*I.J. Information Technology and Computer Science,* **2017, 5, 61-70**
Published Online May 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2017.05.08

# Strategies for Searching Targets Using Mobile Sensors in Defense Scenarios

**Tanmoy Hazra, CRS Kumar, Manisha J. Nene**
Department of Computer Science and Engineering, Defence Institute of Advanced Technology, Pune 411025, India
E-mail: tanmoyhazra316@gmail.com, suthiksnkumar@diat.ac.in, mj.nene@diat.ac.in

***Abstract***—Target searching is one of the challenging research areas in defense. Different types of sensor networks are deployed for searching targets in critical zones. The selection of optimal strategies for the sensor nodes under certain constraints is the key issue in target searching problem. This paper addresses a number of target searching problems related to various defense scenarios and introduces new strategic approaches to facilitate the search operation for the mobile sensors in a two-dimensional bounded space. The paper classifies the target searching problems into two categories: preference-based and traversal distance based. In the preference based problems, the strategies for the mobile sensors are determined by Stable Marriage Problem, College Admission Problem, and voting system; they are analyzed with suitable examples. Alternatively, traversal distance based problems are solved by our proposed graph searching approaches and analyzed with randomly chosen examples. Results obtained from the examples signify that our proposed models can be applied in defense-related target searching problems.

***Index Terms***—Mobile Sensors, Stable Marriage Problem, College Admission Problem, Schulze Method, Target Searching.

## I. INTRODUCTION

Target searching has been attracting a lot of attention from the researchers of various fields such as defense, surveillance, medical, traffic control etc. for several decades. Different types of targets are searched in various search spaces (bounded/unbounded, 2/3 dimensional) under certain constraints. Alternatively, the searchers involved in different search operations can be robots, mobile/immobile sensor nodes, UAVs etc. Based on the nature/functionality, the searchers can be classified into two categories: homogeneous and heterogeneous. Heterogeneous searchers can be deployed for the same search operation. In this section, we have discussed a number of target searching problems where most of the problems have been modeled by game theory (GT) based techniques. GT is an area of Applied Mathematics, used to formulate and select the strategies in order to compute the expected payoffs of the respective players of the game. A game is basically a strategic interaction among the players in a competitive environment and the basic analogies of a game are players, actions, strategies, and payoffs. There are different types of games in GT such as normal-form, extensive-form, cooperative, non-cooperative, Bayesian, stochastic, combinatorial, finitely-long, infinitely-long, zero-sum, non-zero sum, differential, search games etc. In this section, we have mainly focused on different GT-based target searching problems using wireless sensor networks (WSNs). There are several challenging areas in WSNs such as target searching/tracking, topology control, routing protocol, energy saving, power control etc., which are handled by GT. GT is one of the potential tool used to solve several security-related problems in WSNs: preventing DoS attacks, intrusion detection, detecting malicious nodes etc [1-3].

We have discussed a number of target searching, path planning problems including preference-based searching problems using UAVs, WSNs, robots etc. using different mathematical models (mainly focusing upon GT) and the discussed problems that are relevant to our proposed work. In [4], multiple agents aim to independently achieve their goals but multi-agent plan coordination is maximized by cooperatively executing their actions such as avoiding cross events or duplicating action. Algorithmic approaches show that the multi-agent technique is better than most advanced single-agent plan coordination technique. The work in [5] addresses a multi-target multi-searcher search and rescue path planning problem, which finds optimal solution using mixed-integer linear programming. A multi–target multi-robot system in a dynamic environment is solved using a game model in which a rough priori probability map about the target location distribution in a region is given to the searchers [6]. A GT-based mobile object trapping system is introduced in which the mobile target is captured by the mobile sensors by forming coalitions and sharing significant information, thus minimizing the capturing time [7]. In [8], different types of approaches are introduced for evacuation of pedestrians during emergency operation. The problem is modeled as an extension of dynamic network flow and heuristic methods, whose deployment and performances are compared with existing techniques. In [9], the sequence-alignment (SA) distance is introduced as an alternative solution for rank aggregation and partial rank comparison. This distance measure technique considers different sequence characteristics to determine an aggregated rank. In [10], an incomplete information pursuit-evasion game for multiple players is presented and the strategies for the searchers are formulated where the searchers aim to detect/capture multiple intelligent targets and also determine capture time to analyze the performance of the proposed model. The work in [11] discusses a sequential probabilistic decision problem which helps the searchers to detect the target in a region and formulate efficient and robust search control strategies.

A strategic approach based on probabilistic search is discussed that involves multiple observations of the grid cells. The surveillance regions completely/partially cover multiple grid cells and also consider the changes in UAV altitude [12]. The strategies for multi-UAV cooperative search are addressed based on cumulative target existence probabilities and shows that the cooperation and types of shared information determine detection errors and search time. The simulation results show that cumulative information reduces the search time from 27% to 70% while the number of UAVs is increased from 2 to 5 [13]. A novel GT-based technique is presented to optimize a network of fixed multi-static sonar sensors under the sea to detect submarine threats [14]. The work in [15] discusses two well-known game theoretic approaches: Stable Marriage Problem (SMP) and College Admission Problem (CAP) with suitable examples. These approaches are used to match the objects of two same or different sized sets based on their preference ranking. The basic differences between the SMP and the CAP are as follows: two equal sized sets and one-to-one mapping between the sets are required for the SMP whereas two unequal sized sets and many-to-one mapping between the sets are required for the CAP. There doesn't exist any stable matching method for the CAP, which makes it a dominant strategy for all members to disclose their true preferences [16]. Different properties of the SMP and the CAP are demonstrated with the simulation results in [17]. In [18], a voting system technique is addressed in which a single winner is chosen by voting that considers preference ranking of the voters and satisfies Condorcet, Pareto, monotonicity, resolvability, reversal symmetry and independence of clones.

