

Performance of a Class of Multi-Robot Deploy and Search Strategies based on Centroidal Voronoi Configurations

K.R. Guruprasad^a, Debsish Ghose^b

^a*Department of Mechanical Engineering
National Institute of Technology Karnataka
Surathkal, INDIA*

^b*Guidance, Control, and Decision Systems Laboratory
Department of Aerospace Engineering
Indian Institute of Science, Bangalore, INDIA*

Abstract

This paper considers a class of deploy and search strategies for multi-robot systems and evaluates their performance. The application framework used is a system of autonomous mobile robots equipped with required sensors and communication equipment deployed in a search space to gather information. The lack of information about the search space is modeled as an uncertainty density distribution over the search space. A *combined deploy and search* (CDS) strategy has been formulated as a modification to *sequential deploy and search* (SDS) strategy presented in our previous work. The optimal deployment strategy using Voronoi partition forms the basis for these two search strategies. The strategies are analyzed in presence of constraints on robot speed and limit on sensor range for convergence of trajectories with corresponding control laws responsible for the motion of robots. SDS and CDS strategies are compared with standard greedy and random search strategies on the basis of time taken to achieve reduction in the uncertainty density below a desired level. The simulation experiments reveal several important issues related to the dependence of the relative performances of the strategies on parameters such as number of robots, speed of robots, and their sensor range limits.

Email addresses: krgprao@gmail.com (K.R. Guruprasad),
dghose@aero.iisc.ernet.in (Debsish Ghose)

Key words: Multi-robot search, Voronoi partition, Deploy and search, multi-agent systems, centroidal Voronoi configuration, optimal deployment

1. Introduction

In nature, we observe groups of animals performing a large number of complex tasks by cooperating among themselves in a distributed manner. Each member of the group performs a relatively simple task using information available in its neighborhood leading to emergence of a collective behavior. Flocks of birds, schools of fish, foraging of ants are a few examples of this phenomena. Nature seems to solve complicated problems in simple and elegant ways. Instead of increasing the complexity/intelligence of an individual to solve a complex problem, nature relies on a group of individuals with nominal abilities. The outcome of such cooperation is more robust to failure of a few of the group members. From a more philosophical perspective, the focus is on a society rather than on an individual member.

The problem of searching for targets in unknown environments has been addressed in the literature in the past [18],[31],[5], and [22]. Various search strategies available in the literature have been surveyed in [5]. These fundamental works were mostly theoretical in nature and were applicable to a single agent searching for single or multiple, static or moving, targets. It is likely that the same task can be accomplished more effectively by multiple searchers. But when multiple agents are involved, coordination between them becomes an important issue. Although a centralized controller can solve the problem, it has many shortcomings such as requirement of complete connectivity, large communication effort, etc. Further, failure of the central controller leads to failure of the entire system. As discussed earlier, most biological systems such as ants, birds, fish etc., have distributed local decision making capabilities which, in turn, lead to a useful collective behavior such as swarms, flocks, schools, etc. Agents taking decisions based only on available local information and distributed control law (usually referred to as ‘behavior’ in biological systems), lead to coordination among the agents and result in a meaningful collective behavior. Developments in areas such as wireless communication, autonomous vehicular technology, computation, and sensors, facilitate the use of large number of agents (UAVs, mobile robots, or autonomous vehicles), equipped with sensors, communication equipment, and computation ability, to cooperatively achieve various tasks in a distributed manner.

Distributed multi-agent systems have been shown to achieve and maintain formations, move as flocks while avoiding obstacles, etc., thus mimicking their biological counterparts. They can also be used in applications such as search and rescue, surveillance, multiple source identification, and cooperative transportation. The major advantages of distributed systems are immunity to failure of individual agents, their versatility in accomplishing multiple tasks, simplicity of agents' hardware, and requirement of only minimal local information. At the same time it is important to design distributed control laws that guarantee stability and convergence to the desired collective behavior under limited information and evolving network configurations.

1.1. Related Literature

One class of problems discussed in the literature is that, of optimally locating agents or sensors in the domain of interest and belongs to the class of locational optimization or facility location problems [10, 24]. A centroidal Voronoi configuration is a standard solution for this class of problems [11], where each of the agents is located at the centroid of the corresponding Voronoi cell. Cortes et al. [6, 7] use these concepts to solve a spatially distributed optimal deployment problem for multi-agent systems. A density distribution, as a measure of the probability of occurrence of an event, is used, along with a function of the Euclidean distance providing a quantitative assessment of how poor the sensing performance is, to formulate the problem. Centroidal Voronoi configuration, with centroids of Voronoi cells computed based on the density distribution within the cell, is shown to be the optimal deployment of sensors minimizing the sensory error. The Voronoi partition becomes natural optimal partitioning due to monotonic variation of sensor effectiveness function with the Euclidean distance. Schwager et al. [29] interpret the density distribution of [6] in a non-probabilistic framework and approximate it by sensor measurements. Sujit and Beard [32] present an exploration system for multiple unmanned aerial vehicles (UAVs) navigating through a simulated unknown region that contains obstacles of unknown shape, size, and initial position. They perform Monte-Carlo simulation to analyze the effect on area coverage with changes in number of agents, sensor range, and communication range.

Cooperative search by multiple agents has been studied by various researchers. Enns et al. [12] use predefined lanes prioritizing them with the target probability. The vehicles cooperate to ensure that the total path length

covered by them is minimized while exhaustively searching the area. A dynamic inversion based control law is used to make vehicles track the assigned tracks or lanes while considering the maximum turn radius constraint. Spires and Goldsmith [30] use space filling curves such as Hilbert curves to cover a given space and perform exhaustive search by multiple robots. Vincent and Rubin [33] address the problem of cooperative search strategies for UAVs searching for moving, possibly evading, targets in a hazardous environment. They use predefined swarm patterns with an objective of maximizing the target detection probability in minimum expected time and using minimum number of UAVs having limited communication range. Beard and McLain [4] use dynamic programming methods to develop strategies for a team of cooperating UAVs to visit as many opportunities without collision while avoiding hazards, in a search area which contains regions of opportunity and hazards. The UAVs have the additional requirement that they should stay within communication range of each other. Flint et al. [13] provide a model and algorithm for path planning of multiple UAVs searching in an uncertain and risky environment using dynamic programming approach. The search area is divided into cells and in each cell the probability of existence of a target defined. Pfister [25] uses fuzzy cognitive map to model the cooperative control process in an autonomous vehicle. In [26],[35], and [36] the authors use distributed reinforcement learning and planning for cooperative multi-agent search. The agents learn the environment online and store the information in the form of a search map and utilize this information to compute their trajectory. The agents are assumed to have limited maneuvering ability, sensor range and fuel. In [35] the authors show a finite lower bound on the search time. Rajnarayan and Ghose [27] use concepts from team theory to formulate the multi-agent search problem as a nonlinear programming problem in a centralized perfect information case. The problem is then reformulated in a Linear-Quadratic-Gaussian setting that admits a decentralized team theoretic solution. Dell et al. [8] develop an optimal branch-and-bound procedure with heuristics such as combinatorial optimization, genetic algorithm and local start with random restarts, for solving constrained-path problems with multiple searchers. Jin et al. [17] address a search and destroy problem in a military setting with heterogeneous team of UAVs. Altshuler et al. [2] examine the Cooperative Hunters problem, where a swarm of UAVs is used for searching one or more evading targets, which are moving in a predefined area while trying to avoid a detection by the swarm. Sathyaraj et al. [28] provide a comparative study of path planning algorithm for multiple UAVs used in

team for detecting targets and keeping them in its sensor range. Yatsenko et al. [37] discuss problems dealing with cooperative control of multiple agents moving in a region searching for targets.

In [16] we have proposed a search strategy namely *sequential deploy and search* (SDS) for multiple agents such as UAVs or mobile robots using Voronoi partition with some preliminary results. In this work, an uncertainty density was used to model the lack of information about the search area. Search agents are deployed optimally maximizing a single step search effectiveness and then perform search. Each agent performs search within its Voronoi cell where it is most effective. The optimal deployment was shown to be a variation of centroidal Voronoi configuration, where each agent is located at the centroid of its Voronoi cell with perceived uncertainty density. The material presented in this paper is based on the doctoral thesis [15].

In this paper, we provide a more detailed account of the SDS strategy and propose a modification of the this strategy, named as *combined deploy and search* (CDS) where, the mobile robots perform search in discrete steps, as they move toward the centroids. We compare the performance of SDS and CDS strategies with standard *greedy* and *random* search strategies based on simulation experiments.

1.2. Contribution of the paper

We address a multi-robot search problem where N agents, equipped with sensors, search the space Q , a convex polytope in \mathbb{R}^d , the d -dimensional Euclidean space. The lack of information about the search space is modeled as an uncertainty density distribution $\phi : Q \mapsto [0, 1]$. We denote $P(t) = \{p_1, p_2, \dots, p_N\}$, $p_i \in Q$, with $p_i \neq p_j$, whenever $i \neq j$, as the robot configuration at time t . We formulate a *combined deploy and search* (CDS) strategy for multiple mobile robots as a modification to *sequential deploy and search* (SDS) strategy proposed in our previous work [16]. The fundamental concept is “deploy” and “search”. In each search step, the uncertainty is reduced as

$$\phi_{n+1}(q) = \phi_n(q) \min_i \{\beta(\|p_i - q\|)\}$$

where, n is the search step count, and $\beta : \mathbb{R} \mapsto [0, 1]$ is the function acting as a reduction factor for the uncertainty density ϕ . The function β is such that, $1 - \beta$ is the sensor effectiveness function.

We consider following multi-center objective function, which when maximized, maximizes a single step effectiveness

$$\mathcal{H}_n = \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) [1 - \beta(\|p_i - q\|)] dQ$$

where, n is the search step count, V_i is the Voronoi cell corresponding to the i -th agent/robot. The critical points of the objective function for a given n is shown to be centroids of Voronoi cells with perceived density. Unlike SDS, the agents/robots do not wait till getting optimally deployed to perform search. The critical points are used only to get a direction of motion for robots. We show, that the CDS strategy can reduce the average uncertainty to any arbitrarily small value in finite time. The optimal deployment strategy has been analyzed in presence of some constraints on robot speed and limit on sensor range for convergence of the robot trajectories with the corresponding control laws responsible for the motion of robots.

Simulation experiments are carried out to validate and compare the performance of SDS and CDS with two generic strategies, namely, *greedy* and *random* search. The simulation results indicate that both the proposed search strategies perform quite well even when the conditions deviated from the assumed ones such as sensor range limitations and the CDS strategy leads to somewhat shorter and smoother trajectories than those of the SDS strategy with the same parameters.

1.3. Organization of the papers

The paper is organized as follows. We preview a few mathematical concepts used in this work in Section 2. The multi-robot search problem is discussed in Section 3. Section 4 discusses the *combined deploy and search* (CDS) strategy. The objective function, instantaneous critical points, and control law responsible for robot motion are presented in this section. In Section 5 we impose a few constraints on robot speed and also limit on sensor range, propose control laws and provide convergence results. The CDS strategy is explained with help of illustrative examples in comparison with the SDS strategy in Section 6. A few implementation issues are discussed in Section 7. In section 8, we provide and discuss simulation results and the paper concludes in section 9.

2. Mathematical preliminaries

In this section we preview mathematical concepts from Voronoi partition, LaSalle's invariance principle, and Liebniz theorem used in the present work.

2.1. Voronoi partition

Voronoi partition [34, 9] is a widely used scheme of partitioning a given space and finds applications in many fields such as CAD, image processing and sensor coverage. We use the Voronoi decomposition scheme to partition the search space. Here we briefly preview the concept.

