

Optimal Uncertainty Reduction Search using the k -Shortest Path Algorithm

P.B. Sujit¹ and D. Ghose²

Department of Aerospace Engineering, Indian Institute of Science
Bangalore 560 012, India;
sujit+dghose@aero.iisc.ernet.in

Abstract

We propose a search algorithm for uninhabited autonomous aerial vehicles (UAAVs) performing a search and surveillance task, based on the uncertainty map of an unknown region. The search algorithm is based on the k -shortest path algorithm that maximizes the effectiveness of the search in terms of searching through the maximum uncertainty region, given a constraint on the endurance time of the UAAV. We compare the performance of this algorithm with a random search and a greedy strategy search. We also implement the algorithm for the case of multiple UAAVs searching an unknown region. Simulation results that demonstrate the efficacy of the technique are also presented.

1 Introduction

Searching for targets in an unknown environment has been an area of intense research activity in the search theory literature for several decades [1]. However, much of the later work in this area has been somewhat theoretical in nature. Recent efforts to improve the efficacy of search and surveillance operations using uninhabited autonomous aerial vehicles (UAAVs) has again brought this area of work to the forefront. The basic model that is normally used for these applications is based on a discretization of the search space and time, where the search region is represented as a collection of cells, and the dynamics of the aerial vehicles are thus suppressed in these models [2]. The *a priori* knowledge of the location of the target is represented as an uncertainty map which could be the probability distribution on the cells constituting the search space. The uncertainty values associated with each cell reduces as these cells are visited by a UAAV. Hence, the objective of an optimal route for an UAAV would be to maximize the reduction in uncertainty as it searches the unknown region. The problem, as stated here, is somewhat ill-posed and requires further constraints to make it realistic. In this paper, we propose a strat-

egy for a UAAV to determine optimal search routes for each sortie of the vehicle based on its knowledge of the uncertainty map. We incorporate an endurance time constraint on the UAAV, since UAAVs are small aerial vehicles that have very limited sojourn time before requiring re-fuelling. Simulation results for single UAAV, as well as multiple UAAVs, are presented to demonstrate the efficacy of our algorithm.

2 Problem Formulation

2.1 Discretization of the Search Area

We partition the search region into a collection of identical regular hexagonal cells. We use hexagons as the basic unit since it offers the flexibility to the searcher to move in six uniformly distributed directions at each time step and reach a neighboring cell while expending the same amount of energy (see Figure 1).

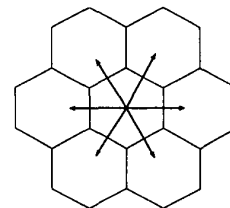


Figure 1: Partitioning into hexagonal cells

2.2 Uncertainty Map

The uncertainty map constitutes of real numbers between 0 and 1 associated with each cell in the search space. These numbers represent the undetected mass in that cell, or it represents the uncertainty with which the location of the target is known in that cell. A probabilistic interpretation of the uncertainty would be as follows: An uncertainty value of 0.6 would imply that any statement about the target's location in cell i (say) would be true only with probability 0.4. An uncertainty value of 0 would imply that everything is known about the cell (that is, one can say with certainty whether a target is located in that cell or not). On the other hand, an uncertainty value of 1 would im-

¹Graduate Research Student

²Associate Professor (Author for correspondence)

ply that nothing can be said about the location of the target in that cell. Hence, a successful search operation is one that manages to visit those cells that have large uncertainty values.

If a cell is visited for a total of n_{k-1} time steps during the $k - 1$ th sortie or mission, then the uncertainty value U_k associated with that cell at the beginning of the next sortie (k th sortie) is given by

$$U_k = \beta^{n_{k-1}} U_{k-1} \quad (1)$$

where, $\beta \in [0, 1)$ is the uncertainty reduction factor associated with each visit to a cell.

