properties of k SP

We now show that k -survivability leads to diverse paths.

Observation 3: For k SP under the fixed trap model with $k = 1$, if n paths with different forbidden index sets exist, then any optimal solution does not have duplicate paths.

Proof: Since paths with the same forbidden index sets are either all-safe or all-blocked, using paths with different forbidden index sets improves 1-survivability. ■

Observation 4: For k SP under the fixed trap model, if at least $\lceil n/k \rceil$ paths with different forbidden index sets exist, then at most k robots follow the same path in an optimal solution.

Proof: If more than k robots follow the same path, moving one robot to another path always improves k -survivability.

Note that when k increases, the number of different paths in optimal solutions may decrease. See figure 5, which shows some high-survivability paths for different values of k . ■

V. ALGORITHMS

In this section, we first describe a complete algorithm to find optimal solutions. Then, we describe a typically faster heuristic method to find good solutions. For simplicity, we assume that all robots have the same start s and the same goal g , but this assumption may be easily lifted.

A. Complete Algorithm

In this section, we design a complete state space search algorithm for k SP under the uniform fixed trap model with $n = 2$ and $k = 1$. Although it is easy to extend this algorithm for larger k and n , solving even small problems becomes computationally infeasible with this approach.

We need several definitions. A path P is an ordered list of vertices. A path P' *extends* another path P , if P is a prefix of P' . A path P' is a *feasible extension* of P if either P ends at g and $P' = P$, or P does not end at g and P' extends P by one vertex. Let $\text{Ext}(P)$ denote the set of all paths that end at g and are extensions of path P .

The complete algorithm is a state space search algorithm. Each state t consists of two simple paths (P_1, P_2) starting from

s . A state (P'_1, P'_2) is a *successor* of a state (P_1, P_2) if P'_1 and P'_2 are feasible extensions of P_1 and P_2 respectively.

The initial state is $(\langle s \rangle, \langle s \rangle)$ and the goal states are all states (P_1, P_2) that both P_1 and P_2 end at g . We will find one goal state with maximum 1-survivability.

Since the state space is a tree, we can use a brute-force approach to traverse the tree to find an optimal solution. In order to speed up the brute-force approach, we design a heuristic function h of states, where $h(t)$ is an upper-bound of 1-survivability of all goal states that are reachable by state t . As long as $h(t)$ is optimistic, then the tree search will find an optimal solution. Using the heuristic function, we can prune unnecessary branches and stop search when the algorithm reaches one of the goal states for the first time.

We construct a heuristic function h as follows. Remember that when $n = 2$, 1-survivability of two paths P_1 and P_2 is $\Pr(\{P_1\}) + \Pr(\{P_2\}) - \Pr(\{P_1, P_2\})$. Suppose that there is a function \hat{h} for paths that $\hat{h}(P)$ is an upper bound of $\Pr(\{P'\})$ for all $P' \in \text{Ext}(P)$. Then, we obtain a heuristic function $h((P_1, P_2)) = \hat{h}(P_1) + \hat{h}(P_2) - \Pr(\{P_1, P_2\})$.

Now, we show how to construct a function \hat{h} . Let M be the uniform fixed trap model. For any path P , 1-survivability of P is $(1 - p)^{|\text{Forbid}(P)|}$, which only depends on the size of $\text{Forbid}(P)$. Let $\text{LB}(P)$ be the minimum number of additional fixed traps that any extension of path P must pass through to reach the goal. Formally,

$$\text{LB}(P) = \min_{P' \in \text{Ext}(P)} |\text{Forbid}(P') \setminus \text{Forbid}(P)|.$$

Then, $(1 - p)^{|\text{Forbid}(P) + \text{LB}(P)|}$ is the least upper bound of $\Pr(\{P'\})$ for all $P' \in \text{Ext}(P)$.

Note that computing $\text{LB}(P)$ exactly is the same as solving $k\text{SP}$ under the uniform fixed trap model with $n = k = 1$, which is a NP-hard problem by Theorem 1. In order to get an upper bound of all $\Pr(\{P'\})$, where P' is in $\text{Ext}(P)$, it suffices to obtain a lower bound of $\text{LB}(P)$.