A novel cooperative approach combined with ant colony and particle swarm optimization is introduced for capturing a mobile object in urban areas by deploying sensor nodes before the object leaves the region [19]. A dynamic path planning algorithm is developed in which a searcher aims to search a moving object by travelling minimum path and it is based on Sub-goal Graphs and Moving Target Search. The experimental results have been compared with G-FRA*, which is still considered the best dynamic algorithm but it shows that Sub-goal Graph is at most 29 times faster in average time and at most 186 times faster in maximum time in unit step [20]. A challenging problem is addressed in [21] where a single searcher can detect the location of an unpredictable object in a simply-connected polygonal region using a randomized strategy and the problem is called as visibility dependent pursuit-evasion problem. In [22], a moving robot aims to search a moving object in a bounded 2D convex region in minimum expected capture time and the target motion model is a crucial parameter to formulate the search strategy and the selection of the model determines the strategy. In [23], a military based target searching problem is modeled using game theory which acts as a decision-making tool for the mission planners in searching a village during an emergency operation.

The flow of the paper is organized as follows: Preference-based target searching techniques (College Admission Problem, Stable Marriage Problem, voting system) are described in brief in section II. Section III models an approach for a searcher to search single/multiple targets by visiting all the nodes of a graph following minimum traversal distance. Section IV presents an approach to determine a minimum number of searchers to search a graph following minimum traversal distance, and in section V, all proposed approaches in the paper are summarized along with discussion and future plan.

## II. Preference-Based Target Searching

Deployment of the sensors/UAVs/robots is highly preferable technologies for target searching in various defense scenarios such as terrorist searching, warfare scenarios etc. Advanced, well-equipped with different capabilities mobile sensors can be deployed in critical and high-risk zones, where they are used as a substitute for human resources. In this section, we have addressed an interesting area of target searching problem where the searchers $(S)$/both ($S$ and targets $(T)$) have the prior information about the locations of $T$/opponents' but the preference ranking of certain parameters related to $T$/opponents' are important for the search operation. The preference ranking of those parameters is known to $S$/both sides depending upon the problem type. From the defense perspective, some crucial parameters related to $S$ and $T$ in a target searching environment are as follows: resource availability (weapon strength, defense-related equipment etc.), risk factor, attacking and defending scores etc. In this problem, individual searcher/target represents a single entity or a group of entities from a side. $S$ and $T$ apply suitable strategies (preference ranking) against each other as per opponents' priority/preference ranking based on the certain parameters. To deal with such problems, we have applied three preference-based mathematical models which are CAP, SMP and voting system, respectively. Before $S$ and $T$ start the game, these models help them to determine final preference sequence/stable matching considering single/multiple parameters depending upon their individual/common preference ranking. In our proposed model, CAP and SMP are applied based on the preference ranking by $S$ and $T$ of a particular parameter and the voting system (Schulze method) is applied considering multiple parameters with equal/unequal weights. But, we have considered all the parameters with equal weights for Schulze method. We begin the discussion with CAP and its role in target searching problem.

### A. College Admission Problem (CAP)

CAP is a many-to-one matching problem whereas SMP is a one-to-one matching problem where each proposer is matched with a single acceptor, and vice-versa. Therefore, SMP is a special case of CAP. CAP is more practical than SMP, thus applicable in many real time scenarios such as recruitments and school/college admissions, etc. In our proposed work, we have considered an unequal number of mobile sensors and targets. We have briefly described CAP and its application to model the target searching problem similar to warfare scenarios and analyzed it with the help of randomly chosen examples.

1) **Model:** CAP deals with two disjoint sets: a set of students and a set of colleges but in the case of target searching problem, the disjoint sets are $S$ and $T$ in place of the students and the colleges, and vice-versa depending upon the problem type. In this problem, it is assumed that $S$ and $T$ rank each other based on the same parameter. In our proposed model, let $n$ number of $S$ be deployed to search $m$ number of $T$, where $v_{T_i}$ is the quota of $i^{th}$ target $(T_i, 1 \le i \le m)$. Each searcher $(S_j, 1 \le j \le n)$ ranks $T$ in the order of its preferences considering the same parameter; while ignoring the targets, which are unacceptable under any conditions (it is sometimes feasible but not always). It is assumed that there are no ties in preference ranking by $S$ and $T$. Therefore, we get two different preference matrices for $S$ and $T$.

Let the number of searchers and the targets to be five, $S = \{S_1, S_2, S_3, S_4, S_5\}$ and three $T = \{T_1, T_2, T_3\}$, respectively. Each searcher has preference ranking for $T$ as follows: $P(T_1) > P(T_2) > P(T_3)$ (i.e. the order signifies $P(T_1)$ is preferred most and $P(T_3)$ is least preferred by $S$). Accordingly, $T$ has the following preference ranking and quotas for $S$: $T_1$ has the preference ranking for $S$ as follows $P(S_1) > P(S_2) > P(S_3) > P(S_4) > P(S_5)$ with quota $v_{T_1} = 2$, $T_2$ has the preference ranking for $S$ as follows $P(S_5) > P(S_4) > P(S_3) > P(S_2) > P(S_1)$ with quota $v_{T_2} = 1$ and $T_3$ has the preference ranking for $S$ as follows $P(S_4) > P(S_3) > P(S_1) > P(S_2) > P(S_5)$ with quota $v_{T_3} = 1$.

Now, we analyze the outcomes of the above example based on the CAP model as follows: $T_1$ accepts the proposals of its top two choices $S_1, S_2$, respectively and $T_1$ reaches its quota of two places. Therefore, both $S_1, S_2$ and $T_1$ don't deviate from their actions. $T_2$ accepts first its top most priority based entity $S_5$. On the other hand, $S_5$ prefers $T_1$ over $T_2$ but $T_1$ has already reached its quota because $T_1$ prefers $S_1, S_2$ over $S_5$. So, $T_1$ cannot replace $S_1$ or $S_2$ by $S_5$. Therefore, both $S_5$ and $T_2$ don't deviate their actions. Based on the preference sequence of $T_3$, it accepts the proposal of $S_4$ and alternatively $S_4$ can't deviate its action because $T_1, T_2$ have already reached their limit. This matching can be said as stable matching. The expected outcomes of the above example as per CAP are as follows: matching $(T_1) = \{S_1, S_2\}$, matching $(T_2) = \{S_5\}$, and matching $(T_3) = \{S_4\}$.