By a *partition* of a set X we mean a collection of subsets W_i of X with disjoint interiors such that their union is X itself. Let $Q \subset \mathbb{R}^d$, be a convex polytope in \mathbb{R}^d , the d -dimensional Euclidean space. We define the configuration space as $\mathbb{P} = \{\mathcal{P} = \{p_1, p_2, \dots, p_N\}\}$, $p_i \in Q$, position of i -th node in Q . The *Voronoi partition* generated by $\mathcal{P} \in \mathbb{P}$ with respect to Euclidean norm is the collection $\{V_i(\mathcal{P})\}_{i \in \{1, 2, \dots, n\}}$ defined as,

$$V_i(\mathcal{P}) = \{q \in Q \mid \|q - p_i\| \leq \|q - p_j\|, \forall p_j \in \mathcal{P}\}$$

The Voronoi cell V_i is the collection of those points which are closest to p_i compared to any other point in \mathcal{P} . The boundary of each Voronoi cell is the union of a finite number of line segments forming a closed C^0 curve. In \mathbb{R}^2 , The intersection of any two Voronoi cells is either null, a line segment, or a point. In a general d -dimensional space, the boundaries of the Voronoi cells are unions of convex subsets of at most $d - 1$ dimensional hyperplanes in \mathbb{R}^d and the intersection of two Voronoi cells is either a convex subset of a hyperplane or a null set. Each of the Voronoi cells is a topologically connected non-null set.

2.2. LaSalle's Invariance principle

Here we state LaSalle's invariance principle [21, 3] used widely to study the stability of nonlinear dynamical systems. We state the theorem as in [23] (Theorem 3.8 in [23]).

Consider a dynamical system in a domain D

$$\dot{x} = f(x), f : D \rightarrow \mathbb{R}^d \tag{1}$$

Let $V : D \rightarrow \mathbb{R}$ be a continuously differentiable function and assume that

- (i) $M \subset D$ is a compact set, invariant with respect to the solutions of (1).

- (ii) $\dot{V} \leq 0$ in M .
- (iii) $E : \{x : x \in M, \text{ and } \dot{V}(x) = 0\}$; that is E is set of all points of M such that $\dot{V}(x) = 0$.
- (iv) N is the largest invariant set in E .

Then every solution of (1) starting in M approaches N as $t \rightarrow \infty$.

Here by *invariant set* we mean that if the trajectory is within the set at some time, then it remains within the set for all time. Important differences of the LaSalle's invariance principle as compared to the Lyapunov Theorem are (i) \dot{V} is required to be negative semi-definite rather than negative definite and (ii) the function V need not be positive definite (see Remark on Theorem 3.8 in [23], pp 90-91).

2.3. Leibniz Theorem and its generalization

The Leibniz Theorem is widely used in fluid mechanics [20], and shows how to differentiate an integral whose integrand as well as the limits of integration are functions of the variable with respect to which differentiation is done. The theorem gives the formula

$$\frac{d}{dy} \int_{a(y)}^{b(y)} F(x, y) dx = \int_a^b \frac{\partial F}{\partial y} dx + \frac{db}{dy} F(b, y) - \frac{da}{dy} F(a, y) \quad (2)$$

Eqn. (2) can be generalized for a d -dimensional Euclidean space as

$$\frac{d}{dy} \int_{\mathcal{V}(y)} F(x, y) d\mathcal{V} = \int_{\mathcal{V}} \frac{\partial F}{\partial y} d\mathcal{V} + \int_{\mathbf{S}} \mathbf{n}(x) \cdot \mathbf{u}(x) F dS \quad (3)$$

where, $\mathcal{V} \subset \mathbb{R}^d$ is the volume in which the integration is carried out, $d\mathcal{V}$ is the differential volume element, \mathbf{S} is the bounding hypersurface of \mathcal{V} , $\mathbf{n}(x)$ is the unit outward normal to \mathbf{S} and $\mathbf{u}(x) = \frac{d\mathbf{S}}{dy}(x)$ is the rate at which the surface moves with respect to y at $x \in \mathbf{S}$.

3. Multi-robot search

In this section we discuss the problem addressed in this paper. N robots perform search operation in an unknown environment. The lack of information is modeled as an uncertainty density distribution over the search space Q . The problem addressed in this paper is that of deploying N robots in Q to collect information, thereby reducing the uncertainty density distribution over Q . The problem formulation is stated formally as

1. $Q \subset \mathbb{R}^d$ is a convex polytope and is the search space.
2. $\phi : Q \mapsto [0, 1]$ defines the uncertainty density function representing lack of information.
3. N robots, equipped with sensors and communication equipment, deploy themselves in Q , and gather information, thus reducing the uncertainty.
4. $P(t) = \{p_1(t), p_2(t), \dots, p_N(t)\} \subset Q$, with $p_i \neq p_j$ whenever $i \neq j$, denotes the configuration of the multi-robot system at time t , $p_i(t)$ denotes the position of the i -th robot at time t . In future, for convenience, we drop the variable t and refer to the positions by just p_i .
5. Sensor's effectiveness at a point reduces with distance from the sensor.
6. Ideally, we are looking for an optimal way of utilizing the robots to acquire complete information about Q , and thus have $\phi(q) = 0, \forall q \in Q$.

During the search operation, sensors gather information about Q , reducing the uncertainty density as,

$$\phi_{n+1}(q) = \phi_n(q) \min_i \{\beta(\|p_i - q\|)\} \quad (4)$$

where, $\phi_n(q)$ is the density function at the n -th iteration; $\beta : \mathbb{R} \mapsto [0, 1]$ is a function of Euclidian distance of a given point in space from the robot, and acts as the factor of reduction in uncertainty by the sensors; and p_i is the position of the i -th sensor. At a given $q \in Q$, only the robot with the smallest $\beta(\|p_i - q\|)$, that is, the robot which can reduce the uncertainty by the largest amount, is active. If any robot searches within its Voronoi cell, then the updating function (4) gets implemented automatically, That is, the function $\min_i \{\beta(\|p_i - q\|)\}$ is simply $\beta(\|p_i - q\|)$, where $p_i \in V_i$.

In the SDS strategy proposed in [16], the agents get optimally deployed before performing search. In order to maximize the search effectiveness in each search step, following objective function was considered to be maximized.

$$\begin{aligned} \mathcal{H}_n &= \int_Q \Delta \phi_n(q) dQ \\ &= \int_Q \max_{i \in \{1, 2, \dots, N\}} \{(|\phi_n(q) - \beta(\|p_i - q\|)\phi_n(q)|)\} dQ \\ &= \int_Q (\phi_n(q) - \min_{i \in \{1, 2, \dots, N\}} \{\beta(\|p_i - q\|)\} \phi_n(q)) dQ \\ &= \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) (1 - \beta(\|p_i - q\|)) dQ \end{aligned} \quad (5)$$

where, n is “deploy” and “search” count, V_i is the Voronoi cell corresponding to the i -th agent, and $p_i \in Q$ is the position of the i -th agent.

The search effectiveness function $\beta : \mathbb{R} \mapsto [0, 1]$ is a non-decreasing function capturing effectiveness of the sensor. Consider

$$\beta(r) = 1 - ke^{-\alpha r^2}, \quad k \in (0, 1) \quad \text{and} \quad \alpha > 0$$

Here, $ke^{-\alpha r^2}$ represents the effectiveness of the sensor which is maximum at $r = 0$ and tends to zero as $r \rightarrow \infty$ and β is minimum at $r = 0$ (effecting maximum reduction in ϕ) and tends to unity as $r \rightarrow \infty$ (Figure 1) (change in ϕ reduces to zero as r increases). Most sensors' effectiveness reduces over distance as the signal to noise ratio increases with the distance. Thus β , which is upside down Gaussian, can model a wide variety of sensors with two tunable parameters k and α .

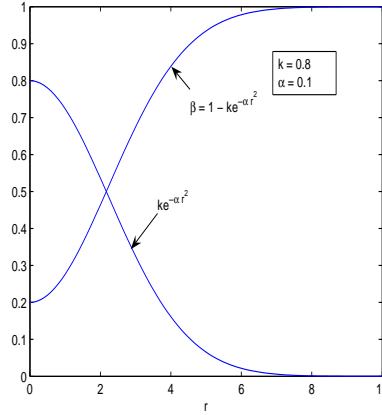


Figure 1: The sensor effectiveness and updating function

The optimal deployment configuration was shown to be a variation of centroidal Voronoi configuration, where each robot is located at the centroid of its Voronoi cell computed with a density $\tilde{\phi}_n(q) = \phi_n(q)ke^{-\alpha r_i^2}$, which is the density as perceived by the sensor, with $\beta(r) = 1 - ke^{-\alpha r^2}$, $k \in (0, 1)$, and $\alpha > 0$, as search effectiveness function.

We use Voronoi partition in formulating the search strategies which along with advantages can cause some computation overhead (see [1] and references therein for time complexity of computing Voronoi partitions). This issue has been addressed in the literature (see [6] and references therein) and there are a few algorithms that efficiently implement Voronoi partition related computations. Also, Voronoi based strategies result in collision free trajectories in a natural way, which is an added advantage.

3.1. Optimal deployment strategy

We have shown that the uncertainty reduction will be maximized in a single step of search, if the agents are located at the centroids of respective Voronoi cells. In SDS, the agents get deployed optimally in this sense before performing search. In this section we will provide the control law to make the agent move toward the centroids and achieve the optimal deployment configuration. Though we have discussed the control law in [16], we provide it here for the sake of completeness and clarity.

Typically search problems do not consider dynamics of search agents as the focus is more on the effectiveness of search, that is, being able to identify region of high uncertainty and distribute search effort to reduce uncertainty. Moreover, it is usually assumed that the search region is large compared to the physical size of the agent or the area needed for the agent to maneuver. In this paper, we assume that the agents are modeled as simple first dynamical systems as

$$\dot{p}_i = u_i \quad (6)$$

Consider the control law

$$u_i = -k_{prop}(p_i - \tilde{C}_{V_i}) \quad (7)$$

Control law (7) makes the agents move toward \tilde{C}_{V_i} for positive control gain, k_{prop} .

We have shown in [16], using LaSalle's invariance principle, that the trajectories of the agents governed by the control law (7), starting from any initial condition $\mathcal{P}(0) \in Q^N$, will asymptotically converge to the critical points of \mathcal{H}_n .

4. Combined deploy and search (CDS) strategy

In the SDS strategy, The robots first get optimally deployed and then search is performed. The “deploy” and “search” steps continue till the uncertainty density is reduced below a desired value. Here the optimal deployment strategy ensures that the uncertainty density reduction is maximized in each search step. But it does not guarantee optimal trajectories of the robot. During the deployment stage, the robots move without utilizing the sensors. Intuitively, it seems that the trajectories will be closer to optimal if, as the robots are moving toward the respective \tilde{C}_{V_i} , they also simultaneously perform the search operation in discrete steps. We define the *latency*, t_s , of

the robots as the maximum time taken to acquire the information, process it, and successfully update the uncertainty density. The time interval between each search should be more than t_s . Here we shall formulate such a strategy and name it *combined deploy and search* (CDS) strategy.