2.3 Searcher Objective and Constraints

We assume that there is a single UAAV that is used to search the unknown region. The UAAV carries limited fuel that allows it to fly a limited length of trajectory per sortie and then return to base for refuelling. After refuelling it flies another sortie of the same length. The objective of the UAAV is to maximize the reduction in uncertainty at each sortie. Since we have split the search space uniformly into identically sized cells, we can represent the energy spent by the UAAV in moving from one cell to another as equivalent to one unit step length. We assume that the entire energy of the UAAV per sortie is limited to N (say) number of steps.

The UAAV has to perform several sorties or missions and the number of search missions is unspecified. Each mission consists of one sortie of the search vehicle, where sortie is a trajectory from source cell to a destination cell. We also consider the possibility that the searcher may devote more search effort in a single cell, because of higher uncertainty value in the cell. This would mean that the UAAV may spend more than the duration of one time step to search this cell.

At this point, it is necessary to define what we mean by searching the cell. Each UAAV is equipped with some kind of sensor with which it collects data or information about the cell it is visiting. So a UAAV that spends a certain number of steps (one or more) in searching any given cell is essentially spending this time in collecting data about the cell and thus the uncertainty in that cell reduces as a function of the time spent there.

The updating of the uncertainty map is done only *after* the completion of a sortie or search mission. That is, irrespective of the number of times a cell is visited in any given sortie, the uncertainty associated with the cell is assumed to remain unchanged during the sortie. This condition is valid even if the searcher moves from one cell to another and back to the same cell. The uncertainty map is updated, taking into account the effect of multiple visits, only after the current mission is complete and before the next mission commences. It would be fair to mention that this is a fairly strong

assumption and is not supported by the probabilistic model of the uncertainty map. However, an example of a practical application where this model has a reasonable validity is that of a UAAV equipped with sensors that collect and store data on-board. This data is processed after the sortie is complete and the UAAV has returned to its base. Based on the results of the data processing, the search map is updated.

2.4 A Graph Theoretical Representation of Uncertainty

The algorithm we propose later is based upon a graph theoretical representation of the uncertainty map. In this section we describe how this model is obtained. We consider the center of each hexagonal cell as a *cell node* or a *cell vertex*. The path that an UAAV takes to traverse from one cell to a neighboring cell has to pass through the boundary between these cells. On one side of the boundary the uncertainty value is different from the other side. Since an edge in a graph cannot have two different values, one can either take an average of the two values or create an *intermediate node* between the two cell nodes. We adopt the latter method since it makes it easier to represent looping (that is, an UAAV spending more than one time step in the same cell) by paths that go from a cell node to an intermediate node and back to the cell node.

As an illustration, consider two cells C_1 and C_2 , having cell nodes V_1 and V_2 , with associated uncertainty values U_1 and U_2 (shown in Figure 2). The intermediate node between them is termed as I_1 , the edge (V_1, I_1) has weight $U_1/2$, and the edge (I_1, V_2) has weight $U_2/2$. In the figure, a path segment given by the node sequence I_1, V_2, I_2 or I_1, V_2, I_1 would constitute one time step equivalent of dwell time in the cell C_2 . A path segment I_1, V_2, I_2, V_2, I_1 or I_1, V_2, I_3, V_2, I_3 would constitute 2 time steps equivalent of dwell time in cell C_2 .

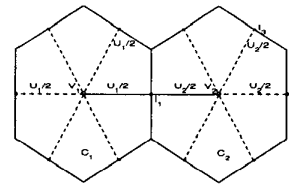


Figure 2: Graph model of the uncertainty map

Finally, the graph $G = (V, E)$ constitutes the model for the search space, where V is the set of cell nodes as well as the intermediate nodes, and E represents the edges between them. Each cell node has a maximum of six edges leading out of it and each intermediate node has a maximum of two edges.