In the above example, $S_3$ is left because it is not matched with any of the targets but in real-time, it is not practical as far as target searching problem is concerned. Therefore, by slightly modifying CAP model, the search scenario can be satisfied for real-time. $S$ chooses it's course of action wherein all searchers are simultaneously deployed in the search operation. Then, $T$ revises the quota for individual entities of $T$ from the original values in such a way that the total number of quota $(v_{T_1} + \cdots + v_{T_i})$ for all the entities of $T$ must be equal to the total number $(|S|)$ of $S$. Now, CAP can be directly applied based on the revised quotas of $T$. Alternatively, if number of $S$ is lesser than number of $T$, then, $S$ and $T$ follow the CAP and $S$ updates its quota from the original values depending upon the number of $T$ as per the discussion mentioned above.

2) **Student (Searcher) Optimal Deferred Acceptance (SODA):** The SODA algorithm is same as the Gale-Shapley algorithm. Each searcher $S_j$ proposes to the most preferable target $T_i$, if $T_i$ has not reached its quota. It matches $S_j$ only if it is preferable to being unmatched. Alternatively, if $T_i$ has reached its quota and $T_i$ prefers the new $S_j$ to an already matched $S_k$, then $T_i$ matches the new $S_j$ and releases the old $S_k$. The matching obtained from the SODA algorithm is always student (searcher)-optimal.
3) **College (Target) Optimal Deferred Acceptance:** CODA algorithm is same as Gale-Shapley algorithm. Each target $T_i$ proposes to the most preferable searcher $S_j$. It matches $T_i$ only if it is preferable to being unmatched. If $S_j$ prefers the new $T_i$ to an already matched target $T_l$ and the new $T_i$ doesn't reach its quota, then $S_j$ matches the new $T_i$ and releases the old $T_l$. The matching obtained from CODA algorithm is always college (target)-optimal.

### B. Stable Marriage Problem (SMP)

The fundamental differences between SMP and CAP model are mapping between the entities and quota constraint. In SMP, a number of men (here, searchers $S$) and women (here, targets $T$) are equal; mapping between $S$ and $T$ is one-to-one and reservation quota for $S$ and $T$ is one. In CAP, we have already observed that the number of students ($S$) and colleges ($T$) are unequal; mapping between $S$ and $T$ is many-to-one and reservation quota for $T$ is determined by the mission planner of the target/searcher side depending upon the problem type. Therefore, SMP can be applied to search a single target ($T_i$) by a single searcher ($S_j$) considering one-to-one mapping rule and preference ranking by both of them. SMP is yet not an ideal approach to model real-time scenarios because in practical one-to-one mapping between $S$ and $T$ rarely happens in target searching scenarios. SMP and its application to model target searching problem similar to warfare scenarios are discussed below.

**Model:** SMP is also called Gale-Shapley algorithm. In this model, the participants "men" and "women" are replaced by $S$ and $T$. First, each unengaged $S_u$ proposes to the most preferable $T_w$ for itself and each target ($T$) responds "can be selected" if it gets better choice than previous one otherwise directly replies "no" to all other $(S - S_a)$ where $S_a$ is its old pair with top most preference. $T_w$ is then temporarily engaged to the most preferable $S_a$ till that time and $S_a$ is also temporarily engaged to $T_w$. In this way in every round, each unengaged $S_u$ first proposes to its most preferable $T_w$ to whom $S_u$ has not proposed till that time without considering whether $T_w$ is already engaged and then each target replies "can be selected" if $T_w$ is presently not engaged or if $T_w$ gets the proposal from $S_u$ of higher preference over its present mate. In this case, $T_w$ rejects its present mate who becomes unengaged. This process continues until everyone in the set is engaged.

In the example, we have considered three searchers $S = \{S_1, S_2, S_3\}$ and three targets $T = \{T_1, T_2, T_3\}$ for computing final pairs applying Gale-Shapley algorithm. In Table 1., the first number in each cell represents the ranking of $T_i (1 \leq i \leq 3)$ by $S_j (1 \leq j \leq 3)$; the second number is the ranking of $S_j$ by $T_i$.

Table 1. Preference ranking by all the searchers and the targets [15]

| | $T_1$ | $T_2$ | $T_3$ |
|---|---|---|---|
| $S_1$ | 1, 3 | 2, 2 | 3, 1 |
| $S_2$ | 3, 1 | 1, 3 | 2, 2 |
| $S_3$ | 2, 2 | 3, 1 | 1, 3 |

$S$ and $T$ formulate their strategies in searching/hiding operation as per SMP model. Therefore, both of them compute the stable matching preference list before they involve themselves in the operation. Final matching between the entities of $S$ and $T$ is determined as follows: In round 1, $S_1$ prefers $T_1$, since $T_1$ is previously unmatched, $S_1$ becomes engaged to $T_1$. Similarly, $S_2$ prefers $T_2$ and since $T_2$ is previously unmatched, gradually $S_2$ becomes engaged to $T_2$. Accordingly, $S_3$ is also engaged to $T_3$. As $S$ approaches $T$, $S$ is benefitted whereas $T$ is not. On the other hand, if $T$ approaches $S$, then, $T_1$ is engaged to $S_2$, $T_2$ is engaged to $S_3$ and $T_3$ is engaged to $S_1$. In this case, $T$ is benefitted whereas $S$ is not benefitted. Another dimension of the SMP is where $S$ and $T$ choose their second most preferred match, which means $S_3$ is engaged to $T_1$, $S_1$ is engaged to $T_2$ and $S_2$ is engaged to $T_3$ [15]. In many examples, it is not compulsory for SMP that $S$ and $T$ get their top preferences to yield a stable matching which is computed as per the method discussed in this section.