4.1. Density update

Here we provide the problem formulation for the *combined deploy and search* strategy. Assume that the index n represents the intermediate step at which the search is performed and uncertainty density is updated. Using the uncertainty density update rule (4) discussed earlier we can get,

$$\Delta_n \phi(q) = \phi_{n+1}(q) - \phi_n(q) = \phi_n(q) \min_i (1 - \beta(\|p_i - q\|)) \quad (8)$$

$$\Phi_n = \int_Q \phi_n(q) dQ \quad (9)$$

Integrating (8) over Q ,

$$\Delta \Phi_n = \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) (1 - \beta(\|p_i - q\|)) dQ \quad (10)$$

4.2. Objective function

The objective function (5), used for SDS strategy [16], is fixed for each deployment step as $\phi_n(q)$ is fixed for the n -th iteration. In *combined deploy and search*, the search task is performed as the robots move. Now an objective function to be maximized in order to maximize the uncertainty reduction at the n -th search step is

$$\mathcal{H}_n = \Delta \Phi_n = \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) (1 - \beta(\|p_i - q\|)) dQ \quad (11)$$

Note that the above objective function is same as (5) except for the fact that n in this case represents the search step count, whereas in (5) it represents ‘deploy and search’ step count. For $\beta(r) = 1 - ke^{-\alpha r^2}$, the objective function (11) becomes,

$$\mathcal{H}_n = \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) k e^{-\alpha r_i^2} dQ \quad (12)$$

It can be noted that for a given n , the uncertainty density $\phi_n(q)$ at any $q \in Q$ is constant. The gradient is given as (see Theorem A.1 in Appendix),

$$\begin{aligned} \frac{\partial \mathcal{H}_n}{\partial p_i} &= \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) k e^{-\alpha(\|p_i - q\|)^2} (-2\alpha)(p_i - q) dQ \\ &= -2\alpha \tilde{M}_{V_i} (p_i - \tilde{C}_{V_i}) \end{aligned} \quad (13)$$

where \tilde{M}_{V_i} and \tilde{C}_{V_i} are the mass and the centroid of V_i with respect to $\tilde{\phi}_n(q) = \phi_n(q) k e^{-\alpha r_i^2}$, which is the density as perceived by the sensor. The critical points are same as those obtained for SDS. But the uncertainty changes in every time step and hence the critical points also change. Hence, the corresponding critical points are only the instantaneous critical points. It should be noted that the above treatment is valid for any non-decreasing continuously differentiable $\beta(\cdot)$ with $\tilde{\phi}(\cdot)$ depending on exact nature of the function $\beta(\cdot)$. We use the control law

$$u_i = -k_{prop}(p_i - \tilde{C}_{V_i}) \quad (14)$$

Control law (7) makes the robots move toward \tilde{C}_{V_i} for positive control gain, k_{prop} .

The instantaneous critical points and the gradient (13) are used in control law (14) only to make the robots move toward the instantaneous centroids rather than deploying them optimally. Thus, it is not possible to prove any optimality of deployment and we do not prove the convergence of the trajectories here in case of CDS. In CDS, compared to SDS, robots perform more frequent searches instead of waiting till the optimal deployment maximizing per step uncertainty reduction.

To implement the control law, centroid of each Voronoi cell needs to be computed. The computational overhead of computing the centroid can be reduced at the cost of slower convergence using methods reported in the literature such as random sampling and stochastic approximation [19, 14]. In addition, we discretize the search space into grids while implementing the strategy. This simplifies the computation of the centroid of Voronoi cells. The main focus of this paper is design and demonstration of the multi-robot search strategy and finer issues such as computation complexities are beyond the scope of this paper.

It can be shown that the *combined deploy and search* strategy is spatially distributed over the Delaunay graph \mathcal{G}_D . Here by spatially distributed we

mean that information from neighboring robots is sufficient for computation of control input. A Delaunay graph \mathcal{G}_D is an undirected graph, where two agents/robots are said to be neighbors (connected by an edge) if the corresponding Voronoi cells are adjacent.

Theorem 1. *The combined deploy and search strategy can reduce the average uncertainty to any arbitrarily small value in finite time.*

Proof. Consider the uncertainty density update law (4) for any $q \in Q$,

$$\phi_n(q) = (1 - ke^{-\alpha r_i^2})\phi_{n-1}(q) = \gamma_{n-1}\phi_{n-1}(q) \quad (15)$$

where, r_i is the distance of point $q \in Q$ from the i -th robot, such that $q \in V_i$, the Voronoi cell corresponding to it and, $\gamma_{n-1} = (1 - ke^{-\alpha r_i^2})$.

Applying the above update rule recursively, we have,

$$\phi_n(q) = \gamma_{n-1}\gamma_{n-2} \dots \gamma_1\gamma_0\phi_0(q) \quad (16)$$

Let $D(Q) := \max_{p,q \in Q}(\|p - q\|)$. We note that

- (i) $0 < k < 1$
- (ii) $0 \leq r_i \leq D(Q)$. $D(Q)$ is bounded as the set Q is bounded.
- (iii) $0 \leq \gamma_j \leq 1 - ke^{-\alpha\{D(Q)^2\}} = l$ (say), $j \in \mathbb{N}$; and $l < 1$

Now consider the sequence $\{\Gamma\}$,

$$\Gamma_n = \gamma_n\gamma_{n-1} \dots \gamma_1\gamma_0 \leq l^{n+1}$$

Taking limits as $n \rightarrow \infty$,

$$\lim_{n \rightarrow \infty} \Gamma_n \leq \lim_{n \rightarrow \infty} l^{n+1} = 0$$

Thus,

$$\lim_{n \rightarrow \infty} \phi_n(q) = \lim_{n \rightarrow \infty} \Gamma_{n-1}\phi_0(q) = 0$$

As the uncertainty density ϕ vanishes at each point $q \in Q$ in the limit, the average uncertainty density over Q is bound to vanish as $n \rightarrow \infty$. Thus, the average uncertainty density can be reduced to arbitrarily small value in finite time. \square

It can be observed that the above proof does not depend on the control law. The theorem depends only on the outcome of the choice of the updating function (4), along with the fact that there is no sensor range limitation, and that the search space Q is bounded. In addition, the theorem does not address the issue of optimality of the strategy which, in fact, depends on the control law which is responsible for the motion of the robots. Further, unlike SDS, maximal uncertainty reduction is also not guaranteed in each search step. In case of SDS, the reduction in the uncertainty in each step in SDS is

$$\mathcal{H}_n^* = \sum_i \int_{V_i} \phi_n(q) k e^{-\alpha(\|\tilde{C}_{V_i} - q\|)^2} dQ \quad (17)$$

which is the maximum possible reduction in a single step. The deployment is such that uncertainty will be reduced to a maximum possible extent in a step, given by the above formula. whereas in CDS the uncertainty reduction in n -th search step is given by

$$\mathcal{H}_n^* = \sum_i \int_{V_i} \phi_n(q) k e^{-\alpha(\|p_i - q\|)^2} dQ \quad (18)$$

where it is not required that $p_i = \tilde{C}_{V_i}$ while performing search. Though the uncertainty reduction in a given search step n in CDS is less than that in SDS, as will be seen in later sections, the CDS performs better compared to SDS in terms of faster uncertainty reduction due to more frequent searches.

5. Constraints on robot speed and sensor range

In previous sections we have formulated the multi-robot search strategies under ideal conditions and provided a few useful analytical results for convergence and spatial distributedness providing an analytically sound platform for analysis. But in a practical situation these conditions may be violated. It is more likely that the robots will have limit on maximum speed or they may be constrained to move with a constant speed. Further, the sensors could have limit on their range. In this section we analyze the proposed strategies in the presence of speed and sensor range limitations.

5.1. Maximum speed constraint

Let the robots have a constraint on maximum speed of U_{maxi} , for $i = 1, \dots, n$. Now consider the control law

$$u_i = \begin{cases} -k_{prop}(p_i - \tilde{C}_{V_i}) & \text{If } u_i \leq U_{maxi} \\ -U_{maxi} \frac{(p_i - \tilde{C}_{V_i})}{\|p_i - \tilde{C}_{V_i}\|} & \text{Otherwise} \end{cases} \quad (19)$$

The control law (19) makes the robots move toward their respective centroids with saturation on speed.

Theorem 2. *The trajectories of the robots governed by the control law (19), starting from any initial condition $\mathcal{P}(0) \in Q^N$, will asymptotically converge to the critical points of \mathcal{H}_n .*

Proof. Consider $V(\mathcal{P}) = -\mathcal{H}_n$.

$$\begin{aligned} \dot{V}(\mathcal{P}) &= -\frac{d\mathcal{H}_n}{dt} = -\sum_{i \in \{1,2,\dots,N\}} \frac{\partial \mathcal{H}_n}{\partial p_i} \dot{p}_i \\ &= \begin{cases} 2\alpha \sum_{i \in \{1,2,\dots,N\}} \tilde{M}_{V_i}(p_i - \tilde{C}_{V_i})(-k_{prop})(p_i - \tilde{C}_{V_i}), & \text{If } u_i \leq U_{maxi} \\ 2\alpha \sum_{i \in \{1,2,\dots,N\}} \tilde{M}_{V_i}(p_i - \tilde{C}_{V_i})(-U_{maxi}) \frac{(p_i - \tilde{C}_{V_i})}{(\|p_i - \tilde{C}_{V_i}\|)}, & \text{otherwise} \end{cases} \\ &= \begin{cases} -2\alpha k_{prop} \sum_{i \in \{1,2,\dots,N\}} \tilde{M}_{V_i}(\|p_i - \tilde{C}_{V_i}\|)^2, & \text{If } u_i \leq U_{maxi} \\ -2\alpha \sum_{i \in \{1,2,\dots,N\}} U_{maxi} \tilde{M}_{V_i} \frac{(\|p_i - \tilde{C}_{V_i}\|)^2}{(\|p_i - \tilde{C}_{V_i}\|)}, & \text{otherwise} \end{cases} \end{aligned} \quad (20)$$

We observe that

1. $V : Q \mapsto \mathbb{R}$ is continuously differentiable in Q as $\{V_i\}$ depends at least continuously on \mathcal{P} , and \dot{V} is continuous as u is continuous if not smooth.
2. $M = Q$ is a compact invariant set.
3. \dot{V} is negative definite in M .
4. $E = \dot{V}^{-1}(0) = \{\tilde{C}_{V_i}\}$.
5. E itself is the largest invariant subset of E by the control law (19).

Thus, by LaSalle's invariance principle, the trajectories of the robots governed by control law (19), starting from any initial configuration $\mathcal{P}(0) \in Q^N$, will asymptotically converge to the set E , the critical points of \mathcal{H}_n , that is, the centroidal Voronoi configuration with respect to the density function as perceived by the sensors. \square

5.2. Constant speed control

The robots may have a constraint of moving with a constant speed U_i . But we let the robots slow down as they approach the critical points. For $i = 1, \dots, n$, consider the control law

$$u_i = \begin{cases} -U_i \frac{(p_i - \tilde{C}_{V_i})}{\|p_i - \tilde{C}_{V_i}\|}, & \text{if } \|p_i - \tilde{C}_{V_i}\| \geq \delta \\ -U_i(p_i - \tilde{C}_{V_i})/\delta, & \text{otherwise} \end{cases} \quad (21)$$

where, $\delta > 0$, predefined value, such that the control law (21) makes the robots move toward their respective centroids with a constant speed of U_i when they are at a distance greater than δ from the corresponding centroids and slow down as they approach them.

Theorem 3. *The trajectories of the robots governed by the control law (21), starting from any initial condition $\mathcal{P}(0) \in Q^N$, will asymptotically converge to the critical points of \mathcal{H}_n .*

Proof. Consider $V(\mathcal{P}) = -\mathcal{H}_n$.

$$\dot{V}(\mathcal{P}) = \begin{cases} -2\alpha \sum_{i \in \{1, 2, \dots, N\}} U_i \tilde{M}_{V_i} \frac{(\|p_i - \tilde{C}_{V_i}\|)^2}{(\|p_i - \tilde{C}_{V_i}\|)}, & \text{if } \|p_i - \tilde{C}_{V_i}\| \geq \delta \\ -2\alpha \sum_{i \in \{1, 2, \dots, N\}} U_i \tilde{M}_{V_i} (p_i - \tilde{C}_{V_i})/\delta, & \text{otherwise} \end{cases} \quad (22)$$

We observe that

1. $V : Q \mapsto \mathbb{R}$ is continuously differentiable in Q as $\{V_i\}$ depends at least continuously on \mathcal{P} , and \dot{V} is continuous as u is continuous if not smooth.
2. $M = Q$ is a compact invariant set.
3. \dot{V} is negative definite in M .
4. $E = \dot{V}^{-1}(0) = \{\tilde{C}_{V_i}\}$.
5. E itself is the largest invariant subset of E by the control law (21).