Our idea of obtaining a lower bound of $\text{LB}(P)$ is as follows. Let F_i be a fixed trap that i does not belong to $\text{Forbid}(P)$. If an extension P' of P passes through one vertex of F_i , then charge P' by $1/|F_i \cap V|$. Thus, if an extension P' of P passes through one $v \in V$, then we charge P' by $\sum_{i \notin \text{Forbid}(P), v \in F_i} 1/|F_i \cap V|$. The minimum charge of any extension of P that reaches the goal, $\text{LB}'(P)$, can be computed efficiently by using a shortest path algorithm.

It is easy to see that $\text{LB}'(P)$ is a lower bound of $\text{LB}(P)$ and we know

$$\text{LB}'(P) \leq \text{LB}(P) \leq \max_i |F_i| \cdot \text{LB}'(P).$$

Thus, we can use $\hat{h}(P) = (1 - p)^{|\text{Forbid}(P) + \text{LB}'(P)|}$ to obtain a heuristic function h .

B. Heuristic Algorithm

The previous algorithm uses a heuristic function for pruning, but is guaranteed to find optimal solutions. The heuristic algorithm described in this section does not provide this guarantee.

Algorithm 1. Heuristic algorithm for $k\text{SP}$

input: $(G, M, \{(s_i, g_i)\}_{i=1}^n, k, w, T)$,
where $(G, M, \{(s_i, g_i)\}_{i=1}^n, k)$ is an instance of $k\text{SP}$,
 w is a parameter of the path generation, and T is a
parameter of the path improvement.

output: n paths connecting (s_i, g_i) respectively.

$\mathcal{R} = \text{path_generation}(w)$

$\mathcal{S} = \text{path_selection}(\mathcal{R})$

$\mathcal{S} = \text{path_replacement}(\mathcal{S}, \mathcal{R})$

$\mathbf{Q} = \emptyset$

while $|\mathbf{Q}| < T$ **do**

$\mathcal{S} = \text{path_shortening}(\mathcal{S})$

$\mathbf{Q} = \mathbf{Q} \cup \{\mathcal{S}\}$

$\mathcal{S} = \text{escape}(\mathcal{S})$

return the best solution in \mathbf{Q} .

There are three phases: path generation, path selection, and path improvement. Due to the high-dimensional search space of $k\text{SP}$, we first generate a set of candidate paths with size $w \gg n$ to reduce the search space to these w paths. Then, we heuristically find n paths among the set of candidate paths as an initial solution. Finally, we use local search to improve the solution until the process is stabilized. Algorithm 1 outlines the approach.

Since computation of k -survivability is potentially expensive, we only use the computation of k -survivability in the last phase. Moreover, this heuristic algorithm only needs a black box to compute k -survivability, and the same algorithm can be used for both fixed trap and variable trap models.

1) *Path Generation:* The purpose of this phase is to generate a set \mathcal{R} of $w \gg n$ paths. We design two methods: random generation, and an iterative penalty approach.

a) *Random generation method:* To generate one random path, we generate a random spanning tree first and then pick the unique path between s and g on the tree. We repeat this process until w paths are generated.

b) *Iterative penalty method:* Another way to generate w paths is repeatedly apply a shortest path algorithm. After a shortest path P is found, we increase the edge weights of all edges in P and repeat. Akguin *et al.* discuss several variants of iterative penalty methods that have different ways to penalize the path [27].

2) *Path Selection:* The purpose of this phase is to generate a set n paths among w candidate paths generated in the path generation phase. Although we can design an algorithm to find n paths that maximize k -survivability, since the computation of k -survivability is exponential in n , this approach would be expensive. Thus, our strategy is to use different heuristics to obtain an initial solution without evaluating k -survivability. Then, improve the initial solution based on k -survivability in the next phase.

