**Multi target tracking using swarm of quadcopters**

**DEPT. OF MECHANICAL ENGINEERING**

**UNIVERSITY AT BUFFAL0**

Ajaipal Singh Chhina

Dept. of Mechanical and Aerospace Engineering

University at Buffalo

Email: [ajaipal@buffalo.edu](mailto:ajaipal@buffalo.edu)

Srinivas Reddy Adulla

Dept. of Mechanical and Aerospace Engineering

University at Buffalo

Email : [sadulla@buffalo.edu](mailto:sadulla@buffalo.edu)

Sri Sadhan Jujjavarapu

Dept. of Mechanical and Aerospace Engineering

University at Buffalo

Email : [srisadha@buffalo.edu](mailto:srisadha@buffalo.edu)

**ABSTRACT**

The ever-increasing applications of Unmanned Aerial Vehicles (UAV) have shown the great capabilities oftechnology. However, for many cases where one UAV is a powerful tool, an autonomous swarm all working cooperatively to the same goal presents amazing potential. Environment that are dangerous for humans, are either too small or too large for safe or reasonable exploration, and even those tasks that are simply boring or unpleasant are excellent areas for UAV swarm applications. In order to work cooperatively, the swarm must allocate tasks and have adequate path planning capability. Multi-target tracking is a widely used application nowadays and is used in many field like surveillance, military, oceanography etc. There are many algorithms which can solve the problem of multi tracking like Modified PSO, Bee Swarm optimization, firefly optimization etc. In this paper we are using three algorithms to solve the problem of multi-tracking system namely 1) Group PSO 2) Firefly Algorithm 3) Modified Bee swarm algorithm.

1. **INTRODUCTION**

The problem of tracking has been widely used and it has emerged out with many defense and civilian applications for example search and rescue in any disaster situation, intelligence surveillance and reconnaissance commonly known as ISR, oceanography, space applications, computer vision, biomedical research, victim identification and autonomous vehicles and robots. The research of multi target tracking is really old and spans for about 50 years, firstly with the stage of using only one robots and with increase in technology in coming decades using of swarm of robots to track the target so as to get effective results in short span of time. During the last two decades, with the advancement of the heuristic techniques and computing technology, the research area in tracking have picked up significant boom.

It’s worth noting that Global optimization strategies are applied for theoretical convergence properties and are quite straightforward to implement. In practical applications the Global optimizer can only obtain suboptimal solutions. In many cases this is acceptable but in some cases there are many suboptimal solutions and we need all of them as in the case of tracking.

The problem we are working on multiple target tracking (MTT’s) by using multiple robot systems(MRS’s). Among many swarm application, target tracking is considered the most challenging one because of many constraints that arises in the problem. In the current given problem, a modified PSO is used to track all the targets with each particle of the swarm intelligence algorithm is represented by a robot. Each region of the search space is divided into multiple parts, say n, with each part assigned to equal number of robots, say m. Hence the total number of robots used in our problem is ‘n x m’. In case of Dynamic PSO, Group PSO and Modified Bee Swarm Optimization every robot shoots from the same point i.e. lower left corner of the work space while in case of firefly the robots are deployed from each corner of the workspace. The multiple robot system interacts with each other using the technique of PSO where each robot velocity is updated according to the local best and global best of the region it’s assigned to. The main task of our problem is to use techniques or algorithms which are able to find all the local optima’s or targets in the given workspace. Different algorithms are used to find all the local optima’s namely Dynamic PSO, Group PSO, Firefly algorithm, Modified Bee swarm algorithm. A brief history of PSO is discussed in the following paragraph to which all the other algorithms are derived from and further each algorithm including PSO is discussed in detail.

The basic PSO was first proposed in 1995 by Eberhart and Kennedy bringing a revolutionary change in the field of heuristics. The proposed algorithm was different from other evolutionary algorithms in such a way that it was inspired from the social behavior of the organisms and how they interact with each other. In this paper we will discuss how the basic PSO works and how it is not suitable for tracking multiple targets as PSO was initially designed for finding the global optima of the given function. A different approach by Shoutao Li, Lina Li, Gordon Lee, Hao Zhang in their article “A Hybrid Search Algorithm for Swarm Robots Searching in an Unknown Environment” will be studied in which a hybrid algorithm of random search and PSO is applied in order to find the multiple targets. The author named the new algorithm as Dynamic PSO or DPSO and for the rest of the paper we will use that notation as the name of the algorithm. The other algorithms namely Group PSO, Firefly, Modified Bee swarm optimization are explained, implemented and their results are discussed in the end. In all the programs initially the particles start with random search and then move onto their respective algorithms.