### C. Voting system: Schulze method

In real-time, sometimes it happens that $S$ is aware of various information about $T$ such as distance from opponents, resource availability, risk factor, attacking and defending score etc, but $T$ doesn't have any information about $S$. Parameters of $T$ can have equal/unequal weights (number of voters in a specific preference sequence pattern as per Schulze method) depending upon the problem type. $S$ determines the final preference sequence for $T$ by applying Schulze method considering certain parameters and weight of each parameter of $T$ which are important for the search operation.

**Schulze method:**

1. Let $p(T_a, T_b)$ denotes the number of voters (assumed to be parameters in our problem) which prefers $T_a$ to $T_b$
2. Determine path from $T_a$ to $T_b$ of weight w which is a sequence of vertices $V_1, V_2, \cdots, V_n$ following these rules mentioned below

   a. $V(1) = T_a$ and $V(n) = T_b$
   b. $p(V(i), V(i+1)) > p(V(i+1), V(i))$ for $i = 1, 2, \cdots, n-1$
   c. $p(V(i), V(i+1)) \geq w$, for $i = 1, 2, \cdots, n-1$

3. Say, $w(T_a, T_b)$ is the weight of the most preferable path from $T_a$ to $T_b$ which has the maximum value, i.e. the value of the path that exists from $T_a$ to $T_b$ amounts to that weight. If no path exists from $T_a$ to $T_b$, then $w(T_a, T_b) = 0$
4. $T_a$ is more preferable than $T_b$ iff $w(T_a, T_b) > w(T_b, T_a)$. $T_a$ is a potential winner iff $w(T_a, T_b) \geq w(T_b, T_a)$ for every $T_b$.
5. It can be shown that $w(T_a, T_b) > w(T_b, T_a)$ and $w(T_b, T_c) > w(T_c, T_b)$, then $w(T_a, T_c) > w(T_c, T_a)$

**Random-scenario:** Let us consider five targets

$T_1, T_2, T_3, T_4$ and $T_5$ respectively. Say, $S$ sets the preference sequence $T_1 > T_2 > T_3 > T_4 > T_5$ based on distance from opponents, $T_3 > T_1 > T_2 > T_5 > T_4$ based on resource availability, $T_2 > T_1 > T_4 > T_3 > T_5$ based on risk factor, $T_1 > T_3 > T_4 > T_5 > T_2$ based on attacking score and $T_1 > T_5 > T_4 > T_2 > T_3$ based on defending score etc. In this example, the weights of all the parameters are assumed as one but the algorithm works for any weights of different parameters of $T$. In real-time, unequal weights of different parameters are quite possible and the weights can be determined by the decision maker for $S$ based on the prior information about $T$.

Table 2. Preference ranking by the searchers for different targets based on certain parameters

| Parameters | Weight | Preference sequence |
|---|---|---|
| Distance from opponents | 1 | $T_1, T_2, T_3, T_4, T_5$ |
| Resource availability | 1 | $T_3, T_1, T_2, T_5, T_4$ |
| Risk factor | 1 | $T_2, T_1, T_4, T_3, T_5$ |
| Attacking score | 1 | $T_1, T_3, T_4, T_5, T$ |
| Defending score | 1 | $T_1, T_5, T_4, T_2, T_3$ |

First, pairwise preferences are computed from the preference sequences of Table 2. Initially, when $T_1$ and $T_2$ are pairwise compared, the total weight 1+1+1+1= 4 is computed where $T_1$ is more preferable than $T_2$ and the total weight is 1 where $T_2$ is more preferable than $T_1$. Therefore, we can evaluate $p(T_1, T_2) = 4$ and $p(T_2, T_1) = 1$. The pairwise preferences are computed for all the pairs in Table 3.

Table 3. Pairwise preference weight for the targets as per Schulze method

| | $p(\#, T_1)$ | $p(\#, T_2)$ | $p(\#, T_3)$ | $p(\#, T_4)$ | $p(\#, T_5)$ |
|---|---|---|---|---|---|
| $p(T_1, \#)$ | - | 4 | 4 | 5 | 5 |
| $p(T_2, \#)$ | 1 | - | 3 | 3 | 3 |
| $p(T_3, \#)$ | 1 | 2 | - | 3 | 4 |
| $p(T_4, \#)$ | 0 | 2 | 2 | - | 3 |
| $p(T_5, \#)$ | 0 | 2 | 1 | 2 | - |

The values of the cells are marked red if $p(T_a, T_b) > p(T_b, T_a)$ else they are indicated with green color. We cannot determine the final preference sequence in this step.

In next step, the highest weighted paths are determined, therefore, a directed graph is drawn based on the weights of the pairs. In Fig.1., arrow from vertex $T_a$ to $T_b$ is represented as $p(T_a, T_b)$ where $p(T_a, T_b) > p(T_b, T_a)$. In the above table, the cells $p(T_a, T_b)$ contain values which determine the direction of the arrow but values in green are not represented by the arrow in opposite direction in the graph.

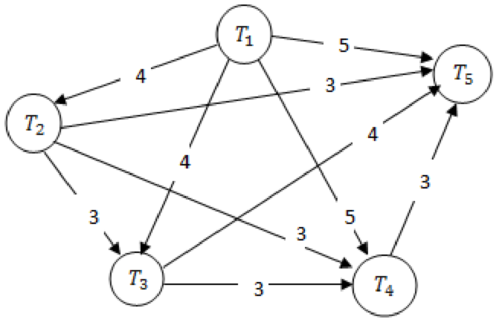


Fig.1. Pairwise highest weighted paths in preference graph

The highest weight from one vertex to another vertex in the graph is computed to determine the highest weighted paths. Sometimes, the highest weighted paths are achieved through intermediate vertices. For an example, computed value of the highest weighted path strength for $p(T_1, T_4)$ is 5: the highest weight path from $T_1$ to $T_4$ is the direct path $(T_1, T_4)$ which is of weight 5. In this example, highest weight paths through intermediate vertices is not feasible (For example, the highest weight is assigned for the path from $T_a$ to $T_b$ and an intermediate vertex from $T_a$ to $T_b$ be $T_c$ and if $(T_a, T_b) < \min(p(T_a, T_c), p(T_c, T_b))$, then, the highest weighted path is computed using intermediate vertices). Therefore, the values in the above table remain unchanged.