We find an initial solution by solving a different but related optimization problem.

a) *Distance-based heuristic:* We use $d_G(P, P')$ to denote the distance between two paths P and P' on a graph G . One candidate of the distance function is discrete Fréchet

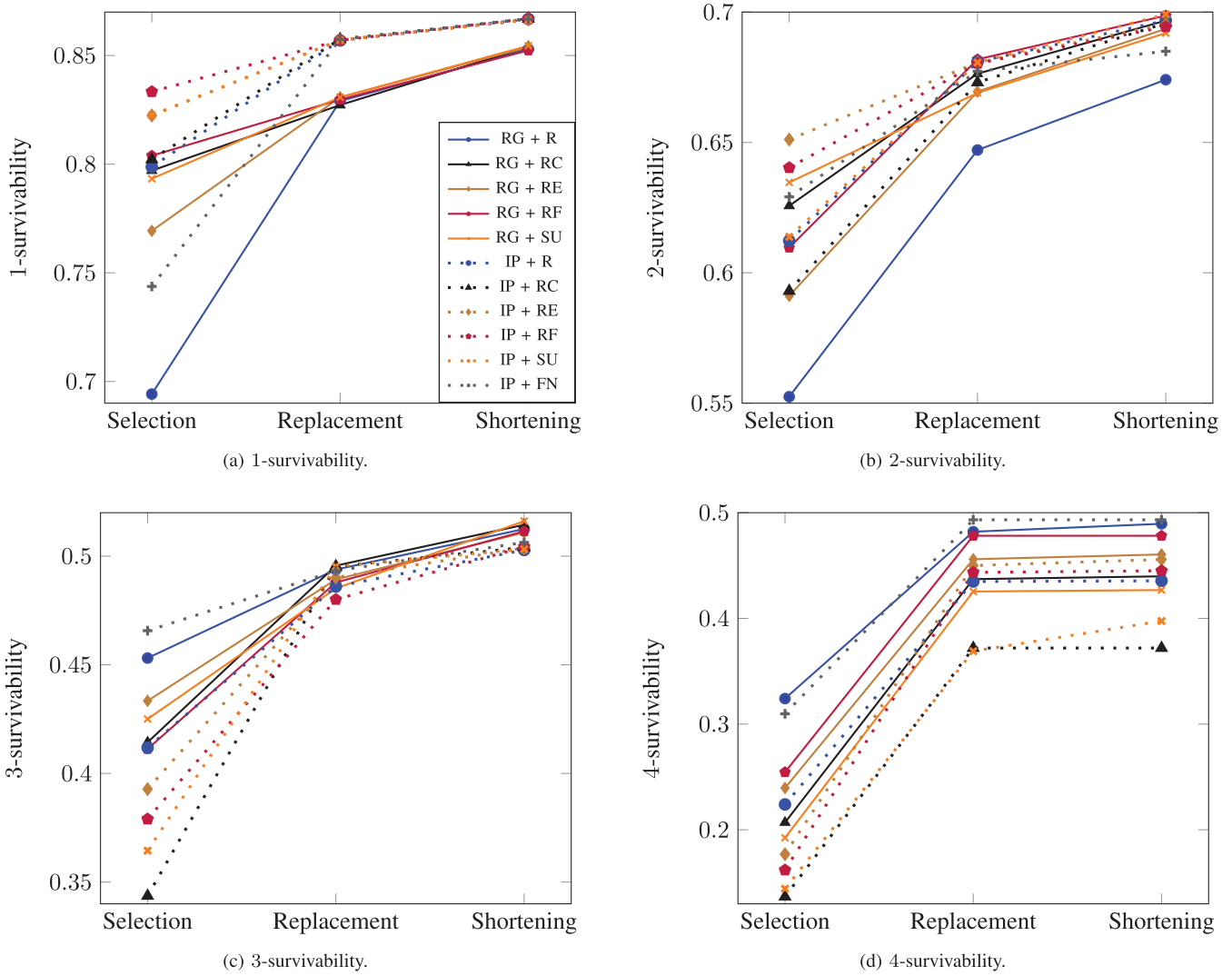


Fig. 6. All methods' k -survivability at the end of path selection phase, path improvement phase and path shortening phase. All methods that are using random generation method are represented by solid lines and all methods that are using iterative penalty method are represented by dotted lines. Different methods in the second phase are represented by different colors.

distance [7] and other candidates of distance function can be found in Knepper's thesis [4].

Based on the distance function, we can set up several optimization problems.