Thus, by LaSalle's invariance principle, the trajectories of the robots governed by control law (21), starting from any initial configuration $\mathcal{P}(0) \in Q^N$, will asymptotically converge to the set E , the critical points of \mathcal{H}_n , that is, the centroidal Voronoi configuration with respect to the density function as perceived by the sensors. \square

5.3. Effect of sensor range limits

In reality, it is likely that the sensors will have limitations on their range. The sensors, in addition to having a monotonically decreasing effectiveness with Euclidean distance, might be totally insensitive to signals at distances larger than R , the sensor range limit. It can also be thought of as follows: when the effectiveness falls below, say, a certain threshold value, for all practical purposes, it can be assumed to be ineffective.

In order to incorporate the sensor range limit, we need to modify the objective function (5). We do so by modifying β suitably.

Proposition 1. *Let $\beta(r)$ and $\hat{\beta}(r) = c + \beta(r)$, where c is a real constant, be two sensor detection functions. The corresponding objective functions $\mathcal{H}(\mathcal{P})$ and $\hat{\mathcal{H}}(\mathcal{P})$, respectively, have the same critical points.*

Proof. The objective function with sensor detection function $\hat{\beta}$ is

$$\begin{aligned}\hat{\mathcal{H}}(\mathcal{P}) &= \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) (1 - \hat{\beta}(\|p_i - q\|)) dQ \\ &= \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) (1 - \beta(\|p_i - q\|) - c) dQ \\ &= \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) (1 - \beta(\|p_i - q\|)) dQ - c \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) dQ \\ &= \mathcal{H} - c \sum_{i \in \{1, 2, \dots, N\}} \int_{V_i} \phi_n(q) dQ\end{aligned}$$

The second term not being a function of p_i , we have,

$$\frac{\partial(\hat{\mathcal{H}})}{\partial p_i}(\mathcal{P}) = \frac{\partial(\mathcal{H})}{\partial p_i}(\mathcal{P})$$

□

Consider the objective function with a saturation on β . Let

$$\tilde{\beta}(r) = \begin{cases} \beta(r), & \text{if } r < R \\ \beta(R), & \text{otherwise} \end{cases} \quad (23)$$

The sensor detection function is shown in Figure 2 with dotted curve for $R = 6$. For $r > 6$, the sensor detection function remains fixed at $\beta(6)$.

Consider the objective function

$$\tilde{\mathcal{H}} = \sum_{i \in \{1, 2, \dots, N\}} \int_{(V_i \cap \bar{B}(p_i, R))} \phi_n(q) (1 - \tilde{\beta}(\|p_i - q\|)) dQ \quad (24)$$

It is easy to show that the gradient of the objective function with the new update function $\tilde{\beta}$ is given by,

$$\frac{\partial(\tilde{\mathcal{H}})}{\partial p_i}(\mathcal{P}) = 2\tilde{M}_{(V_i \cap \bar{B}(p_i, R))}(\tilde{C}_{(V_i \cap \bar{B}(p_i, R))} - p_i) \quad (25)$$

where, the mass \tilde{M} and the centroid \tilde{C} are now computed within the region $(V_i \cap \bar{B}(p_i, R))$, that is, the region of Voronoi cell V_i , which is accessible to the i -th robot. The critical points are $p_i = \tilde{C}_{(V_i \cap \bar{B}(p_i, R))}$.

However, in reality, the detection function should become one beyond the range R (that is there is no reduction in uncertainty at points which are at a distance more than R from the sensor. To achieve this, let

$$\begin{aligned} \hat{\beta}(r) &= \begin{cases} \beta(r) + (1 - \beta(R)) = 1 - k(e^{-\alpha r^2} - e^{-\alpha R^2}), & \text{if } r < R \\ 1, & \text{otherwise} \end{cases} \\ &= \tilde{\beta}(r) + (1 - \beta(R)) \end{aligned} \quad (26)$$

Figure 2 shows the sensor detection function $\hat{\beta}$ in solid line for $R = 6$. After $r = 6$, the limit on sensor range, the sensor is ineffective indicated by the value 1. There will not be any reduction in uncertainty density in this region $r > 6$.

Now by Proposition 1, $\hat{\mathcal{H}}(\mathcal{P})$, the objective function corresponding to the detection function $\hat{\beta}$, has the same critical points as those of the objective function $\tilde{\mathcal{H}}(\mathcal{P})$.

The control law making robots move toward the new critical point is

$$u_i = -k_{prop}(p_i - \tilde{C}_{(V_i \cap \bar{B}(p_i, R))}) \quad (27)$$

Remark 4. The control law (27) is spatially distributed under the r -limited Delaunay graph \mathcal{G}_{LD} , for any robot configuration \mathcal{P} .

Theorem 4. *The trajectories of the robots governed by the control law (27), starting from any initial condition $\mathcal{P}(0) \in Q^N$, will asymptotically converge to the critical points of $\hat{\mathcal{H}}$.*

Proof. The proof is based on LaSalle's Theorem similar to those of Theorems 2 and 3 with $V = -\tilde{\mathcal{H}}(\mathcal{P})$. It can be shown that V is continuously differentiable using Theorem 2.2 in [7]. \square

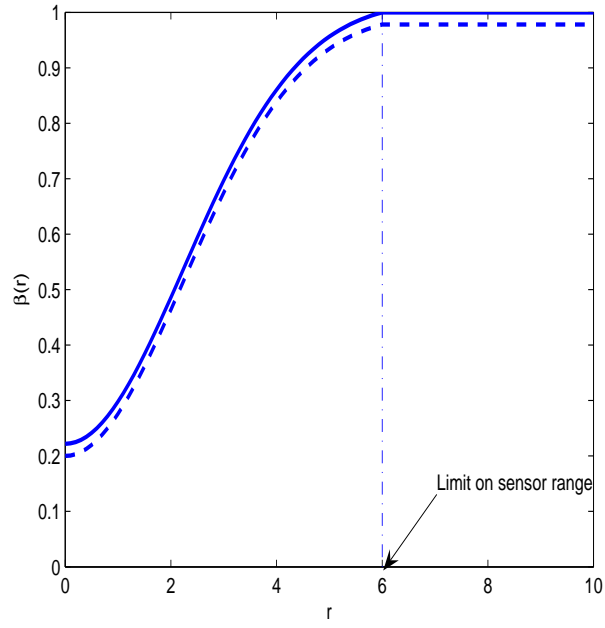


Figure 2: Illustration of $\tilde{\beta}$ and $\hat{\beta}$ in presence of the limit on sensor range. The dotted curve represent the sensor detection function $\tilde{\beta}$ and solid curve is the actual sensor detection function $\hat{\beta}(r) = \tilde{\beta}(r) + (1 - \beta(R))$

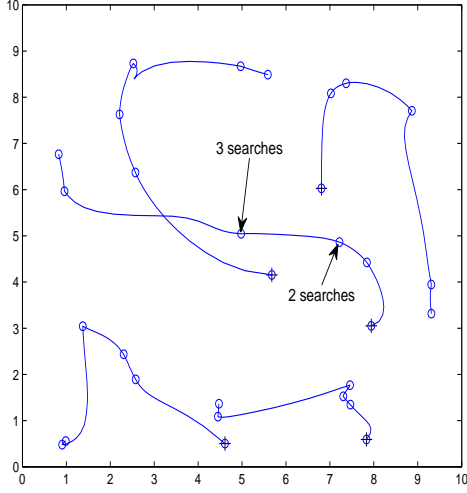
6. Illustrative examples

In this section we show some simulation results to illustrate the CDS strategy in comparison with SDS [16]. More detailed simulation results will be presented in a later section.

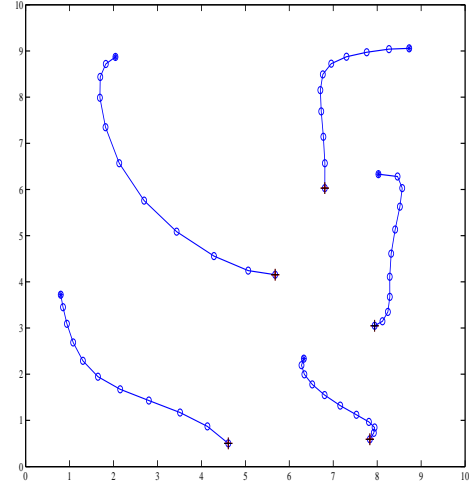
Figure 3 (a) and (b) compares the trajectories of robots with SDS and CDS strategies. The trajectories with CDS are much smoother and shorter. The instances of search are indicated by ‘o’ along the trajectories. It can be seen that the search is performed at every discrete step in CDS, whereas the search is performed only after each optimal deployment SDS. Though there are 8 “deploy and search” steps in SDS, only 5 ‘o’s are visible. In two of steps, multiple searches have been performed as the centroids in successive steps were closer than some tolerance limit $d_{tol} = 0.3$. Thus, there was no movement in corresponding deployment step.

Figure 3(c) compares the history of uncertainty density of SDS and CDS, and it can be observed that the CDS reduces uncertainty relatively faster than SDS, in terms of number of time steps. Figure 3 (d) shows the reduction in average uncertainty density with number of searches for SDS and CDS. It can be observed from this figure that SDS reduces the uncertainty in relatively fewer steps. This is apparent by very concept of optimal deployment in SDS. CDS takes about 4 searches to reduce uncertainty below 0.1, whereas SDS does this in only 3 searches. If SDS requires over 30 time steps to achieve this reduction, CDS needs 4 time steps. We can observe a tradeoff between the number of searches and number of time steps required to accomplish in CDS and SDS. Once the uncertainty reduces to a large extent in initial search steps, by nature of the uncertainty density update rule (4), amount of reduction in subsequent searches is less in both SDS and CDS.