3 Optimization Problem and Solution Method

3.1 The Optimization Problem

The problem of maximum reduction in uncertainty can be transformed to a constrained optimization problem, wherein we need to find a path P that satisfies the following properties: (1) Maximize the sum of the reduction in uncertainty values associated with the cells that constitute the path P for each sortie (2) The path has the constraint that it starts from a source cell and ends at a destination cell, both of which are pre-specified. In fact, in this problem we assume the source cell and the destination cell to be the same. In other words, the UAAV takes off from its base and after the sortie, returns to the base. (3) Because of the fixed fuel capacity, the UAAV has a fixed endurance time, which translates to a fixed number of time steps for which the UAAV can operate. We denote this number as N .

An exact solution to this problem constitutes the solution of a combinatorial optimization problem where all combination of cells that constitute a path of length N is evaluated for reduction in uncertainty. Obviously, this is a time-taking affair even for values of N that are quite small. For example, $N = 10$ would require us to generate $7^{10} \cong 3 \times 10^8$ paths. So, instead of trying to solve this problem exactly, we formulate another problem which is slightly different and then propose a solution that has a much less complexity. In order to do this, the requirement of Item 1 above is changed to "maximize the sum of the uncertainty values associated with the cells that constitute the path P for each sortie". Note that this problem is equivalent to Item 1 above if the number of visits allowed to a cell is just one. Since we do not have any such restriction (as the UAAV is allowed to spend more than one time step in a cell, and even revisit a cell), these two requirements, strictly speaking, are not equivalent.

3.2 Solution Method

Given two nodes S (source node) and T (target node) in a graph $G(V, E)$, where V represents the vertices of the graph and E represents the edges or arcs of the graph, we need to find a path P such that it maximizes the sum of the costs associated with the edges constituting the path, where the path is of fixed length n .

Let $P = (V_1, V_2, \dots, V_n)$, where $V_1 = S, V_n = T, V_i \in V$, and $V_{i+1} \in \mathcal{N}(V_i)$, where $\mathcal{N}(V_i)$ denotes the neighboring nodes of V_i , that is, those nodes that are connected with V_i with an edge.

Hence, we need to find

$$\max_P \sum_{i=0}^{n-1} U(V_i, V_{i+1}) \quad (2)$$

where, $U(V_i, V_{i+1})$ is the uncertainty value associated

with the edge connecting the nodes V_i and V_{i+1} , and \mathcal{P} is the collection of all possible paths of length n from the source node S to the target node T . In principle, this problem can be solved using the combinatorial approach. However, we propose a less complex solution here using some well known shortest path algorithms available in the literature. A problem in the same spirit is the Dijkstra's shortest path problem which finds the shortest path (in terms of minimizing the sum of the weights on the edges of the path) between S and T [3]. However, our problem is different since we have a maximization problem here. First we convert the problem to a minimization one by replacing the uncertainties with certainties. We define $C = 1 - U$ as the certainty value associated with each edge in the graph.

Thus, the new problem formulation is,

$$\min_P \sum_{i=0}^{n-1} C(V_i, V_{i+1}) \quad (3)$$

where, $C(V_i, V_{i+1}) = 1 - U(V_i, V_{i+1})$ is the certainty value associated with the corresponding edge.

Suppose we try using Dijkstra's shortest path algorithm. Then, we have to first remove the constraint on the number of steps n since Dijkstra's algorithm does not admit such constraints. Suppose $P' = (V'_1, \dots, V'_{n'})$ be the corresponding shortest path of length n' . Then the cost associated with path P' is,

$$J(P') = \sum_{i=0}^{n'-1} C(V'_i, V'_{i+1}) = n' - \sum_{i=0}^{n'-1} U(V'_i, V'_{i+1}) \quad (4)$$

If $n' = n$ then this would be the solution to the maximization problem. But there is no guarantee that $n' = n$, especially when the source and target cells are the same, as assumed in this problem.