**The paper is organized as follows:** 1) The background and basic parameters of PSO is discussed 2) The discussion on DPSO and how it’s different from PSO and can be used for finding multiple targets 3) The working of the proposed modified Bee Swarm optimization and how it a can be able to find multiple optima’s(targets) 4)The Firefly algorithm is used to solve the same problem 5) Lastly, The Group PSO is implemented and the results obtained of each algorithm is shown.

1. **LITERATURE REVIEW**
   1. **Particle Swarm Optimization**

PSO algorithm is inspired from the social behavior of the group of organisms whom interact, move with respect to a leader in the group, and try to do their job done with minimum energy and least amount of time. PSO holds some of basic similarity to GA (Genetic Algorithm) like updating the solution with each generation starting the problem with a population of random solutions, however, unlike GA, it’s parameter and working is way different and doesn’t involve any crossover or mutation.

While searching for food the animals either go scattered or go in swarm to locate the food. PSO is inspired from the animals who go in groups for the search of food such as fish flock, swarm of birds. Each individual of the group is known as particle and the collective population is known as swarm. The swarming behavior of the particles could be implemented in algorithm to find optima’s (food resources). PSO can be used to find the model of N-dimensional, nonlinear, convex and multi modal functions. While the swarm is searching of food there is always a chance where an individual of the group is able to sense or percept the food more efficiently than the rest of the group. Each individual is compared to the swarm best and individual best solution and the position of the particle is updated accordingly. With each iteration the best solution is updated and rest of the swarm get their position updated relative to the particle with best solution. Due to its simplicity and better convergence PSO has been widely used in many applications.

**Algorithm**

In basic PSO, we are considering swarm of ‘N’ Particles which are randomly distributed over the work space and the position of each particle can be in D-dimensional space. The particle changes their position w.r.t three principles

1. To maintain its inertia
2. To change its position with respect to most optimum solution
3. To change its position with respect to swarms most optimum position.

Where the most optimum solution is the best solution obtained by a particle into consideration till that iteration and the swarms most optimum position is the global best function value found by all the particles till now.

The general equation for PSO is described below where is the most optimum solution and the is the swarms most optimum solution.

The position is updated by

The inertial factor represented by w is generally taken in between 0.4 to 1.4 so as to control the velocity so it won’t explode. A large value of inertia weight provides global exploration while the small value of the facilitates the local exploration. The variable is the updated velocity at t+1 iteration and is the velocity at velocity at t iteration. C1 is the self-confidence factor which is generally a constant and ranges from 1.5 to 2, C2 is the swarm influence which also varies from 1.5 and 2. In many cases these are kept constant but in DPSO we are changing the values of these parameters with each iteration. In normal cases the value used is 2. is the local best at that iteration and is the global best found till now. The variable rand generates a random number between 0 and 1. is the updated position of the particles and is the position of particle at previous iteration.

There are many advantages of using PSO some of them are

1. It has no crossover and mutation hence it is computationally less expensive than GA.
2. It's based on intelligence and hence can be applied on wide variety of research and engineering use.
3. The calculation is very simple compared to the other heuristic techniques.

The PSO is quite versatile and can be use by wide variety of linear, nonlinear problems, multi modal problems, however it has some limitations and cannot be used to solve the multi-target tracking problems.

The limitations are stated below:

1. It calculates function evaluation at each particle movement even when it is not in signal range of the target optima or tower. This can lead to higher number of variable evaluation and hence takes more time to run a problem.
2. Even when the signal is sensed it will converge either to the global optima or single optima. Thus, PSO was initially implemented to find global and not multiple optima’s.

**2.2 The hybrid algorithm based on random search and DPSO search algorithm**

This algorithm was proposed by Shoutao Li, Lina Li, Gordon Lee, Hao Zhang in their article “A Hybrid Search Algorithm for Swarm Robots Searching in an Unknown Environment”. It's inspired by the behavior of predator while they are searching for the food or prey(optima’s). When animals search for prey, in the absence of food or possible prey location, the predators search the entire space in certain direction. Once they sense the signal or found the prey they slow down and ponder over that small area or region.

Hence it can be observed that the predator undergoes behavior change while it is searching for the prey. Firstly, the random search which is fast and computationally less expensive and is used until the prey, signal or optima is sensed and it tries to cover maximum search space in minimum time. Secondly, when the prey is sensed it slows down the search process and the algorithm shifts to DPSO, also known as Dynamic PSO. Dynamic PSO is different from PSO in sense that the PSO parameters are time varying and are tuned according to particular scenarios where inertia parameter ranges from 0 to 2 in case of C1 and C2 ranges from 0 to 1. The equation for DPSO is same as that of PSO. The hybrid of random search and DPSO is used to find the single and multiple targets.