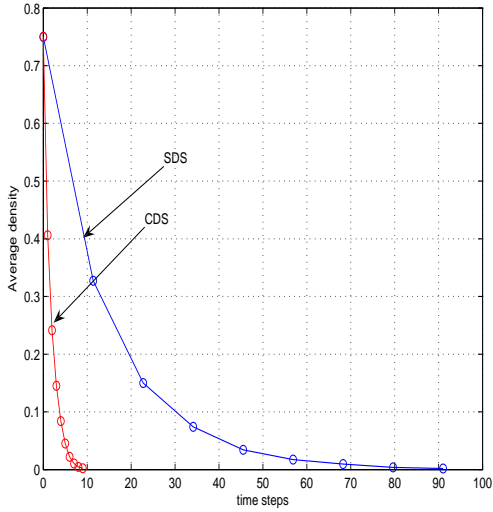
Figure 4(a) illustrates ‘robot 2’ moving toward centroid corresponding to its Voronoi cell with SDS strategy. Robot’s positions are marked with ‘+’ while ‘o’ marks the centroids at successive time instances. Positions of robot in first two time steps are marked as 1 and 2, while those of centroids marked with 1’ and 2’. It can be observed that the robot is tracking the centroid, which is changing as the Voronoi cell is changing. Deployment stops and search is performed when the robot is sufficiently close to the corresponding centroid. One of the search instances is also marked, where, after search, in order to track the next centroid, the robot takes an abrupt turn. This leads to a non-smooth trajectory. Figure 4 (b) illustrates one of the robots moving toward centroid corresponding to its Voronoi cell with CDS strategy. It can



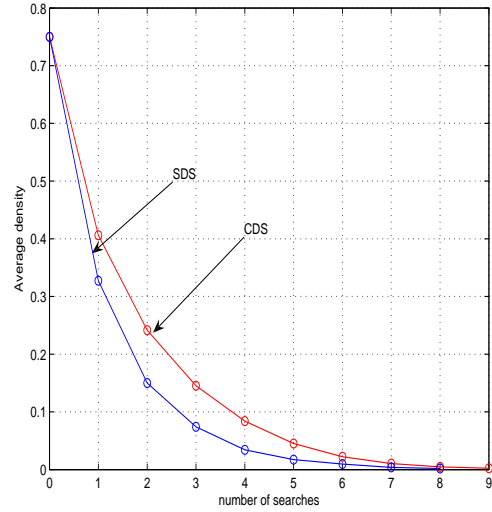
(a)



(b)

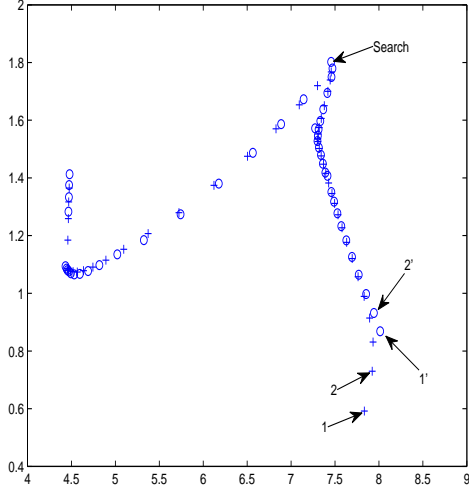


(c)

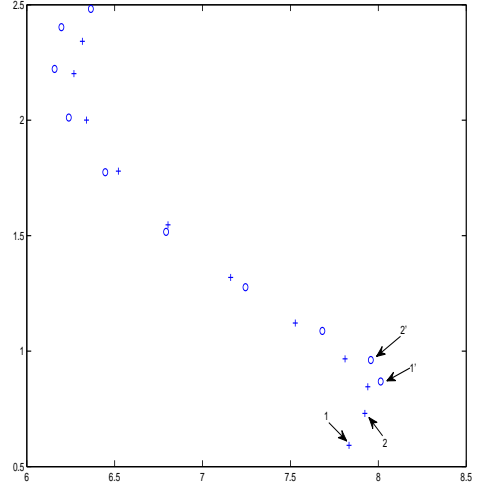


(d)

Figure 3: Trajectories of robots with $N=5$ and without limit on the sensor range for (a) SDS strategy and (b) CDS strategy. In both cases the points marked ‘+’ indicate the starting locations of robots and ‘o’ indicate the end of deployment and points in space where search is being performed. In SDS, at a few places the search is performed more than once. This is indicated in trajectory of ‘robot 1’. The reduction in average uncertainty density is shown in (c) against the number of time steps and (d) against the number of searches, for SDS and CDS. Even in (c) and (d), ‘o’ indicate the search instances.

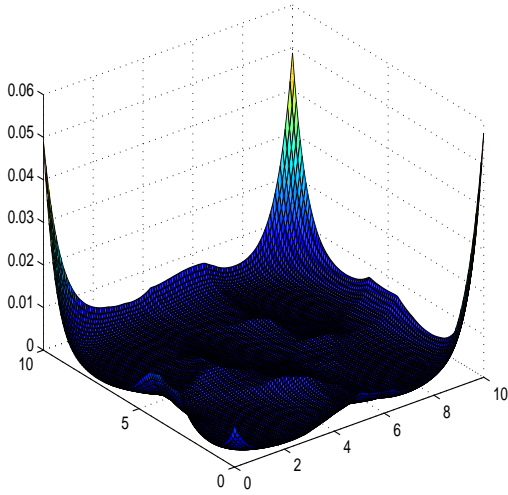


(a)

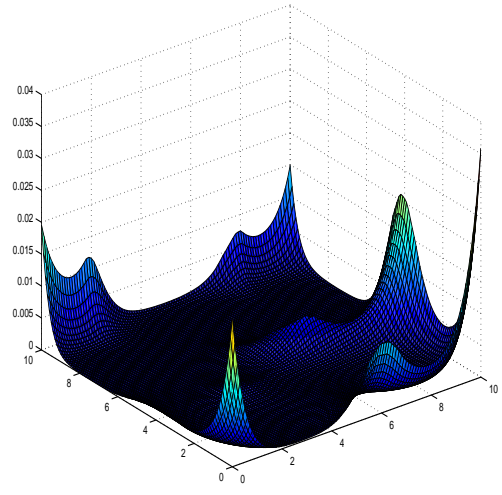


(b)

Figure 4: Figure illustrating the process of a robot, following the respective centroid (a) with SDS and (b) with CDS. In both cases the robot location in each time step is shown by '+', and corresponding centroid locations are shown by 'o'.



(a)



(b)

Figure 5: The final uncertainty density distribution for (a) SDS and (b) CDS strategies, with $N = 5$ and without sensor range limits

be observed that the robot is tracking the centroid, which is changing as the Voronoi cell and the uncertainty density are changing.

7. Implementation Issues

Here we discuss some of the theoretical and implementation issues involved in *combined deploy and search* strategy.

7.1. Spatial distributedness

Here we discuss the implication of spatial distributedness of the proposed search strategies from a practical point of view. We have seen that both the search strategies are spatially distributed in the Delaunay graph. These results imply that all the robots need to do computations based on only local information, that is, by the knowledge about position of neighboring robots. Also, the robots should have access to the updated uncertainty map within their Voronoi cells. This can be achieved in several ways. One such way is that all the robots communicate with a central information provider. But it is not necessary to have this global information. The i -th robot can communicate with its Voronoi neighbors ($(\mathcal{N}_{\mathcal{G}}(i))$) and obtain the updated uncertainty information in a region $\cup_{\mathcal{N}_{\mathcal{G}}(i)} V_i$. As the Voronoi partition $\{V_i\}$ depends at least continuously on \mathcal{P} , the robot configuration [6], in an evolving Delaunay graph, the communication within the neighbors is sufficient for each robot to obtain the uncertainty within its new Voronoi cell. The issues related to communication of uncertainty information are not addressed in the paper except to assume that uncertainty information is available to the robots. It is also possible that the robots can estimate the uncertainty map as done in [29].

In practical conditions, the robots can communicate with other robots only when they are within the limits of the sensor range. The Delaunay graph does not allow sensor range limitations to be incorporated. We need to use r -Delaunay graph \mathcal{G}_{LD} to incorporate the sensor range limitations. The scenario changes with incorporation of sensor range limitations into the search strategies. The updating of uncertainty density will also be within the sensor range limits (in fact, it is within the region $V_i \cap \bar{B}(p_i, R)$). The centroid that is computed will also be within the new restricted area. For an optimal deployment problem, from the perspective of sensor coverage, it has been observed that the corresponding control law is still spatially distributed (in r -limited Delaunay graph) and globally asymptotically stable.

7.2. Synchronization

Synchronization plays an important role in multi-agent systems. Here we discuss this issue for both SDS and CDS strategies. In the case of SDS, theoretically all the robots reach the respective centroid at infinite time. But in a practical implementation, the robots are required to be sufficiently close, where the closeness is suitably defined, to the respective centroids before starting the search operation. It is possible that at any point in time, different robots are at different distances from the corresponding centroid. The robots need to come to a consensus as to when to end the deployment and perform search operation. We have implemented the strategy in a single centralized program using MATLAB. In a practical situation, synchronization can be attained by robots communicating a flag bit indicating if a robot has reached its centroid or not. When all the robots have reached the respective centroid within a tolerance distance, the search can be performed. We also assume that sensing and communication are instantaneous. In our simulation experiments we assume such a communication exists. Since the objective of this work is to evaluate the effectiveness of the search strategies, we make assumptions that simplify implementation without affecting the search effectiveness.

CDS operates in a synchronous manner by design. If all the robots start at the same instant of time and have synchronized clocks, the search task is performed by every robot after the same interval of time. Given an accurate global clock, synchronization is not a major issue in case of CDS.

Further in [6], authors provide an asynchronous implementation for coverage control which can be suitably modified for CDS to operate asynchronously.

8. Results and Discussion

In this section we present simulation experiment results to compare the performance of SDS and CDS strategies with *greedy search* and *random search* tailored to suit the problem setting addressed in this work. *Random* and *greedy* are generic strategies which can be tailored to suit any problem setting. We restrict the robots to move at a constant speed and also discretize the heading direction of robots at 1 degree resolution. Below, we discuss simple strategies used for the purpose of comparing the performance.

Typically search problems do not consider dynamics of search robots as the focus is more on the effectiveness of search, that is, being able to identify

region of high uncertainty and distribute search effort to reduce uncertainty. Moreover, it is usually assumed that the search region is large compared to the physical size of the robot or the area needed for the robot to maneuver. We assume a first order dynamics for the robots for the purpose of simulations.

Greedy search

We discuss two simple *greedy* strategies. In the CDS, the i -th robot moves toward the centroid of $V_i \cap \bar{B}(p_i, R)$ with $\tilde{\phi}$ as density. The Voronoi partition takes care of the coordination among the neighboring robots. In greedy search we simply let the i -th robot move toward corresponding centroid of $\bar{B}(p_i, R)$. Thus there is no cooperation between robots.

According to the update rule (4), only the robot which is most effective will perform search task at any given point $q \in Q$. This leads to the idea of each robot performing search task within its own Voronoi cell. We call such a greedy strategy as *Voronoi greedy search* (VGS) strategy. In this strategy, only the control law is greedy whereas the search is performed in a cooperative manner. In case of cooperative search, each robot discards information about the area which is better accessible to its neighboring robots.

In a true greedy search strategy we expect no cooperation among robots at any point in time. A *true greedy strategy* (TGS) can be achieved if each robot performs search task independently at every point within $\bar{B}(p_i, R)$, leading to duplication of the search task. The update law takes the form,

$$\phi_{n+1}(q) = \phi_n(q) \prod_{\{i|p_i \in \bar{B}(q,R)\}} \beta(\|p_i - q\|) \quad (28)$$

It should be noted that the idea of collecting information by all sensors in contrast to using the information from the most effective sensor is problem dependent. If the sensor is a camera, then many cameras taking picture of the same area may not add to the information.

Random search

Random search (RS) is probably the simplest search strategy. Here we assume the robots move with a constant speed and the direction of the robots is generated randomly. The uncertainty density updating can be given either by (4) or by (28). We have used the update law given by (4) in this case.

In TGS, VGS, and RS strategies, the search task is performed in every time step as in CDS.

Simulation results for performance comparison

The comparison is based on the number of time steps required to reduce the average uncertainty below a specified value.

The parameters which were varied during the simulations are the number of robots (N), the sensor range limit (R), and the speed of robots (U). It was desired that the average uncertainty density should be reduced to a value below 0.002.

Table I compares the performance of the search strategies discussed in this paper in terms of the number of searches performed before the termination condition is reached. The first column gives the parameters N , R , and $100 \times U$ as $N.R.100U$. Thus, 20.4.10 means $N = 20$, $R = 4$, and $U = 0.1$. From Table I, it can be observed that in most cases, the *combined deploy and search* strategy performs better followed by *true greedy* strategy, *Voronoi greedy* strategy, and random search in order of degrading performance as indicated by the number of time steps. In terms of the number of searches being performed, *sequential deploy and search* strategy performs better than *combined deploy and search* strategy.