Hence, we propose the following method, which is based on the k -shortest path algorithm, that gives satisfactory solution. Assume that the minimization problem as posed above is solved using the k -shortest path algorithm [4, 5]. This algorithm produces paths of monotonically increasing costs starting with the shortest path. Consider Table 1. In this table, the path P^1 is the shortest path, followed by paths of increasing costs, so that $J(P^{j+1}) \geq J(P^j)$. In this table, consider the first path, as we go down the table, that has a length of n . We claim that this path is the optimal path for the original maximization problem. This path is also the path with the minimum cost among all paths of length n . Note that there could be several such paths which have the same length n and satisfy the minimum cost criterion. However, we may select any one of them. Let this path be the j -th one in the above sequence. Then,

$$J(P^j) \leq J(P^{j+s}), \quad s = 1, 2, \dots, l \quad (5)$$

Table 1: Results of the k -shortest path algorithm

Path	Length	Cost
P^1	n^1	$J(P^1) = \sum_{i=0}^{n^1-1} C(V_i^1, V_{i+1}^1)$
P^2	n^2	$J(P^2) = \sum_{i=0}^{n^2-1} C(V_i^2, V_{i+1}^2)$
\vdots	\vdots	\vdots
P^k	n^k	$J(P^k) = \sum_{i=0}^{n^k-1} C(V_i^k, V_{i+1}^k)$

where, P^{j+l} is the last path in this sequence which has length n . Then,

$$\sum_{i=0}^{n-1} C(V_i^j, V_{i+1}^j) \leq \sum_{i=0}^{n-1} C(V_i^{j+s}, V_{i+1}^{j+s}), s = 1, \dots, l(6)$$

$$\Rightarrow \sum_{i=0}^{n-1} U(V_i^j, V_{i+1}^j) \geq \sum_{i=0}^{n-1} U(V_i^{j+s}, V_{i+1}^{j+s}), s = 1, \dots, l(7)$$

Thus, P^j is the solution to the original problem. It may so happen that there are no paths of length n that figure in this table. In that case, a solution to the problem does not exist. This can happen, for example, when the specified path length is too small to allow the UAAV to get from S to T . Or else, it can happen if S and T are the same and the n specified is an odd number. However, these are pathological cases and do not detract from the utility of the algorithm.

3.3 The k -Shortest Path Algorithm

The Recursive Enumeration Algorithm (REA) [5] is a k -shortest path algorithm that lists k paths, with increasing costs, connecting a given source-destination pair in a directed graph. The algorithm is recursive, and extends Dijkstra's shortest path algorithm using Bellman's principle of optimality [3]. The problem is formally stated as follows: Given a directed graph (or digraph) $G = (V, E)$. An edge (i, j) connects nodes i and j . As before, each edge (i, j) has a non-negative length, and it is denoted as $C(i, j)$ and the predecessor set of j is denoted as the set $pred(j) = \{i : (i, j) \in E\}$. A path π between two nodes $u, v \in V$, is defined as a sequence of nodes, $\pi = \pi_1.\pi_2.\pi_3 \dots \pi_{|\pi|}$, where $\pi_i \in V, \pi_1 = u, \pi_{|\pi|} = v$, and the edges $(\pi_i, \pi_{i+1}) \in E$ for $1 \leq i \leq |\pi|$, where $|\pi|$ is the number of nodes in the path π . The cost associated with path π is,

$$C(\pi) = \sum_{i=1}^{i=|\pi|} C(\pi_i, \pi_{i+1}) \quad (8)$$

where $C(\pi) = 0$ if $|\pi| = 1$. Now, the problem is to compute the k shortest paths from the source node $s \in$

V to a destination node $t \in V$. For every node $v \in V$, the k -th shortest path from s to v is denoted by $\pi^k(v)$ and the cost associated with it is $C^k(v) = C(\pi^k(v))$. Define $\mathcal{P}^k(v)$ as the set of the k shortest paths starting at s and ending at v . Note that it contains all the paths starting from the shortest to the k -th shortest. This is the set that we want to create.