1. **METHODOLOGY AND RESULTS**

**3.1 Objective Function**

The objective function for the problem which will be further used for every algorithm discussed in report is such that each target is represented by the Gaussian distribution with their respective mean and variance. The optima or the target lies at the mean of their respective Gaussian distribution. As the Normal distribution spreads to infinity, each particle will evaluate the function value at each iteration which will be small for most of the part and the process will be time consuming. In order to avoid this a threshold value or a standard deviation from mean is used which leaves the infinity part, and the remaining function value is used in the objective function.

The objective function obtained for the given problem is the sum of all the n-target Gaussian distributions. We are trying to find all the local maxima’s or targets in the given region which are nothing but the peaks of the distribution or mean of the distributions. The objective function is given as follows:

= (()|()

=

The figure1 represents the pictorial view of the 4 target Gaussian distribution and figure2 represents the contour plot of figure1.

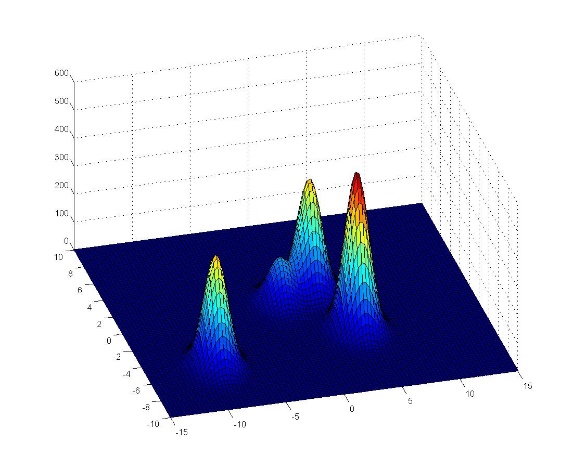


Fig 1 Gaussian distribution of 4 targets with different intensities

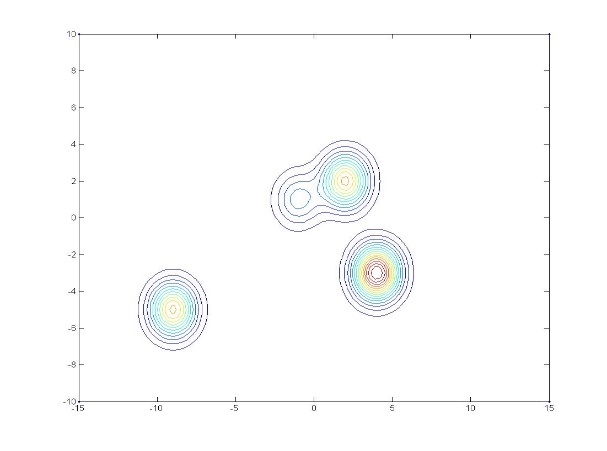


Fig 2 Contour Plot

The results obtained in the paper were good and the algorithm was tested on single and multiple targets. In multiple target system, concept of Bee swarm is used and the whole system of particles is divided into three parts

1) The random search particles or ‘scouts’ which continuously look for targets ,this type of behavior is named as liberty or scouts, 2) an organizational behavior in which the particle who sensed the given target will inform the other particles within the communication range to help detect the optima or the target, they are called as organizers or ‘experienced’ particles 3), the particles which are called upon for help by the organizer to detect the target, these particles are commonly knowns as collaborators or ‘onlookers’.

* 1. **Modified Bee swarm optimization:**

The second approach we are following to solve the problem is modified BSO (Bee Swarm optimization). The Bee swarm optimization was

initially developed in 2005 and emulates the behavior of bees while they are searching for food.

This algorithm is different from PSO in the sense that the particles can move even when there is no objective function evaluation. In addition to that the particles are divided into groups depending upon the signal they detect.

The modified BSO is different from BSO in such way that in BSO there is no acceleration restriction on particles, hence there is no limitation on velocity of particles while in real time scenario the quadcopters have limitation and there is no such thing as infinite acceleration. Hence in modified BSO we are updating the velocity instead of particle position, so as it’s easy to apply constraint on the design variable.

There are two phases of this algorithm. In the first phase, swarm starts at a given location and searches randomly. Members of the swarm are considered as particles. All the particles move with a random step length, but the angles of movement are defined so as to cover the whole domain. Unless a signal is detected, particle tends to move in the same direction. The step angle is defined in the following way:

Where, n is the number of particles and the polygon represents the search space.