1) remote-clique problem: find

$$\mathcal{S} = \arg \max_{\mathcal{S} \subseteq \mathcal{R}, |\mathcal{S}|=n} \sum_{P, P' \in \mathcal{S}} d_G(P, P').$$

2) remote-edge problem: find

$$\mathcal{S} = \arg \max_{\mathcal{S} \subseteq \mathcal{R}, |\mathcal{S}|=n} \min_{P, P' \in \mathcal{S}, P \neq P'} d_G(P, P').$$

3) remote-pseudoforest problem: find

$$\mathcal{S} = \arg \max_{\mathcal{S} \subseteq \mathcal{R}, |\mathcal{S}|=n} \sum_{P \in \mathcal{S}} \min_{P' \in \mathcal{S}, P \neq P'} d_G(P, P').$$

The remote-edge problem is sensitive to the closest-pair of paths, since two solutions with the same closest pair of paths

will have the same minimum distance, even if one solution is much longer than the other [7]. Since all these maximum diversity problems are NP-hard, we use heuristic methods to find a good solution [30].

b) Survivability-based heuristic: We also can use Erickson's and LaValle's notion of survivability [14] in our heuristic. We heuristically find n paths with high survivability and use this set as an initial solution.

3) Path Improvement: The purpose of this phase is to improve k -survivability of an initial solution \mathcal{S} by using local operations: *path replacement* and *path shortening*. Path replacement iteratively replaces one path to improve k -survivability. Path shortening iteratively replaces a subpath of one path to improve k -survivability.

We first apply path replacement to improve k -survivability and then apply path shortening. Since path improvement is a local search method, the search process may be trapped in a local maximum. Thus, when the search reaches a local

TABLE I
RUNNING TIMES MEASURED IN MILLISECOND FOR $k = 1$

Path generation method	Path selection method	Path generation	Path selection	Path replacement	Path improvement	Total time
Random generation method	Random	37.3	0.0	2124.3	17360.9	19533.3
	Remote-clique	31.0	2894.6	1577.1	17329.0	21841
	Remote-edge	29.3	2505.0	1788.5	16791.9	21125.8
	Remote-pseudoforest	28.6	2649.5	1854.2	16441.6	20986.7
	Survivability	27.2	48686.9	1613.3	18391.3	68733.7
Iterative penalty method	Random	430.0	0.0	2475.7	15752.9	18676.3
	Remote-clique	381.9	4344.8	3300.1	18795.2	26841.3
	Remote-edge	384.7	3550.3	3017.6	17998.5	24972.5
	Remote-pseudoforest	379.9	4112.4	2673.5	18782.7	25970.8
	Survivability	380.0	56769.0	1991.2	15978.2	75144.8
	First- n	375.3	0.0	2968.9	18934.4	22305.3

maximum, we use a randomized method to escape from the local maximum and then apply path shortening again.

a) *Path replacement*: Replace one path in the current solution by another path in \mathcal{R} giving the maximum k -survivability for the set; repeat until no further improvement can be made.

b) *Path shortening*: Find the maximum improvement of k -survivability that can be made by replacing one subpath of a path in the current solution by a shortest path on G connecting the endpoints of the subpath. Repeat shortening until no further improvement can be made.

Although path shortening is very effective under the fixed r -disc trap model, path shortening may not be useful in general models. Moreover, for k SP with $k > 1$, shortening just one path at a time may lead to getting trapped in local maxima easily. For example, Figure 5c shows such a case; all four overlapping paths would need to be shortened simultaneously and in the same way to allow the four robots to follow a better route.

c) *Escape from local maxima*: Since path shortening is a local search method, path shortening may get trapped by local maxima. When no path shortening can be made, we randomly pick a path in the current solution and reroute a subpath randomly. Then, run the path shortening method again to reach another local maximum. We repeat this process until T local maxima are found for a threshold value T , and choose the best.

VI. EXPERIMENTAL RESULTS

In this section, we describe several experiments (in simulation) on different heuristic methods, and compare them in terms of computation time and k -survivability. Remember that our heuristic method consists of three phases. We suggest two choices in the path generation phase: random generation (RG) and iterative penalty (IP) methods. We suggest four choices in the path selection phase: remote-clique (RC), remote-edge (RE), remote-pseudoforest (RF), and survivability (SU). Finally, we test two additional methods in the path selection phase:

- 1) random (R): pick n paths in \mathcal{R} uniformly at random.
- 2) first n paths (FN): if the paths are generated by the iterative penalty method, we pick the first n generated paths.