The problem is solved recursively and in two parts. First, $C^1(v)$ (the cost associated with the shortest path from source node to a node v) and $\pi^1(v)$ (the shortest path from source node to node v) are computed for all v in V using the shortest path algorithm. Next, we use the recursive relation obtained from the generalized Bellman's equation to obtain the the cost of the k -th shortest path. This recursive equation is given as,

$$L^k(v) = \begin{cases} 0, & \text{if } (k = 1) \text{ and } (v = s) \\ \min_{u \in pred(v)} L^{p(u,v,k)}(u) + C(u, v), & \text{otherwise} \end{cases} \quad (9)$$

where

$$p(u, v, k) = \begin{cases} 1, & \text{if } k = 1 \\ p(u, v, k-1) + 1, & \text{if } (k > 1) \text{ and} \\ & (\pi^{k-1}(v) = \pi^{p(u,v,k-1)}(u).v) \\ p(u, v, k-1), & \text{otherwise} \end{cases} \quad (10)$$

Note that $p(u, v, k)$ is the number of paths in $\mathcal{P}^{k-1}(v)$ that ends with the edge (u, v) , plus one. Finally, the k -th shortest path is obtained from (9) as:

$$\pi^k(v) = \pi^{p(\hat{u}, v, k)}(\hat{u}).v \quad (11)$$

where,

$$\hat{u} = \arg \min_{u \in pred v} L^{p(u,v,k)}(u).v \quad (12)$$

3.4 Application to Multiple UAAVs Case

Normally, teams of UAAVs are used for search and surveillance task. The search algorithm given above is for a single searcher, but can be easily used for multiple UAAVs that carry out this task in a cooperative fashion as described in [2]. The initial uncertainty map is assumed to be available to all the UAAVs (which may be located at different bases in the search region). Each UAAV would determine the optimal route for its first sortie using the search algorithm. At the end of the sortie, the base stations cooperatively exchange the data collected by individual UAAVs and use the collective data to update the search map for the next sortie. If the information content (or data) of all the base stations are the same (which occurs when there is noiseless and fully connected communication between all the base stations) then all the UAAVs implement the algorithm using the same uncertainty map. On the other hand, if there is only partial connectivity or noisy communication, the UAAVs may have different search maps.

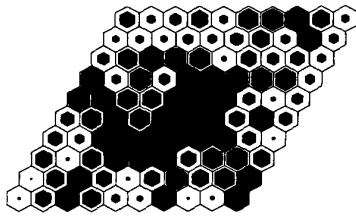


Figure 3: Initial uncertainty map

4 Simulation Results

For the purpose of simulation we consider a region composed of a 10×10 grid of hexagonal cells. The initial uncertainty map is created by generating random numbers between 0 and 100 (thus representing uncertainty as a percentage). This is shown in Figure 3. The uncertainty is shown as the grey area within the cell. The percentage of uncertainty in a cell is proportional to the size of the grey area in the cell. The position of the base station is marked with a '•' in the figure. Initially, we assume a single UAAV that flies a total of 18 sorties through this region. Each sortie is for a limited length of 16 steps through the graphical model, which means that the UAAV flies for 8 steps in the hexagonal grid, which in turn implies that it can reach only 4 cells deep from its base station since it has to come back to the base after a sortie. Using the algorithm proposed in this paper, Figure 4(a) shows the first two sorties of the UAAV. Note that the routes shown for the UAAV shows as if the UAAV goes up to the boundary of a cell and comes back again. This is actually the interpretation of the route as obtained from the graphical model. In reality, these path segments should be interpreted to mean that the UAAV remains in the cell for more than one time step duration. In Figure 4(b) we show all the 18 sorties of the UAAV. Figure 5 shows the uncertainty map at the end of 5, 10, 14, and 18 sorties. This figure gives a visual representation of how the uncertainties fade with increasing number of sorties. To illustrate the efficacy of our algorithm, we compare its performance with a random search route selection algorithm and a greedy search strategy. In the random strategy, the UAAV selects its next cell for search by generating an uniformly distributed random number for the initial 4 steps on the hexagonal grid, and then it retraces its