When a signal is detected by at least one particle, the algorithm steps into second phase. In the second phase, each particle is given a status according to its search with respect to other particles. Three different status’ are defined for the particles, Experienced, Onlookers and Scouts. The search starts with the status of the all the particles defined as scouts. Scouts are the particles which move randomly in the entire search domain. Always twenty percent of the population is reserved for scouts, so that the search continues even if a target is found. Experienced status is provided to the particles with best fitness. If two close particles have the experienced status, then only the one with best fitness will have that status and the status of the other particle is changed to onlooker. As the particles are searching randomly, there can be more than one experienced particles. These experienced particles will communicate with the nearby scouts and pull them towards the global optimum of the experienced particles, these are named onlookers. Onlookers will blindly follow the respective experienced particles as they don’t have the local best parameter. With multiple experienced particles in search domain, there will be groups of experienced and onlookers which will try to locate the target.

The Pseudo code for the algorithm is shown below. As it can be seen the onlookers only have the global parameter,

*Initialize population with random positions and velocities*

***While*** *termination condition* ***do***

*Compute fitness of each bee in Swarm;*

*Sort Bees Based on their fitness;*

*Partition the swarm into experienced, onlookers and scouts;*

***For*** *each experienced Bee* ***do***

*Update the previous best position*

*Select elite bee for all the experienced bees;*

*Update position by adding velocity to it;*

***End***

***FOR*** *each onlooker bee* ***do***

*Select an elite bee from experienced bee by roulette wheel for onlooker I;*

*Update Position of bee;*

***End***

***For*** *each scout bee* ***do***

*Update position of bee;*

***End***

***End***

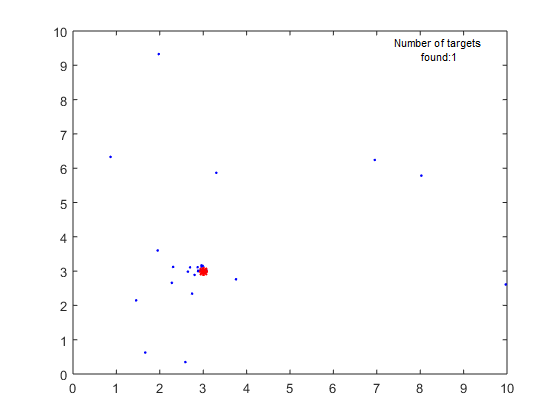


Fig 3 Modified-BSO for one target

We tested this algorithm with single target and multiple targets and it was quite efficiently able to find the all the optima or targets in the given search space. The objective function used in the problem is same as discussed above.

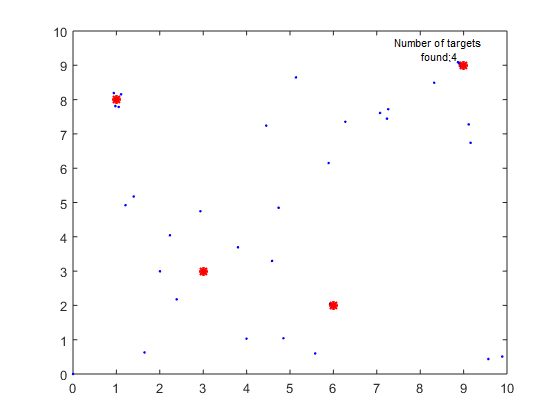


Fig 4 Modified-BSO for multiple targets

**3.3 Modified Firefly algorithm**

This algorithm has a quite different approach to find the optimum when compared to particle swarm optimization (PSO) or Bee swarm optimization (BSO). Firefly Algorithm, a recent discovery of Xin-She Yang (2008) is one of the best nature inspired meta-heuristic algorithms which stimulates the flashing behavior of natural fireflies. Natural fireflies use their flashing power to attract other fireflies. The Algorithm was implemented assuming following things about their behavior.

1. All fireflies are unisexual, so that one firefly will be attracted to all other fireflies.

2. Attractiveness is proportional to their brightness, for any two fireflies, the less bright one will be attracted by (and thus move to) the brighter one; however, the brightness can decrease as their distance increases.

3. If there are no fireflies brighter than a given firefly, it will move randomly. The algorithm was originally proposed for continuous domain problems but its meta-heuristic property allows users to adopt it for discrete problems as well. The pseudo code of the original firefly algorithm is as follows.