The numbers in Table I indicate that with an increase in N and R the performance of all the strategies improves. Figure 6(a) illustrates this. In this figure, the points have been interpolated using a shape preserving interpolation scheme.

Performance of CDS strategy deteriorates at higher speeds and lower limit on sensor range as illustrated in Figure 6(b) for *combined deploy and search*. The robots do not move if the centroid of respective $V_i \cap \bar{B}(p_i, R)$ is closer than $U/2$, the tolerance set in the program. With large U and small R , quite often this condition restricts the motion of robots. The condition adversely affects the performance of CDS as the region $V_i \cap \bar{B}(p_i, R)$ is always smaller than the region $\bar{B}(p_i, R)$, which is used by greedy strategies for computing the centroid. That is, in case of CDS, it is more likely that the centroids are closer than $U/2$ to robots. In case of greedy strategies, the robots need to move toward the centroids of corresponding $\bar{B}(p_i, R)$. This region being relatively larger, it is less likely that the robots will be closer to centroids. It has been observed during simulations, particularly with CDS, that many of the robots do not move during the entire search operation. When number of robots is lower, difference in areas of above two regions is more likely to be higher.

Figures 7 (a) and (b) show the time history of the average uncertainty density distribution for *combined deploy and search*, *true greedy*, *Voronoi*

| | CDS | TGS | VGS | SDS search steps (time) | RS |
|----------|------|------|------|----------------------------|-------|
| 5.1.10 | 427 | 615 | 1115 | 193 (1207) | 2000+ |
| 5.1.25 | 338 | 541 | 1416 | 192 (520) | 2000+ |
| 5.1.50 | 1292 | 1150 | 1856 | 193 (261) | 1931 |
| 5.2.10 | 264 | 285 | 250 | 53 (850) | 2000+ |
| 5.2.25 | 99 | 217 | 251 | 53 (359) | 1188 |
| 5.2.50 | 81 | 179 | 267 | 53 (164) | 814 |
| 5.2.100 | 112 | 96 | 209 | 52 (82) | 246 |
| 5.3.10 | 100 | 202 | 538 | 25 (532) | 2000+ |
| 5.3.25 | 76 | 97 | 193 | 26 (239) | 1645 |
| 5.3.50 | 42 | 66 | 177 | 25 (115) | 427 |
| 5.3.100 | 35 | 49 | 127 | 25 (52) | 252 |
| 5.3.150 | 42 | 39 | 144 | 25 (34.7) | 46 |
| 5.3.200 | 42 | 43 | 57 | 25 (26) | 54 |
| 5.3.300 | 132 | 63 | 83 | 25 (17.34) | 56 |
| 5.4.10 | 61 | 61 | 73 | 17 (404) | 2000+ |
| 5.4.25 | 29 | 74 | 87 | 17 (141.6) | 509 |
| 5.4.50 | 33 | 39 | 99 | 17 (77.25) | 230 |
| 5.4.100 | 20 | 24 | 80 | 17 (41.5) | 74 |
| 5.4.150 | 26 | 43 | 216 | 17 (27.6) | 52 |
| 5.4.200 | 28 | 30 | 72 | 17 (20.74) | 54 |
| 5.4.400 | 34 | 36 | 47 | 17 (10.37) | 31 |
| 20.1.10 | 124 | 264 | 429 | 48 (397.6) | 2000+ |
| 20.1.25 | 79 | 200 | 846 | 48 (163.5) | 2000+ |
| 20.1.50 | 96 | 92 | 909 | 48 (81.7) | 2000+ |
| 20.2.10 | 70 | 71 | 109 | 15 (217.6) | 2000+ |
| 20.2.25 | 31 | 43 | 65 | 15 (86.0) | 375 |
| 20.2.50 | 23 | 31 | 123 | 15 (45.0) | 305 |
| 20.2.100 | 25 | 23 | 135 | 15 (22.5) | 73 |
| 20.2.150 | 33 | 22 | 56 | 15 (15.0) | 50 |
| 20.3.10 | 61 | 58 | 64 | 9 (131.1) | 2000+ |
| 20.3.25 | 28 | 28 | 56 | 9 (52.6) | 398 |
| 20.3.50 | 17 | 17 | 70 | 9 (25.8) | 254 |
| 20.3.100 | 14 | 12 | 76 | 9 (12.8) | 48 |
| 20.3.150 | 16 | 11 | 30 | 9 (8.5) | 29 |
| 20.3.200 | 20 | 11 | 22 | 9 (6.4) | 26 |
| 20.3.300 | 38 | 13 | 46 | 9 (4.2) | 20 |
| 20.4.10 | 54 | 50 | 58 | 8 (120) | 1599 |
| 20.4.25 | 25 | 24 | 31 | 8 (38.7) | 615 |
| 20.4.50 | 15 | 15 | 94 | 8 (22.2) | 160 |
| 20.4.100 | 11 | 10 | 77 | 8 (11.2) | 45 |
| 20.4.150 | 11 | 10 | 98 | 8 (6.3) | 30 |
| 20.4.200 | 13 | 12 | 47 | 8 (4.8) | 18 |
| 20.4.400 | 95 | 9 | 30 | 8 (2.4) | 11 |

Table 1: Comparison of performance of various multi-robot search strategies (CDS: Combined deploy and search; SDS: Simultaneous deploy and search; VGS: Voronoi greedy search; TGS: True greedy search; RS: Random search. The first column gives the parameter of simulation experiments as (N).(R).(100U). The numbers indicate the number of search tasks performed before reaching the termination condition. The maximum number of steps was restricted to 2000. The figures within parentheses for the case *sequential deploy and search* are the actual times required for achieving the task.

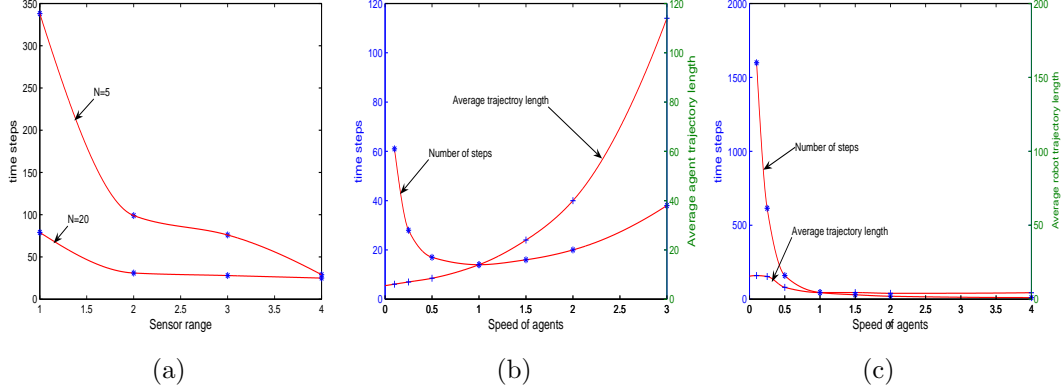


Figure 6: (a) Number of time steps vs. sensor range R for *combined deploy and search* with a speed of 0.25 for different number of robots (b) Number of time steps versus robots' speed U for *combined deploy and search* for 20 robots with sensor range $R = 3$ (c) Number of time steps vs. robots' speed U for *random search* for 20 robots with sensor range $R = 4$.

greedy, and *random* search strategies for two different sets of parameters, one with $N = 5$, $R = 2$ units, and $U = 0.5$ units and another with $N = 20$, $R = 4$ units, and $U = 0.5$ units. Both the figures illustrate that the *combined deploy and search* takes minimum time steps while *true greedy*, *Voronoi greedy*, and *random* search strategies follow in order of degrading performance in terms of number of time steps. It should be noted though all these strategies perform search task in every time step, CDS performs better because of better coverage of the search space due to cooperation among the robots through the Voronoi partition. The CDS reduces uncertainty, in most cases, more than the other strategies (TGS, VGS, and RS) for same number of searches.

Figures 8 (a) and (b) show the trajectories of robots for the above sets of parameters. Figures 9-11 show the robot trajectories with *Voronoi greedy*, *true greedy*, and *random* search strategies, respectively, for the same sets of parameters.

With the greedy strategies, when the parameters are such that the convergence is slower, it is more likely that robots start moving together in clusters. This happens due to lack of cooperation between robots and each of them moving toward points with higher uncertainty density. Once this happens, the robots move together, and in case of *Voronoi greedy* search

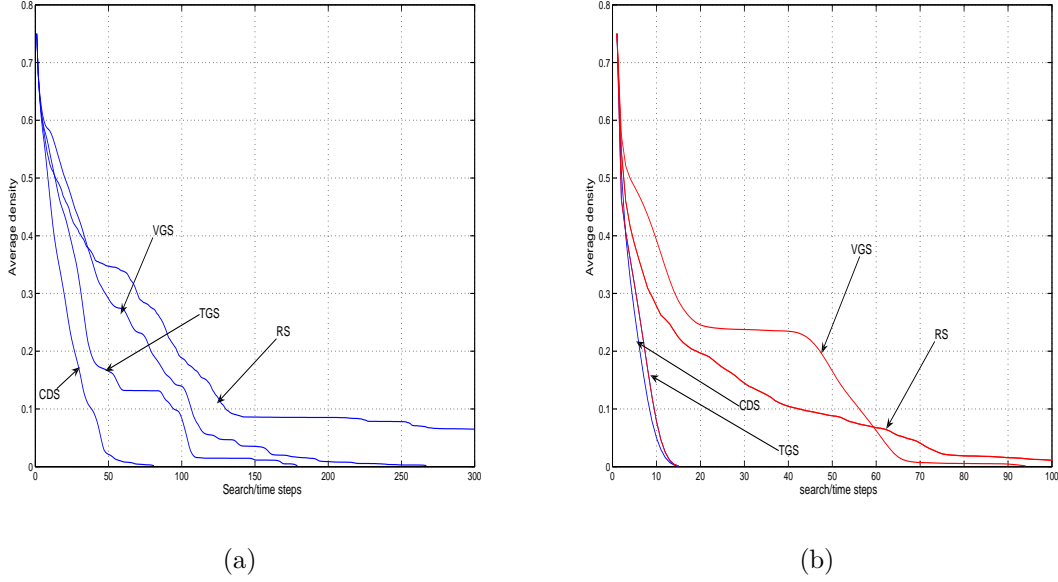


Figure 7: Average uncertainty density distribution with $R = 2$, $U = 0.5$ and (a) with 5 robots and (b) with 20 robots.

where only one of the robot is active at a given point, the basic purpose of deploying multiple robots for search is defeated, leading to a much slower convergence. The strategy is equivalent to a single robot searching the space. With *true greedy* search strategy, though all the robots perform search within their sensor range, only in a very few cases, the performance is comparable to *combined deploy and search*. Cooperation among robots in *combined deploy and search* and *sequential deploy and search* strategies ensure a better coverage of the space and sharing of search load among robots.