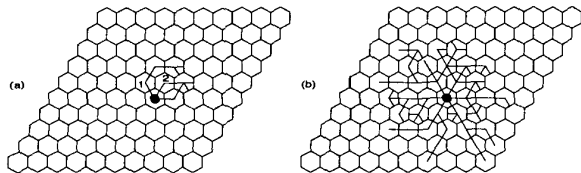


Figure 4: (a) First two sorties (b) All sorties

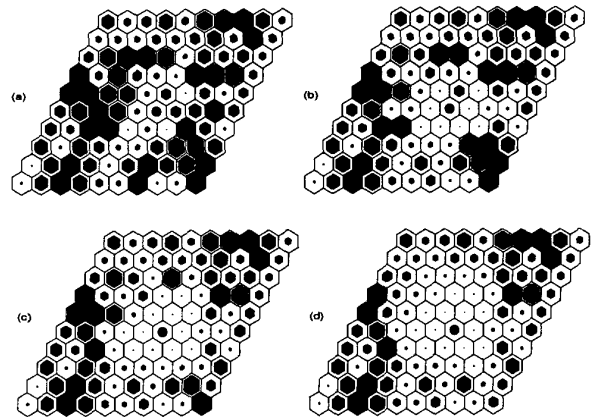


Figure 5: Uncertainty map after (a) 5 sorties (b) 10 sorties (c) 14 sorties (d) 18 sorties

path back to the base. The final uncertainty map for the random search strategy after 18 sorties is shown in Figure 6(a). In the greedy search strategy, the UAAV selects the next cell in its path by choosing the neighboring cell with the maximum uncertainty. This is done for the initial 4 steps on the hexagonal grid, and then the UAAV retraces its path back to the base. The final uncertainty map for the greedy search strategy after 18 sorties is shown in Figure 6(b). In Figure 7 we show the reduction in average percentage uncertainty value for the search region with increasing number of sorties for the three search algorithms. It clearly shows the superiority of the k -shortest path algorithm.

Finally, we apply our search algorithm to multiple UAAVs. The search region is the same but we have three UAAVs which use the same algorithm and cooperatively update their uncertainty maps based on information received from the other UAAVs at the end of each sortie. The simulations are done for 12 sorties each by the 3 UAAVs, each of which is assumed to have an endurance of 12 steps in the graph model (that is, an endurance of 6 steps in the hexagonal grids which implies that UAAVs can search 3 cells deep from their base stations). The uncertainty maps after 3, 6, 9, and 12 sorties are shown in Figure 8. The '•' indicates the positions of the base stations. Figure 9 shows the reduction in average uncertainty with increasing number

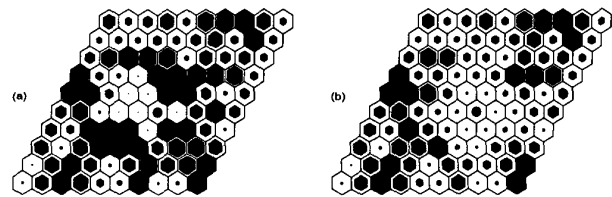


Figure 6: Uncertainty map after 18 sorties for (a) Random search strategy (b) Greedy strategy