*Objective function f(x), x= (x1, …, xd)*

*Generate initial population of fireflies xi (i= 1, 2, ..., n)*

*Light intensity Ii at xi is determined by f (xi)*

*Define light absorption coefficient γ*

***while*** *(t < MaxGeneration)*

***for*** *i= 1 : n all n fireflies*

***for*** *j= 1 : n all n fireflies*

***if*** *( I j > I i),*

*move firefly i towards j;*

***end*** *if*

*Vary attractiveness with distance r via (exp (-γ r2))*

*Evaluate new solutions and update light intensity;*

***end*** *for j*

***end*** *for i*

*Rank fireflies and find the current best;*

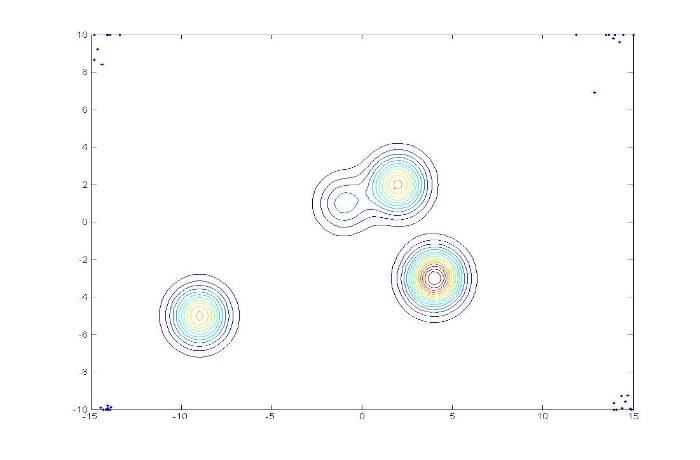
***end*** *while*

*Post-processing the results and visualization;*

The original implementation of firefly algorithm by Xin-She Yang proved that it is superior to both Genetic Algorithms and Particle Swarm Optimization Algorithm. It was very efficient and the success rate was remarkably better than the two algorithms. These conclusions were made only for the continuous domain problems and to check its performance over discrete domains several researches were carried out.

In this modified firefly algorithm, we changed the start position of particles. In conventional algorithm the particles are scattered randomly over the search domain and they are allowed to converge to all the local optima. In this algorithms to mimic the real life scenario particles are deployed from corner of the search space. To make the search even more effective we started particles from 4 different corners for this algorithm which is shown in the figure 5. The light intensity of the flies varies with distance and the fly with high light intensity attracts the other flies closer to it. The dynamics of the particles in this algorithm are not constrained. There are two phases in this algorithm 1) no signal detected by robots, 2) signal detected by robots. When no signal is detected, the swarm moves randomly with certain direction and velocities for each robot. When signal is detected by any of the particles in the swarm firefly algorithm is activated and ultimately robots converge to all the targets. The position update of the particles in firefly algorithm is as follows.

Xt is the current position, α is the randomness parameter, β is the brightness parameter, γ is the absorption coefficient, and εt is the parameter with which randomness can be controlled with increase in iteration.

Fig 5 Start locations for robots in blue dots

The results obtained from using this algorithm are good when compared to Modified BSO and the time taken is less. The parameters used for the algorithm are α=0.6, β=1, γ=2. The algorithm converged in around 180 iterations for four targets. The results are shown in the subsequent figures.