A. Experiment Setup

We used an environment containing 2500 vertices and 80 rectangular obstacles under the fixed r -disc trap model, where $r = 5$ and $p = 0.004$. The environment is shown in Figure 5. We used the heuristic algorithm to find $n = 5$ paths with high k -survivability, for $k = 1 \dots 4$. We generated $w = 100$ paths in the path generation phase and found $T = 3$ local maxima in the path improvement phase.

The heuristic algorithm is implemented in Java and all tests were conducted on a laptop (2010 MacBook Pro) with an Intel Core i5 2.4 GHz CPU and 8GB RAM. We repeated the experiments ten times and took the average of the results.

B. Results

We first show k -survivability of each phase for each method in Figure 6. When k is small, path shortening effectively improves the k -survivability and the iterative penalty method tends to perform better. However, when $k = 4$, path shortening is not effective, since our algorithm only tries to shorten one path at a time but escaping from a local minimal may require shortening several paths at the same time.

We measured the running time for all methods maximizing k -survivability; the running times for each method for $k = 1$ are shown in Table I. The naïve algorithms IP + FN and RG + R are the most efficient methods. This may hint that although k SP is hard in general, k SP under the fixed r -disc trap model may be tractable. For $k > 1$, the naïve algorithm IP + FN is slightly faster than other methods, but we omit the results due to the page limitation.

VII. CONCLUSION & FUTURE WORK

This work is preliminary, and considers only simple k -survivability problems; however, we believe that k -survivability motivates a wealth of interesting practical and theoretical problems. For example, the problem of k -survivability might be reversed to plan defenses against infiltration or attack. Not all applications of k -survivability need be violent. For example, k -survivability can be considered in the context of visibility or stealth, as has turned out to be central in multi-robot

pursuit-evasion games [31]–[33] for search-and-rescue operations. With a model of feedback or communication, we imagine that *k*-survivability might also provide some insights into collaboration and cooperation problems such as those that arise in sports [34] or control of large robot swarms [35].

Several future directions of theoretical research are possible. Continuous-space models might be approached using variational calculus or optimal control techniques [15], [16]. Obstacles such that the risk of a path depends on the distance between the robot and the obstacle, as for paths in mined water [18] are a potential future direction, as are time-dependent obstacles.