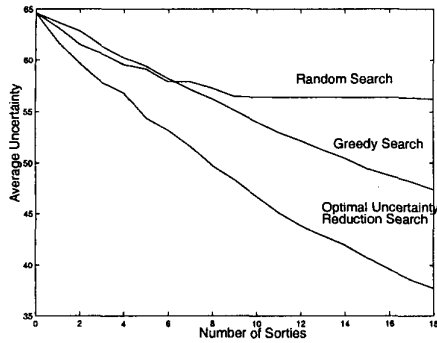


Figure 7: Reduction in average uncertainty

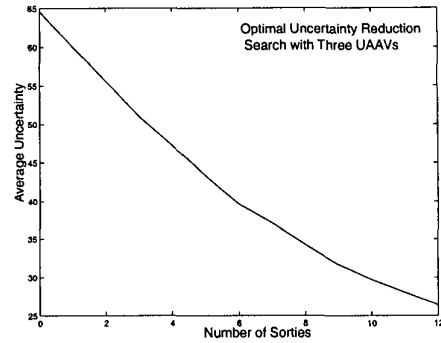


Figure 9: Reduction in average uncertainty for the three UAAVs case

of sorties by the three UAAVs.

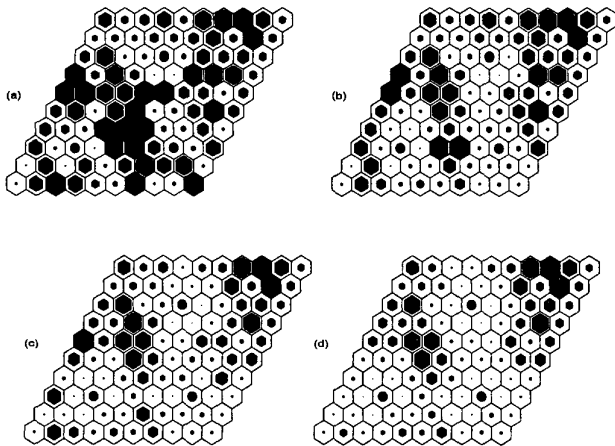


Figure 8: Uncertainty map for three UAAVs after (a) 3 sorties (b) 6 sorties (c) 9 sorties (d) 12 sorties

5 Conclusions

We proposed a search algorithm based on the well-known k -shortest path algorithms used in network flow applications, for use in UAAVs searching for targets in unknown environments. The results, compared with standard random search and greedy strategies, show the superiority of the proposed algorithm. Its utility in a cooperative search using multiple UAAVs was also demonstrated. However, it was assumed that the UAAVs communicate with each other instantaneously and without any loss in information. In reality there could be appreciable delays in the process of UAAVs updating their information or communicating them to other UAAVs [6]. An assumption that needs reconsideration is that of the uncertainty map remaining unchanged during a sortie, because of which the search path is optimal in terms of maximizing the total uncertainty in a path, but is sub-optimal in terms of maximum reduction in uncertainty in a given path.

However, this would require a non-trivial modification of the k -shortest path algorithm to a case when the edge weights change as a function of the number of times that the searcher passes through that edge.

Acknowledgements

The authors would like to acknowledge the financial support received from the IISc-DRDO program on advanced research in mathematical engineering. The authors would also like to acknowledge helpful comments received from D. Eppstein and A. Marzal on the use of their k -shortest path algorithm.

References

- [1] 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.
- [2] K. Passino, M. Ploycarpou, D. Jacques, M. Pachter, Y. Liu, Y. Yang, M. Flint, and M. Baum, Cooperative control for autonomous air vehicles, In *Proceedings of the Cooperative Control and Optimization Workshop*, Florida, December 2000.
- [3] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin: *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, NJ, 1993.
- [4] D. Eppstein: Finding the k -shortest paths, *SIAM Journal of Computing*, Vol. 28, 1998, pp. 652-673.
- [5] V. Jiminez and A. Marzal: Computing the k shortest paths: A new algorithm and an experimental comparison, in *Algorithm Engineering*, J.S. Vitter and C.D. Zaroliagis (eds.), Springer-Verlag, Lecture Notes in Computer Science: 1668, pp. 15-29, 1999.
- [6] D. Ghose, J.S. Shamma, and J.L. Speyer, Cooperative search using teams of partially connected autonomous agents: Information latency and search strategies, Technical Report, Department of Mechanical and Aerospace Engineering, University of California at Los Angeles, July 2002.