Now, we can determine final preference sequences using Schulze method. As an example, we compare $T_1$ and $T_2$ and observe $(p(T_1, T_2) = 4) > (p(T_2, T_1) = 1)$. Therefore, it can be inferred that $T_1$ is better than $T_2$. We compare for all the vertex pairs following this procedure.

Finally, we determine the relations for all the pairs

$$T_1 > T_2, T_1 > T_3, T_1 > T_4, T_1 > T_5, T_2 > T_3,$$

$$T_2 > T_4, T_2 > T_5, T_3 > T_4, T_3 > T_5, T_4 > T_5,$$

Based on these relations, it can be inferred that-

$$T_1 > T_2 > T_3 > T_4 > T_5$$

Based on this final preference sequence considering all the parameters of $T$, $S$ formulates strategies to accomplish the search operation. The Schulze method is effective when either multiple parameters of $T$ are required to choose the decision or final preference sequence of a single parameter is determined considering the preference sequences of each entity of $S$ based on prior information or past experiences.

## III. Minimizing Traversal Distance for Searching Targets

In this section, we develop and demonstrate a novel approach that can be applied in terrorist searching/warfare scenarios, where an advanced, well-equipped mobile sensor is deployed as a searcher $(S_a)$. It is assumed that $S_a$ has the map of the entire search space but doesn't know the exact locations of the immobile hidden targets $(T)$ in the search space. The search space is represented as a graph $(G)$ where the set of vertices $(V)$ are all possible hiding locations of $T$ including the starting position of $S$ and the edges $(E)$ are all possible routes among all possible hiding locations and the starting position of $S$, respectively. All the vertices have equal distance from their neighbor vertices considered in our proposed model. The aim of this proposed work is to search $T$ by visiting all the vertices $(V)$ with minimum path length. To achieve this goal, we introduce a novel graph search approach.

*A. Method*

1. Find the adjacent matrix $(A_{ij}, 1 \leq (i,j) \leq |V|)$ based on $G$ and take initially two disjoint vertex sets: uniquely visited vertex set $(V_v)$ and unvisited vertex set $(V_u)$, where $V = V_v \cup V_u$ and a variable path length $(L)$ which is initialized as zero. Initially, $V_v$ contains only the starting vertex $(v_s)$ and $V_u$ contains $(V - v_s)$. Both the sets $V_v$ and $V_u$ are separately updated iteration-wise for different path distributions and their values are updated from the values inherited from the previous iteration.
2. Apply any 'all pair shortest path algorithm' to compute the shortest distances from any vertex $(v_i, 1 \leq i \leq |V|)$ to all other vertices $(V - v_i)$ and represent a minimum distance matrix $(D_{ij})$.
3. $S$ starts its journey from $v_s$. So, indices of 1's in the row of $v_s$ in $A_{ij}$ are found and the column indices containing 1's in the row of $v_s$ simply represents adjacent vertices of $v_s$. Those adjacent vertices are separately appended with $v_s$ to form separate paths with $L$=1. Then, separate $V_v$ and $V_u$ are assigned for separate paths. This step has three possible cases, where $V_v$, $V_u$ and $L$ are separately updated for new vertices based on the values inherited from the previous iteration as per conditions 3.a., 3.b. and 3.c. discussed below:

   a. For the rows containing single 1 (single path) and $V_u \neq \phi$, $S$ backtracks to its predecessor vertices to reach nearest (determined from step 2) unvisited vertex $(v_k \in V_u)$ and $V_v$, $V_u$ and $L$ are updated in each backtrack. Simultaneously, backtracked vertices are appended with their parent vertices to generate the updated paths.

   b. For the rows containing multiple 1's (multiple paths) and $V_u \neq \phi$, the search paths are extended in all feasible directions from the current vertex to the unvisited adjacent vertices. Separate $V_v$ and $V_u$ are assigned for new unvisited vertices based on the values inherited from the previous iteration and simultaneously, $L$ is updated for all the new unvisited vertices and new vertices are appended with their parent vertices to generate the updated paths.

   c. For the rows containing single/multiple 1's where $V_u = \phi$ and $V_v = V$, traversing process for $S$ terminates and $L$ is finally updated. The column indices containing 1's in the row of a vertex in $A_{ij}$ are mapped to the same indices of the rows to find 1's in the columns of those rows to further find the corresponding neighbors of their neighbor vertices and the conditions 3.a., 3.b. and 3.c. are repeatedly tested in every iteration.

4. The earliest iteration in which all the vertices $(V)$ of $G$ are visited, the search process terminates and rest of the possible search paths are ignored. Finally, we determine the sequence of vertices $(v_s, v_i, \dots, v_l)$ that form the optimal search path for $S$.

Note: There can be multiple optimal paths, any one among all the optimal paths can be chosen.