REFERENCES

- [1] M. S. Branicky, R. A. Knepper, and J. J. Kuffner, "Path and trajectory diversity: Theory and algorithms," in *Proc. IEEE Int. Conf. Robot. Autom. ICRA*, May 2008, pp. 1359–1364 [Online]. Available: <http://dx.doi.org/10.1109/ROBOT.2008.4543392>.
- [2] R. A. Knepper and M. T. Mason, "Path diversity is only part of the problem," in *Proc. IEEE Int. Conf. Robot. Autom. ICRA*, May 2009, pp. 3224–3229 [Online]. Available: <http://dx.doi.org/10.1109/ROBOT.2009.5152696>.
- [3] C. J. Green and A. Kelly, "Toward optimal sampling in the space of paths," in *Robotics Research*, vol. 66, M. Kaneko and Y. Nakamura, Eds. New York, NY, USA: Springer, 2011, pp. 281–292 [Online]. Available: http://dx.doi.org/10.1007/978-3-642-14743-2_24.
- [4] R. A. Knepper, "On the fundamental relationships among path planning alternatives," Ph.D. dissertation, Carnegie Mellon Univ., Jun. 2011 [Online]. Available: <http://repository.cmu.edu/dissertations/180/>.
- [5] R. A. Knepper, S. S. Srinivasa, and M. T. Mason, "Toward a deeper understanding of motion alternatives via an equivalence relation on local paths," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 167–186, Feb. 2012 [Online]. Available: <http://dx.doi.org/10.1177/0278364911430418>.
- [6] A. H. Quispe, T. Kunz, and M. Stilman, "Generation of diverse paths in 3D environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. IROS*, Nov. 2013, pp. 5994–5999 [Online]. Available: <http://dx.doi.org/10.1109/IROS.2013.6697226>.
- [7] C. Voss, M. Moll, and L. E. Kavradi, "A heuristic approach to finding diverse short paths," in *Proc. IEEE Int. Conf. Robot. Autom. ICRA*, May 2015, pp. 4173–4179 [Online]. Available: <http://dx.doi.org/10.1109/ICRA.2015.7139774>.
- [8] D. Xu, Y. Chen, Y. Xiong, C. Qiao, and X. He, "On the complexity of and algorithms for finding the shortest path with a disjoint counterpart," *IEEE/ACM Trans. Netw.*, vol. 14, no. 1, pp. 147–158, Feb. 2006 [Online]. Available: <http://doi.acm.org/10.1145/1133553.1133565>.
- [9] P. Dell'Omo, M. Gentili, and A. Scozzari, "On finding dissimilar pareto-optimal paths," *Eur. J. Oper. Res.*, vol. 162, no. 1, pp. 70–82, Jun. 2005 [Online]. Available: <http://dx.doi.org/10.1016/j.ejor.2003.10.033>.
- [10] B. Chandra and M. M. Halldórsson, "Approximation algorithms for dispersion problems," *J. Algorithms*, vol. 38, no. 2, pp. 438–465, Feb. 2001 [Online]. Available: <http://dx.doi.org/10.1006/jagm.2000.1145>.
- [11] M. J. Kuby, "Programming models for facility dispersion: The p-dispersion and maximum dispersion problems," *Geogr. Anal.*, vol. 19, no. 4, pp. 315–329, Sep. 1987 [Online]. Available: <http://dx.doi.org/10.1111/j.1538-4632.1987.tb00133.x>.
- [12] I. D. Moon and S. S. Chaudhry, "An analysis of network location problems with distance constraints," *Manage. Sci.*, vol. 30, no. 3, pp. 290–307, Mar. 1984 [Online]. Available: <http://dx.doi.org/10.1287/mnsc.30.3.290>.
- [13] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi, "Heuristic and special case algorithms for dispersion problems," *Oper. Res.*, vol. 42, no. 2, pp. 299–310, Mar. 1994 [Online]. Available: <http://www.jstor.org/stable/171673>.
- [14] L. H. Erickson and S. M. LaValle, "Survivability: Measuring and ensuring path diversity," in *Proc. IEEE Int. Conf. Robot. Autom. ICRA*, May 2009, pp. 2068–2073 [Online]. Available: <http://dx.doi.org/10.1109/ROBOT.2009.5152773>.
- [15] M. Zabarankin, S. Uryasev, and R. Murphey, "Aircraft routing under the risk of detection," *Nav. Res. Logist. (NRL)*, vol. 53, no. 8, pp. 728–747, Dec. 2006 [Online]. Available: <http://dx.doi.org/10.1002/nav.20165>.
- [16] B. Miller, K. Stepanyan, A. Miller, and M. Andreev, "3D path planning in a threat environment," in *Proc. IEEE Conf. Decis. Control Eur. Control Conf. CDC-ECC*, Dec. 2011, pp. 6864–6869 [Online]. Available: <http://dx.doi.org/10.1109/CDC.2011.6160385>.
- [17] E. Boidot, A. Marzuoli, and E. Feron, "Optimal planning strategy for ambush avoidance," in *Proc. Workshop Artif. Intell. Transp.: Advice Interactivity Actor Model. Conf. Artif. Intell. AAAI*, Jan. 2015 [Online]. Available: <http://www.aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10194/10153>.