As expected, *random search* does not perform well when the speed is less. But as the robot speed is increased, the *random search* performs better. Unlike in other strategies, in case of the *random search* strategy, the performance always increases with speed as illustrated in Figure 6(c). We compute the average length of the trajectory of a robot by multiplying the time, which is equal to the number of steps with the speed. Figure 6(c) also shows the average trajectory length of robots indicating that after a speed of 1 unit, average trajectory length is almost a constant value. It is interesting to note that the *random search* strategy performs better than all other strategies at high robot speeds. This is due to better coverage of the search space by

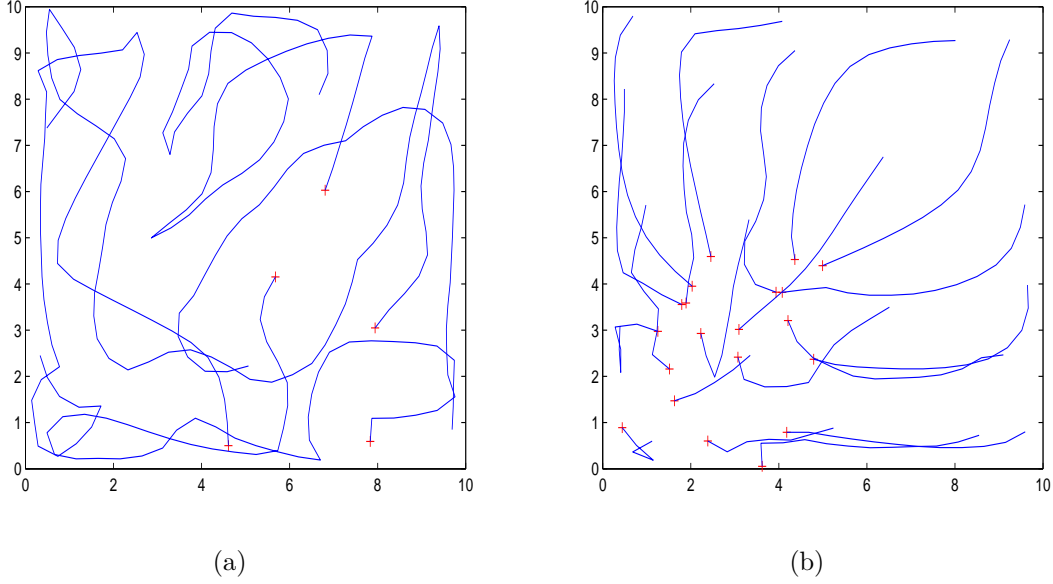
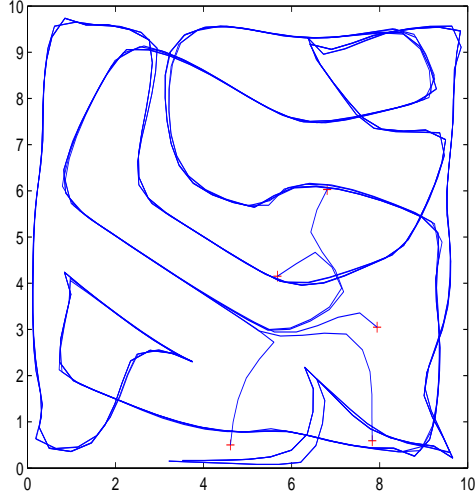


Figure 8: Trajectories of robots with *combined deploy and search* strategy for $U = 0.5$ and (a) 5 robots with sensor range $R = 2$, (b) 20 robots with $R = 4$. '+', indicate initial positions of robots.

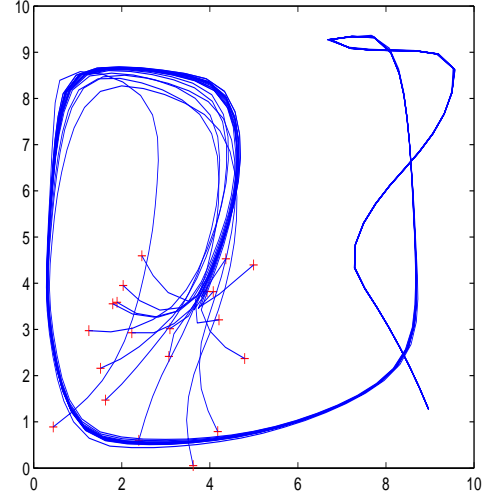
robots. Motion of robots are restricted by the specified control law in case of other strategies, which does not happen in case of *random search*.

In the case of *sequential deploy and search* strategy, as the robots approach the optimal deployment, we allow partial stepping to enable robots to move to as close as possible the respective centroids. Figure 12 shows robot trajectories for the *sequential deploy and search* strategy. With a small sensor range, the time required for completing search is more than that for the rest. But as the sensor range is increased, there is a continuous improvement in the performance leading to the fastest convergence amongst the compared strategies. Note that in the case of the rest of the strategies, the time taken is the same as the number of search steps. The performance of *sequential deploy and search* strategy seems to monotonically increase with N , R and U .

In our formulation of the objective function, we do not attach any cost to the search operation. This is the basic motivation for strategies such as *combined deploy and search*. In such a situation we observe that the *combined deploy and search* is best suited for reducing uncertainty. But in a

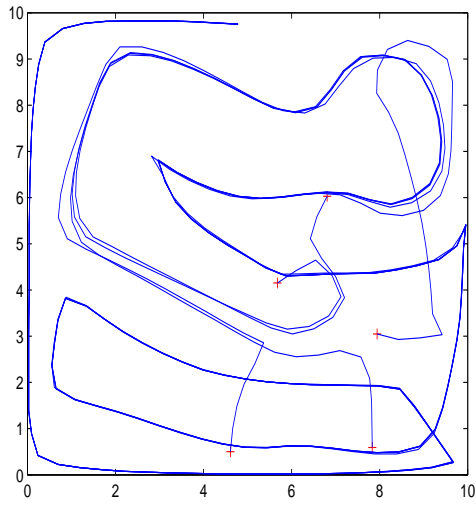


(a)

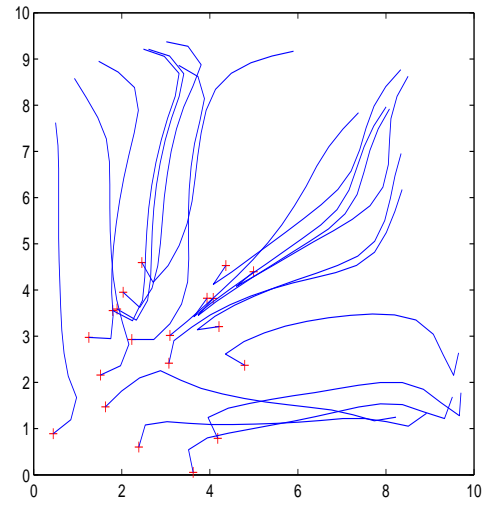


(b)

Figure 9: Trajectories of robots with *Voronoi greedy* strategy for (a) $N = 5$, $R = 2$ and $U = 0.5$ (b) $N = 20$, $R = 4$ and $U = 0.5$. '+' indicate initial positions of robots.

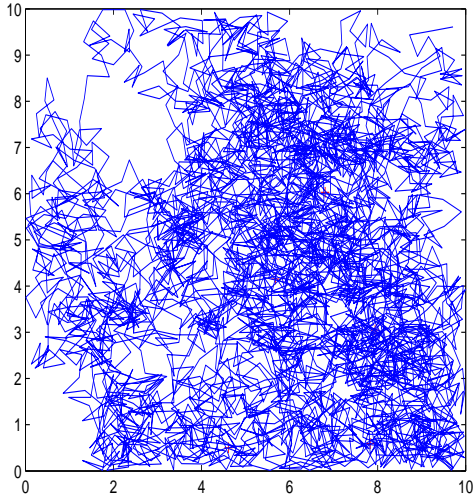


(a)

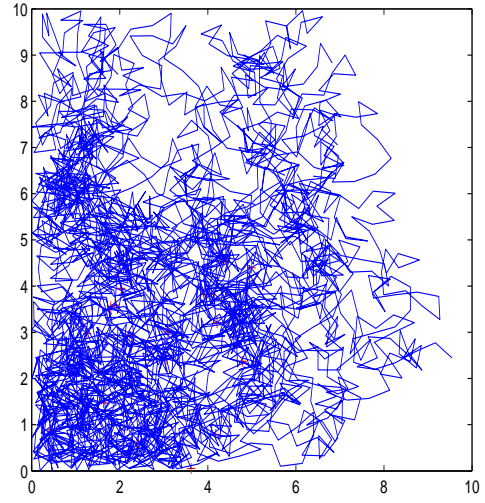


(b)

Figure 10: Trajectories of robots with *true greedy search* strategy for (a) $N = 5$, $R = 2$ and $U = 0.5$ (b) $N = 20$, $R = 4$ and $U = 0.5$. '+' indicate initial positions of robots.

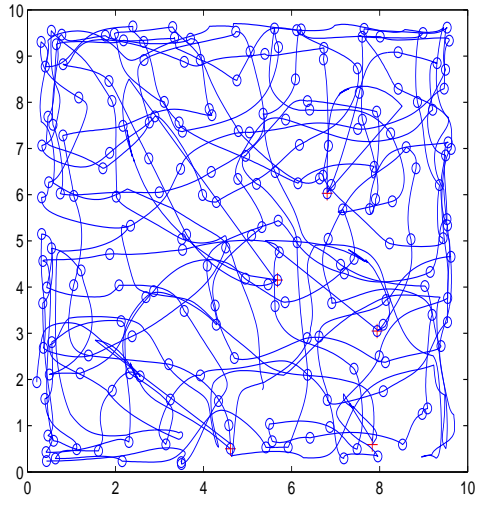


(a)

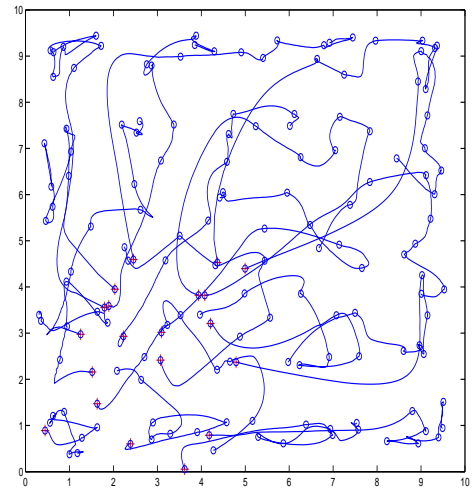


(b)

Figure 11: Trajectories of robots with *random search* strategy for (a) $N = 5$, $R = 2$ and $U = 0.5$ (b) $N = 20$, $R = 4$ and $U = 0.5$. The initial positions of robots are as in Figure 8. The number of points in each trajectory is 814.



(a)



(b)

Figure 12: Trajectories of robots with *sequential deploy and search* strategy for (a) $N = 5$, $R = 2$ and $U = 0.5$ (b) $N = 20$, $R = 4$ and $U = 0.5$. '+' indicate initial positions of robots and 'o' indicate end of deployment and points along the trajectories where search is performed.

practical situation, some cost may be associated with each search operation, in which case *sequential deploy and search* strategy is more efficient. *True greedy* is effective in certain parameter ranges and when in a situation where the duplication of information by more than one sensor is beneficial. *Voronoi greedy* is always below both *combined deploy and search* and *true greedy* in performance in terms of total time.

It can be noted that, though the Voronoi partition based strategies proposed in this paper are computationally complex, their performance is better than the simple strategies such as *true greedy* and *random search*. The Voronoi based strategies result in collision free trajectories in a natural way, whereas non-Voronoi based search strategies need additional collision avoidance schemes and require additional computation.

9. Conclusion and future work

The problem of multi-robot search in an unknown environment with a known uncertainty probability distribution function is addressed in this paper. A multi-robot search strategy namely *combined deploy and search* (CDS) was presented as a modification of *sequential deploy and search* (SDS) strategy proposed in our previous work. We have shown that the centroidal Voronoi configuration with respect to the density as perceived by the sensors are the instantaneous critical points of the objective function maximizing the single step search effectiveness, if the robots are located at corresponding centroids. It was shown, that the CDS strategy can reduce the average uncertainty to any arbitrarily small value in finite time. The optimal deployment strategy has been analyzed in presence of some constraints on robot speed and limit on sensor range for convergence of the robot trajectories with the corresponding control laws responsible for the motion of robots. The proposed CDS strategy was explained with help of a illustrative example, in comparison with SDS strategy.