**Random-scenario:**

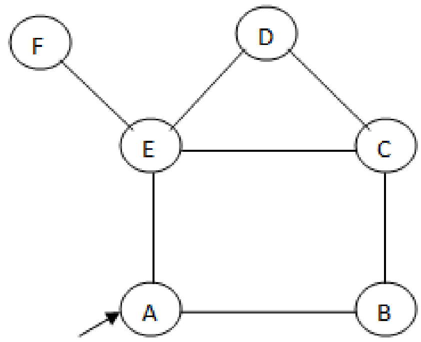


Fig.2. Example: The graph to be searched by $S$ from vertex A following minimum traversal distance

We have randomly chosen a graph illustrated in Fig. 2. for analyzing the proposed approach for searching $T$ by travelling the minimum distance. The given graph in the example contains six vertices A, B, C, D, E and F connected with certain edges. The starting vertex of $S$ is A represented by an arrow in Fig. 2. Adjacent matrix for the above graph is represented below.

$$
\begin{array}{c|cccccc}
 & A & B & C & D & E & F \\
\hline
A & 0 & 1 & 0 & 0 & 1 & 0 \\
B & 1 & 0 & 1 & 0 & 0 & 0 \\
C & 0 & 1 & 0 & 1 & 1 & 0 \\
D & 0 & 0 & 1 & 0 & 1 & 0 \\
E & 1 & 0 & 1 & 1 & 0 & 1 \\
F & 0 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

Table 4. Iteration-wise parameter updation using proposed graph searching method for $S$ as per Fig. 2

| Iteration | $V_v$ | Paths | $V_u$ | $L$ |
|---|---|---|---|---|
| 0 | {A} | A | {B,C,D,E,F} | 0 |
| 1 | {A,B} | AB | {C,D,E,F} | 1 |
| 1 | {A,E} | AE | {B,C,D,F} | 1 |
| 2 | {A,B,C} | ABC | {D,E,F} | 2 |
| 2 | {A,E,C} | AEC | {B,D,F} | 2 |
| 2 | {A,E,D} | AED | {B,C,F} | 2 |
| 2 | {A,E,F} | AEF | {B,C,D} | 2 |
| 3 | {A,B,C,D} | ABCD | {E,F} | 3 |
| 3 | {A,B,C,E} | ABCE | {D,F} | 3 |
| 3 | {A,E,C,B} | AECB | {D,F} | 3 |
| 3 | {A,E,C,D} | AECD | {B,F} | 3 |
| 3 | {A,E,D,C} | AEDC | {B,F} | 3 |
| 3 | {A,E,F,E} | AEFE | {B,C,D} | 3 |
| 4 | {A,B,C,D,E} | ABCDE | {F} | 4 |
| 4 | {A,B,C,E,D} | ABCED | {F} | 4 |
| 4 | {A,B,C,E,F} | ABCEF | {D} | 4 |
| 4 | {A,E,C,B,C} | AECBC | {D,F} | 4 |
| 4 | {A,E,C,D,C} | AECDC | {B,F} | 4 |
| 4 | {A,E,C,D,E} | AECDE | {B,F} | 4 |
| 4 | {A,E,D,C,B} | AEDCB | {F} | 4 |
| 4 | {A,E,F,E,C} | AEFEC | {B,D} | 4 |
| 4 | {A,E,F,E,D} | AEFED | {B,C} | 4 |
| 5 | {A,B,C,D,E,F}* | ABCDEF* | {ϕ} | 5 |

Once $S$ visits all the vertices of $G$, then, the search operation terminates. In Table 4., the last row represents the final outcome. It shows that ABCDEF is the optimal path to accomplish the search operation by traversing the minimum distance and the length of the path is five units.

The proposed approach for the given problem works for all kind of graphs but suffers from the computational overhead. The computation of graph searching becomes costlier for a large size graph in which the number of vertices and edges are large but in a parallel computing environment, the computational overhead of the search operation can be reduced. To the best of our knowledge, there is no convincing solution for this problem till now.

## IV. Determining Minimum Number of Searchers to Search a Graph in Minimum Traversal Distance

In this section, we develop and demonstrate another novel approach that can be applied in terrorist searching/warfare scenarios in which a set of mobile sensors are deployed as searchers $(S)$. It is assumed that $S$ has the map of the entire search space but they don't know the exact locations of the immobile targets $(T)$ in the search space. The search space is represented as a graph $(G)$ where the set of vertices $(V)$ and edges $(E)$ are probable hiding locations for $T$ and all possible routes among the hiding locations and the starting position of $S$ respectively. It is assumed in our proposed model that all the vertices have equal distance from their neighbor vertices. The aim of this proposed work is to search $T$ in the graph by traversing the minimum distance from $v_s$ (starting location of all the searchers are same) using minimum number of $S$. To achieve the goal, we introduce a novel graph searching method.

### A. Method

1. Find adjacent matrix $(A_{ij}, 1 \leq i,j \leq |V|)$ based on $G$ and take two disjoint sets: visited vertex set $(V_v)$, unvisited vertex set $(V_u)$ where $V = V_v \cup V_u$ and a variable searcher count (δ) respectively. Initially, $V_v$ contains only the starting vertex $(v_s)$ and $V_u$ contains $(V - v_s)$ and δ=0. $V_v$, $V_u$ and δ are updated iteration-wise.
2. Apply any 'all pair shortest path algorithm' to determine the distance from any vertex $(v_i, 1 \leq i \leq |V|)$ to all other vertices $(V - v_i)$ and represent a minimum distance matrix $(D_{ij})$. Find the vertex/vertices $(v_{max}, |max| \geq 1, max \subseteq V)$ of the maximum path length $(M)$ from $v_s$ through a maximum number of possible distinct intermediate vertices (minimizing common intermediate vertices in different paths if multiple $v_{max}$ exist). There are two possible cases:

   a. If there are multiple paths of length $M$ from $v_s$ to multiple vertices $(v_{max}, |max| > 1)$, then, deploy multiple searchers $(S_r, 1 < r \leq max)$ for different paths of length $M$ and update $V_v$, $V_u$ and δ.

   b. If there are single/multiple paths of length $M$ from $v_s$ to a specific vertex $(v_{max}, |max| = 1)$, then, choose any one path among all the paths of length $M$ and update $V_v$, $V_u$ and δ.

3. Sort remaining vertices of $V_u$ in increasing order of path length from $v_s$ after determining the paths to reach $v_{max}$.
4. Perform permutation/combination operation on the remaining unvisited vertices in increasing order of path length from $v_s$ in such a way that the largest possible subsets of the unvisited vertices including $v_s$ forms single/multiple paths whose length is lesser than or equal to the path length of $M$ and

simultaneously update $V_v$, $V_u$ and δ.