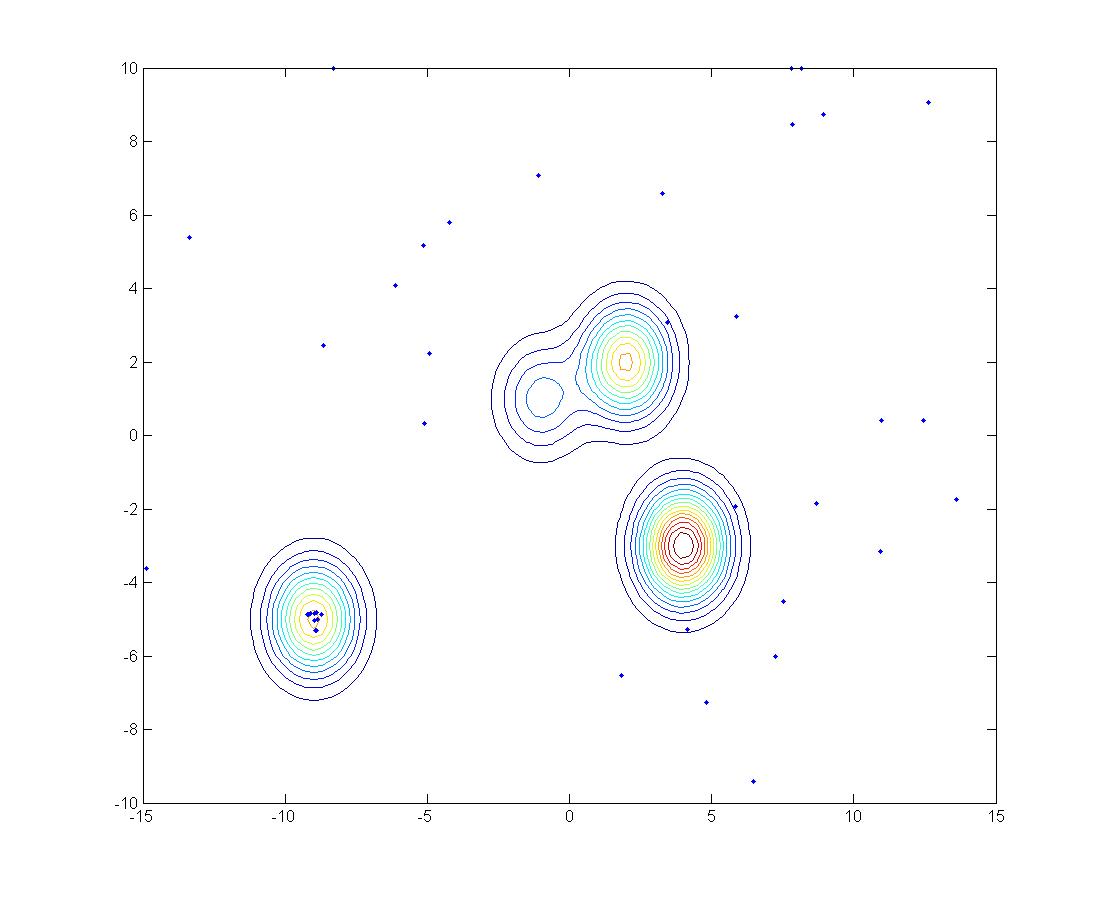


Fig 6 Robots moving randomly

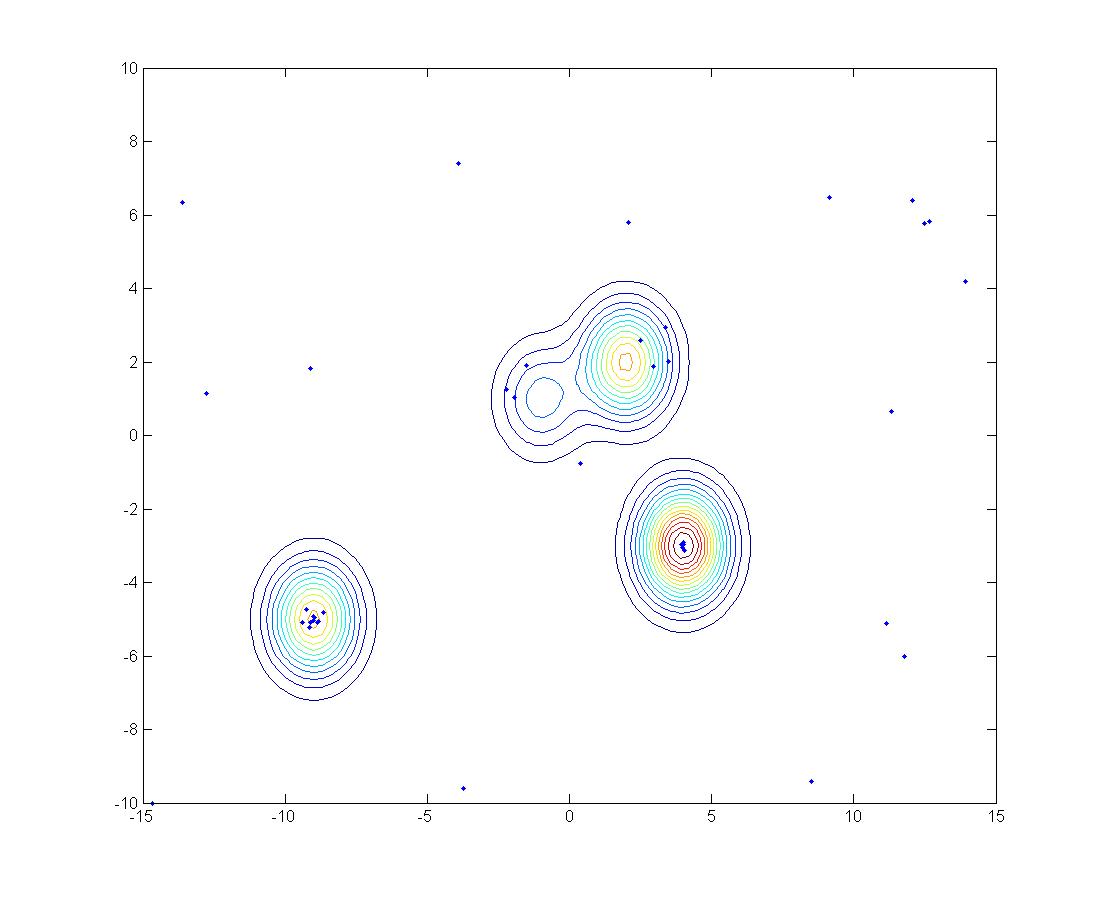


Fig 7 Robots moving towards the targets.