- [18] L. Babel and T. Zimmermann, "Planning safe navigation routes through mined waters," *Eur. J. Oper. Res.*, vol. 241, no. 1, pp. 99–108, Feb. 2015 [Online]. Available: <http://dx.doi.org/10.1016/j.ejor.2014.08.002>.
- [19] J. Hershberger and S. Suri, "An optimal algorithm for euclidean shortest paths in the plane," *SIAM J. Comput.*, vol. 28, no. 6, pp. 2215–2256, 1999 [Online]. Available: <http://dx.doi.org/10.1137/S0097539795289604>.
- [20] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Auton. Robots*, vol. 33, no. 3, pp. 273–290, Oct. 2012 [Online]. Available: <http://dx.doi.org/10.1007/s10514-012-9304-1>.
- [21] S. D. Eriksson-Bique *et al.*, "Geometric *k* shortest paths," in *Proc. ACM-SIAM Symp. Discrete Algorithms SODA*, Jan. 2015, pp. 1616–1625 [Online]. Available: <http://dx.doi.org/10.1137/1.9781611973730.107>.
- [22] J. Y. Yen, "Finding the *k* shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971 [Online]. Available: <http://dx.doi.org/10.1287/mnsc.17.11.712>.
- [23] J. W. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, no. 2, pp. 125–145, 1974 [Online]. Available: <http://dx.doi.org/10.1002/net.3230040204>.
- [24] H. Jösch, "The shortest route problem with constraints," *J. Math. Anal. Appl.*, vol. 14, no. 2, pp. 191–197, May 1966 [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0022247X66900205>.
- [25] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*. New York, NY, USA: Springer, 1999, vol. 477 [Online]. Available: <http://www.springer.com/us/book/9780792383819>.
- [26] J. P. Rohrer, A. Jabbar, and J. P. G. Sterbenz, "Path diversification for future internet end-to-end resilience and survivability," *Telecommun. Syst.*, vol. 56, no. 1, pp. 49–67, May 2014 [Online]. Available: <http://dx.doi.org/10.1007/s11235-013-9818-7>.
- [27] V. Akgün, E. Erkut, and R. Batta, "On finding dissimilar paths," *Eur. J. Oper. Res.*, vol. 121, no. 2, pp. 232–246, Mar. 2000 [Online]. Available: [http://dx.doi.org/10.1016/S0377-2217\(99\)00214-3](http://dx.doi.org/10.1016/S0377-2217(99)00214-3).
- [28] S. Yuan, S. Varma, and J. P. Jue, "Minimum-color path problems for reliability in mesh networks," in *Proc. IEEE Int. Conf. Comput. Commun. INFOCOM*, Mar. 2005, pp. 2658–2669 [Online]. Available: <http://dx.doi.org/10.1109/INFCOM.2005.1498549>.
- [29] D. Y. C. Chan and D. G. Kirkpatrick, "Multi-path algorithms for minimum-colour path problems with applications to approximating barrier resilience," *Theor. Comput. Sci.*, vol. 553, pp. 74–90, Oct. 2014 [Online]. Available: <http://dx.doi.org/10.1016/j.tcs.2014.04.009>.
- [30] R. Martí, M. Gallego, A. Duarte, and E. G. Pardo, "Heuristics and meta-heuristics for the maximum diversity problem," *J. Heuristics*, vol. 19, no. 4, pp. 591–615, Aug. 2013 [Online]. Available: <http://dx.doi.org/10.1007/s10732-011-9172-4>.
- [31] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics—A survey," *Auton. Robots*, vol. 31, no. 4, pp. 299–316, Nov. 2011 [Online]. Available: <http://dx.doi.org/10.1007/s10514-011-9241-4>.
- [32] R. R. Murphy, *Disaster Robotics*. Cambridge, MA, USA: MIT Press, Feb. 2014 [Online]. Available: <https://mitpress.mit.edu/books/disaster-robotics>.
- [33] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments," *Int. J. Robot. Res.*, vol. 21, no. 4, pp. 345–366, Apr. 2002 [Online]. Available: <http://dx.doi.org/10.1177/02783640230556368>.
- [34] J. Biswas, J. P. Mendoza, D. Zhu, B. Choi, S. D. Klee, and M. M. Veloso, "Opponent-driven planning and execution for pass, attack, and defense in a multi-robot soccer team," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst. AAMAS*, May 2014, pp. 493–500 [Online]. Available: <http://dl.acm.org/citation.cfm?id=2615812>.
- [35] M. Rubenstein, C. Ahler, N. Hoff, A. Cabrera, and R. Nagpal, "Kilobot: A low cost robot with scalable operations designed for collective behaviors," *Robot. Auton. Syst.*, vol. 62, no. 7, pp. 966–975, Jul. 2014 [Online]. Available: <http://dx.doi.org/10.1016/j.robot.2013.08.006>.