5. If unvisited vertices are still left ($V_u \neq \phi$), then, repeat step 3 and 4 and continue the iteration until $V_v$ becomes full ($|V_v| = |V|$) and $V_u$ becomes empty ($V_u = \phi$) and finally update δ.

In this way, we can determine the minimum number of searchers which are deployed to search the entire graph by traversing the minimum distance.

**Random-scenario:**

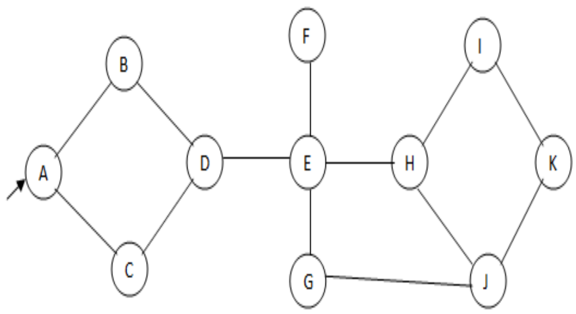


Fig.3. Example: The graph to be searched by minimum number of $S$ in minimum traversal distance

We have randomly chosen a graph illustrated in Fig. 3 for analyzing the proposed approach for searching $T$ by the time taken to travel $v_{max}$ from $v_s$ using a minimum number of $S$. The given graph in the example contains 11 vertices A, B, C, D, E, F, G, H, I, J and K, connected with certain edges. The starting vertex for $S$ is A represented by an arrow in Fig. 3.

In this section, we determine the minimum number of $S$ required to accomplish the search operation by the distance traversed to reach $v_{max}$ from $v_s$. Due to the large size of the graph, $D_{ij}$ and $A_{ij}$ are not illustrated in the form of matrices. Initially, $V_u$ contains all the vertices of the graph except source vertex (A), $V_v$ contains only A and δ=0. We directly move to step 2 as per the proposed method and show the outcomes obtained from this step in Table 5.

Table 5. Possible paths, visited vertices and unvisited vertices for $S_1$ determined in the 1st iteration as per Fig. 3

| $V_v$ | $V_u$ | Paths | Value of δ in initial stage |
|---|---|---|---|
| {A,B,D,E,H,I,K} | {C,F,G,J} | ABDEHIK | 1 |
| {A,B,D,E,H,J,K} | {C,F,G,I} | ABDEHJK | |
| {A,B,D,E,G,J,K} | {C,F,H,I} | ABDEGJK | |
| {A,C,D,E,H,I,K} | {B,F,G,J} | ACDEHIK | |
| {A,C,D,E,H,J,K} | {B,F,G,I} | ACDEHJK | |
| {A,C,D,E,G,J,K} | {B,F,H,I} | ACDEGJK | |

The above example satisfies step 2.b., therefore, $S_a$ can randomly choose one of the combinations among six possible $V_v$ and $S_a$ finds a path. Say, $S_a$ has chosen the path ABDEHIK, therefore, C, F, G and J vertices are left in $V_u$. So, one searcher is required to be deployed to traverse the path ABDEHIK. Next, we move to step 4 and thereafter step 5. The path ABDEHIK has a path length of six units. So, we have to find the path of at most $M = 6$ using the set of vertices of $V_u$ (C, F, G, J). So, we can find a path from any subset of $V_u$ and the possible paths are ACDEFEG, ACDEGEF and ACDEGJ, respectively. We can randomly choose one of the three paths. Let us choose the path ACDEFEG. So, another searcher is required to be deployed to traverse the path ACDEFEG. Now, $V_u$ contains a single element J, then, we can find one of the shortest paths from A to J. There are four possible paths ABDEHJ/ ABDEGJ/ ACDEHJ/ ACDEGJ, respectively. So, another searcher is required to be deployed to search one of the four paths mentioned above. Therefore, we need to deploy minimum three searchers (δ=3) to search the graph by traversal distance six units. This approach works for all kind of networks but suffers from computational overhead like the previous approach in section III. The computation of graph searching becomes costlier for a large size graph but in a parallel computing environment, the computational overhead of the search operation can be reduced.

Section III and section IV do not illustrate the overall performance of the proposed methods based on large-scale simulations because the results depend upon the structure of the graph (number of vertices and edges, connectivity among the vertices and the initial positions of the searchers and the targets). We haven’t shown the results for different graphs due to the space constraints.

## V. CONCLUSIONS AND FUTURE WORK

This paper highlights different types of defense-related target searching problems and the proposed approaches are analyzed with the results obtained from randomly chosen examples. It is already mentioned in the paper that searchers are basically mobile sensors and the targets can be different types and capabilities of objects. We have aimed to facilitate the strategy selection for the searchers to search the targets in different defense scenarios. Three preference-based target searching problems have been addressed; the preference-based target searching problems have been solved using Stable Marriage Problem, College Admission Problem, and voting system, respectively. On the other hand, two novel target searching approaches have been developed and demonstrated to minimize the traversal distance and the number of searchers is minimized to accomplish the search operation. The results obtained from the given examples justify the significance of our proposed models. In future, we plan to model more complex real-time target searching problems related to defense considering more constraints and involving more intelligent targets in a complex environment, where game theoretic and other mathematical approaches can be applied to model such problems.

## ACKNOWLEDGEMENT

We are thankful to Ms. Monica Ravishankar, research scholar of Dept. of Computer Science And Engineering, DIAT for her valuable suggestions.


## REFERENCES


[1] S. Shen, G. Yue, and Q. Cao, "A survey of game theory in wireless sensor networks security," J. Networks, 6 (3), pp. 521-532, 2011.

[2] H. Y. Shi, W. L. Wang, N.M. Kwok, and S. Y. Chen, "Game theory for wireless sensor networks: A survey," Sensors 12 (7), pp. 9055-9097, 2012.