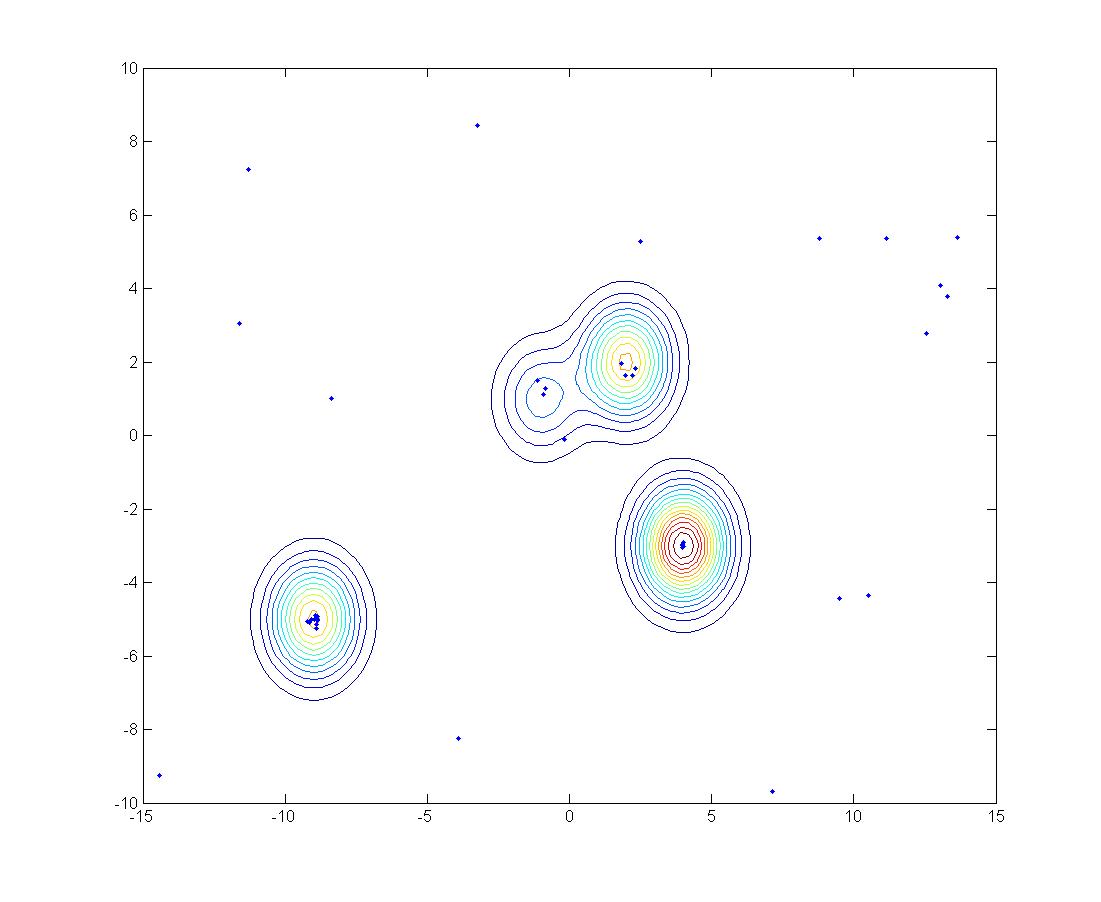


Fig 8 Robots found the target

The advantage of this algorithm compared Group PSO which is going to be discussed in next section is targets in this algorithm need not be shut down once they are found.

**3.4 Group PSO algorithm**

A group of robots can converge to the target much more quickly than a single robot with the help of intra-group cooperation, since the trends of the fitness in the environment are clearer within the group than a single individual. The more the robots in a group, the quicker the group can converge to a target, yet resulting in fewer targets, and the swarm is searching simultaneously since the number of groups is reduced. If the sizes of groups become too large, searching efficiency of the entire swarm declines instead. Therefore, a balanced group size should be adopted and the swarm can thus take the advantage of quick convergence from intra-group cooperation as well as searching several targets in parallel as inter-group cooperation. With a carefully designed strategy, the swarm can search and collect the targets more efficiently. In this algorithm, robots first retrieve information from the environment and their neighbors, then calculate their new movements of this iteration (referred as velocity) according to the sensing data, and carry out the move before next iteration. When the robots detect the target, the target is shut down and the robots found that target. The robots are given random velocity using Levy function after the target is found. This helps in detecting the remaining targets.