Based on a set of simulation experiments, the performance of SDS and CDS strategies have been compared with standard strategies such as *greedy* and *random search* with a constant speed constraint on robots and sensor range limit. The parameters N , the number of robots, R , the sensor range limit and U , the speed of robots were varied to compare the performance of different strategies. The simulation results indicated that the *combined deploy and search* strategy is best suited when search task does not involve any cost. The *true greedy* search strategy performed well in a few parameter

ranges and is acceptable only in a situation where gathering of information by more than one sensor in an area makes sense. When a cost is associated with the search task, *sequential deploy and search* strategy was found to be more suitable.

Though the Voronoi partition based strategies such as SDS and CDS are computationally complex, their performance is better than the simple non-Voronoi based strategies such as *true greedy* and *random search*. The Voronoi based strategies result in collision free trajectories in a natural way, whereas non-Voronoi based search strategies need additional collision avoidance schemes and require additional computation.

References

- [1] F. Aurenhammer, Voronoi Diagrams - a survey of a fundamental geometric data structure, *ACM Computing Surveys*, vol. 23, no. 3, 1991, pp. 345-405.
- [2] Y. Altshuler, V. Yanovsky, I. A. Wagner and A. M. Bruckstein, Efficient cooperative search of smart targets using UAV Swarms, *Robotica*, vol. 26, 2008, pp. 551-557.
- [3] E.A. Barbashin and N.N. Krasovski, Ob ustoychivosti dvizheniya v tselom, *Dokl. Akad. Nauk.*, USSR, vol. 86, no. 3, 1952, pp. 453-456.
- [4] R.W. Beard and T.W. McLain, Multiple UAV cooperative search under collision avoidance and limited range communication constraints, *Proc. of IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 25-30.
- [5] S.J. Benkoski, M.G. Monticino, and J.R. Weisinger, A survey of the search theory literature, *Naval Research Logistics*, vol. 38, no. 4, August 1991, pp. 469-494.
- [6] J. Cortes, S. Martinez, T. Karata, and F. Bullo, Coverage control for mobile sensing networks, *IEEE Transactions on Robotics and Automation*, vol 20, no. 2, 2004, pp. 243-255.
- [7] J. Cortes, S. Martinez, and F. Bullo, Spatially-distributed coverage optimization and control with limited-range interactions, *ESAIM: Control, Optimization and Calculus of Variations*, vol. 11 no. 4, 2005 pp. 691-719.

- [8] R.F. Dell, J.N. Eigel, G.H.A. Martins, and A.G. Santos: Using multiple searchers in constrained-path, moving-target search problems, *Naval Research Logistics*, vol. 43, 1996, pp. 463-480.
- [9] G.L. Dirichlet. ber die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal fr die Reine und Angewandte Mathematik*, vol. 40, 1850, pp. 209-227.
- [10] Z. Drezner, *Facility Location: A Survey of Applications and Methods*, New York, NY: Springer, 1995.
- [11] Q. Du, V. Faber and M. Gunzburger, Centroidal Voronoi Tessellations: Applications and Algorithms, *SIAM Review*, vol. 41, no. 4, 1999, pp. 637-676.
- [12] D. Enns, D. Bugajski, and S. Pratt, Guidance and control for cooperative search, *Proc. of American Control Conference, Anchorage, Alaska*, May 2002, pp. 1923-1029.
- [13] M. Flint, E.F.A. Gaucherand, and M. Polycarpou, Cooperative control for UAVs searching risky environments for targets, *Proc. of IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 3568-3572.
- [14] J-C Fort and Page's G. On the a.s. convergence of the Kohonen algorithm with a general neighborhood function, *The Annals of Applied Probability*, vol.5, no. 4, 1995, pp. 1177-1216.
- [15] K.R. Guruprasad, *Multi-agent search using Voronoi partition*, Ph.D. Thesis, Department of Aerospace Engineering, Indian Institute of Science, Bangalore India, December 2008.
- [16] K.R. Guruprasad and D. Ghose, Automated multi-agent search using Voronoi partition, *Conditionally accepted in IEEE Transactions on Automation Science and Engineering*.
- [17] Y. Jin, A. A. Minai, M. M. Polycarpou, Cooperative Real-Time Search and Task Allocation in UAV Teams, *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii USA, December 2003, pp. 7-12.

- [18] B.O. Koopman, *Search and Screening* (2nd Ed.), Pergamon Press, 1980.
- [19] E.B. Kosmatopoulos and M.A. Christodoulou, Convergence properties of a class of learning vector quantization algorithms, *IEEE Transactions on Image Processing*, vol. 5, no. 2, February, 1996, pp. 361-368.
- [20] P.K. Kundu and I.M. Cohen, *Fluid Mechanics* (2nd edition), Academic Press, 2002.
- [21] J.P. LaSalle, Some extensions of Liapunov's second method, *IRE Transactions on Circuit Theory*, CT- vol. 7, no. 4, December, 1960, pp. 520-527.
- [22] K. Lida, *Studies on Optimal Search Plan*, Lecture Notes in Statistics, vol. 70, Springer-Verlag, Berlin, 1992.
- [23] H.J. Marquez, *Nonlinear Control Systems - Analysis and Design*, John Wiley & Sons, Inc., 2003.
- [24] A. Okabe and A. Suzuki, Locational optimization problems solved through Voronoi diagrams, *European Journal of Operations Research*, vol. 98, no. 3, 1997, pp. 445-456.
- [25] H L. Pfister, Cooperative control of autonomous vehicles using fuzzy cognitive maps, *Proc. of the AIAA "Unmanned unlimited" Conference, Workshop, and Exhibit*, San Diego, California, September 2003, AIAA-2003-6506, pp. 1-7.
- [26] M. Polycarpou, Y. Yang and K. Passino, A cooperative search framework for distributed agents, *Proc. of IEEE International Symposium on Intelligent Control*, Mexico, September 2001, pp. 1-6.
- [27] D.G. Rajnarayan and D. Ghose, Multiple agent team theoretic decision making for searching unknown environment, *Proc. of IEEE Conference on Decision and Control*, Maui, Hawaii, December 2003, pp. 2543-2548.
- [28] B. M. Sathiyaraj, L. C. Jain , A. Finn, and S. Drake, Multiple UAVs path planning algorithms: a comparative study, *Fuzzy Optimization and Decision Making*, vol. 3, no. 7, September 2008, pp. 257-267.

- [29] M. Schwager, J. McLurkin and D. Rus, Distributed coverage control with sensory feedback for networked robots, *Proc. of Robotics: Science and Systems*, Philadelphia, PA, Aug 16-19, 2006.
- [30] S. V. Spires and S. Y. Goldsmith, Exhaustive search with mobile robots along space filling curves, *Collective Robotics* (Eds. A. Dragoul, M. Tambe, and T. Fukuda), Lecture Notes in Artificial Intelligence, 1456, Springer-Verlag, 1998 pp. 1-12.
- [31] L.D. Stone, *Theory of Optimal Search*, Academic Press, New York, 1975.
- [32] P.B. Sujit and R. Beard, Multiple UAV exploration of an unknown region, *Annals of Mathematics and Artificial Intelligence*, vol 52, no. 2-4, April 2008, pp. 335-366.
- [33] P. Vincent and I. Rubin, A framework and analysis for cooperative search using UAV swarms, *Proc. of ACM Symposium on Applied Computing*, Nicosia, Cyprus, 2004, pp 79-86.
- [34] G. Voronoi, Nouvelles applications des parametres continus a la theorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik*, vol. 133, 1907, pp. 97-178.
- [35] Y. Yang, A.A. Minai, and M. M. Polycarpou, Analysis of opportunistic method for cooperative search by mobile agents, *Proc. of IEEE Conference on Decision and Control*, Las Vegas, Nevada, December 2002 pp 576-577.
- [36] Y. Yang, A.A. Minai, and M.M. Polycarpou, Decentralized cooperative search in UAVs using opportunistic learning, *Proc. of AIAA Guidance, Navigation and Control Conference*, Monterey, California, August 2002, AIAA pp 2002-4590.
- [37] V.A. Yatsenko, M.j. Hirsch, and Panos M. Pardalos, Cooperative Control of Multiple Agents and Search Strategy, *Lecture Notes in Control and Information Sciences*, vol. 369, 2007, pp. 231-263.

Theorem A.1: The gradient of the multi-center objective function (5) with respect to p_i is given by

$$\frac{\partial \mathcal{H}_n}{\partial p_i} = \int_{V_i} \phi(q) \frac{\partial}{\partial p_i} (1 - \beta(r_i)) dQ \quad (29)$$

where $r_i = \|p_i - q\|$.

Proof. Rewrite (5) as

$$\mathcal{H} = \sum_{i \in \{1, 2, \dots, N\}} \mathcal{H}^i \quad (30)$$

where $\mathcal{H}^i = \int_{V_i} (f(r)) \phi(q) dQ$. Now,

$$\frac{\partial \mathcal{H}_n}{\partial p_i} = \sum_{j \in \{1, 2, \dots, N\}} \frac{\partial \mathcal{H}_n^j}{\partial p_i} \quad (31)$$

where, $f(\cdot) = 1 - \beta(\cdot)$.

Applying the general form of the Leibniz theorem ([20])

$$\begin{aligned} \frac{\partial \mathcal{H}_n}{\partial p_i} &= \int_{V_i} \phi(q) \frac{\partial f}{\partial p_i} (\|q - p_i\|) dQ \\ &+ \sum_{j \in N_i} \int_{A_{ij}} \mathbf{n}_{ij}(q) \cdot \mathbf{u}_{ij}(q) \phi(q) f(\|q - p_i\|) dQ \\ &+ \sum_{j \in N_i} \int_{A_{ji}} \mathbf{n}_{ji}(q) \cdot \mathbf{u}_{ji}(q) \phi(q) f(\|q - p_j\|) dQ \end{aligned} \quad (32)$$

where,

1. N_i is the set of indices of agents which are neighbors of the i -th agent in the Delaunay graph \mathcal{G}_{LD} .
2. A_{ij} is the part of the bounding surface (line segment in two dimensional case) common to V_i and V_j .
3. $\mathbf{n}_{ij}(q)$ is the unit outward normal to A_{ij} at $q \in A_{ij}$. Note that $\mathbf{n}_{ij}(q) = -\mathbf{n}_{ji}(q)$, $\forall q \in A_{ij}$.
4. $\mathbf{u}_{ij}(q) = \frac{dA_{ij}}{dp_i}(q)$, the rate of movement of the boundary at $q \in A_{ij}$ with respect to p_i . Note that $\mathbf{u}_{ij}(q) = \mathbf{u}_{ji}(q)$.
5. Note also that $f(\|q - p_i\|) = f(\|q - p_j\|)$, $\forall q \in A_{ij}$, as $\|q - p_i\| = \|q - p_j\|$, by definition of the Voronoi partition.

By (3)-(5) above, it is clear that for each $j \in N_i$,

$$\int_{A_{ij}} \mathbf{n}_{ij}(q) \cdot \mathbf{u}_{ij}(q) \phi(q) f(\|q - p_i\|) dQ = - \int_{A_{ji}} \mathbf{n}_{ji}(q) \cdot \mathbf{u}_{ji}(q) \phi(q) f(\|q - p_j\|) dQ$$

Hence,

$$\frac{\partial \mathcal{H}_n}{\partial p_i} = \int_{V_i} \phi(q) \frac{\partial f}{\partial p_i}(\|q - p_i\|) dQ$$

□

It can be observed that necessary smoothness conditions are valid as

- (i) $\phi, f \in C^0$.
- (ii) At the boundaries of the Voronoi cells, the objective function does not have any jumps as $f(r_i) = f_j$ at the boundary, by definition of the generalized Voronoi partitions.
- (iii) The Voronoi partition itself varies smoothly with \mathcal{P} .