[3] R. Machado, and S. Tekinay, "A survey of game-theoretic approaches in wireless sensor networks," Comput. Networks (Elsevier), 52, pp. 3047-3061, 2008.

[4] J. S. Cox, and E. H. Durfee, "An efficient algorithm for multiagent plan coordination," 4th International joint conference on Autonomous agents and multiagent systems, pp. 828-835, 2005.

[5] J. Berger, N. Lo, and M. Noel, "A new multi-target, multi-agent search-and-rescue path planning approach," Int. J. Comput., Elec., Auto., Contr. and Info. Engineering 8(6), pp. 935-944, 2014.

[6] Y. Meng, "Multi-robot searching using game-theory based approach," Int. J. Adv. Robotic Systems 5(4), pp. 341-350, 2008.

[7] T. Das, and S. Roy, "Game theory inspired mobile object trapping system in mobile wireless sensor network," IEEE International Conference Electronic Systems, Signal Processing and Computing Technologies (ICESC), pp. 245-250, 2014.

[8] T. Hadzic, K. N. Brown, and C. J. Sreenan, "Real-time pedestrian evacuation planning during emergency," 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 597-604, 2011.

[9] Prinzie, H. Oppewal, "A sequence alignment distance measure for preference rank aggregation," ANZMAC Conference: Marketing Research and Research Methodologies (quantitative), pp. 100-107, 2005.

[10] Antoniades, H. J. Kim, and S. Sastry, "Pursuit-evasion strategies for teams of multiple agents with incomplete information," 42nd IEEE International Conference Decision and Control (ICDC), pp. 756-761, 2003.

[11] T. H. Chung, and J. W. Burdick, "Multi-agent probabilistic search in a sequential decision-theoretic framework," IEEE International Conference of Robotics and Automation (ICRA), pp. 146-151, 2008.

[12] S. Waharte, A. Symington, and N. Trigoni, "Probabilistic search with agile UAVs," IEEE International Conference of Robotics and Automation (ICRA), pp. 2840-2845, 2010.

[13] Khan, E. Yanmaz, and B. Rinner, "Information merging in multi-UAV cooperative search," IEEE International Conference of Robotics and Automation (ICRA), pp. 3122-3129, 2014.

[14] Strode, "Optimising multistatic sensor locations using path planning and game theory," Computational Intelligence for Security and Defense Applications (CISDA), IEEE Symposium on, pp. 9-16, 2011.

[15] Gale, and L. S. Shapley, "College admissions and the stability of marriage," The American Math. Monthly 69(1), pp. 9-15, 1962.

[16] E. Roth, "The college admissions problem is not equivalent to the marriage problem," J. Econ. Theory, vol. 36, pp. 277-288, 1985.

[17] P. Teo, J. Sethuraman, and W. P. Tan, "Gale-Shapley stable marriage problem revisited: strategic issues and applications," Management Sci. 47 (9), pp. 1252-1267, 2001.

[18] M. Schulze, "A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method," Soc. Choice Welf. 36, pp. 267-303, 2011.

[19] S. M. Nieberg, E. Kropat, S. Pickl, and A. Bordetsky, "Intercepting a target with sensor swarms," 46th IEEE Hawaii International Conference on System Sciences (HICSS), pp. 1222-1230, 2013.

[20] Nussbaum, and A. Y ̈or ̈ukc ̈u, "Moving target search with subgoal graphs," Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, pp. 179-187, 2015.

[21] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion in a polygonal environment," IEEE Trans. on Robototics, 21 (5), pp. 875-884, 2005.

[22] Renzaglia, N. Noori, and V. Isler, "The role of target modeling in designing search strategies," IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 4260-4265, 2014.

[23] P. Maxwell, A. A. Maciejewski, H. J. Siegel, J. Potter, and J. Smith, "A mathematical model of robust military village searches for decision making purposes," International Conference on Information and Knowledge Engineering (IKE 09), pp. 311-316, 2009.


## Authors' Profiles

**Tanmoy Hazra** received his B.Tech degree in Computer Science and Engineering (2009) from West Bengal University of Technology, M.Tech in Computer Application (2012) from Indian School of Mines, Dhanbad and pursuing Ph.D in Defence Institute of Advanced Technology, Pune; a Defence Research and Development Organization (DRDO) Establishment under the Ministry of Defence, India. He published a research oriented book. His research area is Game Theory and Sensor Network.


**C. R. S. Kumar** received his M.Tech degree from IIT, Madras and Ph.D from University of Melbourne. He is currently HOD in the Department of Computer Science and Engineering, Defence Institute of Advanced Technology (DIAT), Pune; a Defence Research and Development Organization (DRDO) Establishment under the Ministry of Defence, India. His areas of interest are Wireless Communication and Networking, Network Security, Jamming/Anti jamming, Game Theory, Cognitive radio. He published many reputed journal and conference papers. He is Fellow of Institution of Engineers (India), IETE, senior member of CSI and IEEE, member of Indian Society for Technical Education.

 

**Manisha Nene** received M.E. degree from University of Pune, Ph.D. degree in Computer Science and Engineering from the Defence Institute of Advanced Technology (DIAT), Pune; a Defence Research and Development Organization (DRDO) Establishment under the Ministry of Defence, India. She is currently a Faculty Member with the Department of Computer Science and Engineering, DIAT, Pune. Her areas of interest are wireless sensor networks, cyber physical systems, cognitive radios, analysis of algorithms, high performance computing, modeling, and simulations. She published many reputed journal and conference papers. She is professional member of ACM, IEEE, CSI.

**How to cite this paper:** Tanmoy Hazra, CRS Kumar, Manisha J. Nene,"Strategies for Searching Targets Using Mobile Sensors in Defense Scenarios", International Journal of Information Technology and Computer Science(IJITCS), Vol.9, No.5, pp.61-70, 2017. DOI: 10.5815/ijitcs.2017.05.08