The algorithm used is same as PSO, that means the velocity and position updates of the particles is same as PSO. The PSO algorithm is used for each group of robots separately and group operates on its own without any communication from other groups. This is one downside of this algorithm, but works efficiently in finding all the targets.

The algorithm is tested with changes in the parameters and found that the optimum parameters for this algorithm are βl = 0.1, βg = 0.1, and ѡ = 0.93. βl controls the localization, βg controls the socialization and ѡ is the inertial coefficient. The small values of βl and βg are considered keeping in mind that the robots in real life cannot have high accelerations. In order to reduce sudden change in positions of the robots these parameters are have low values. The grid used is 15x15 to test the algorithm and objective function is represented as a Gaussian function as described in section 3.1. The process of the robots moving is shown in the figures below

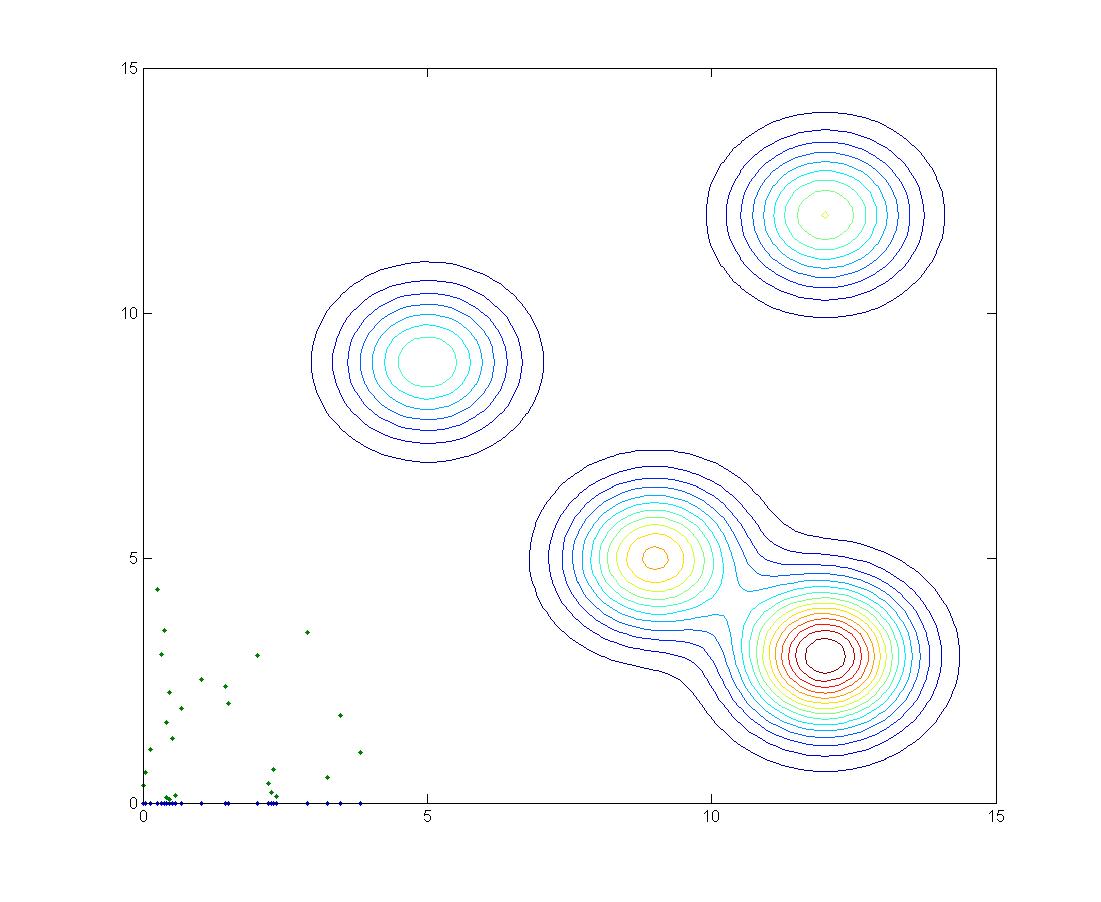


Fig 10 Robots started with some random velocities from initial location

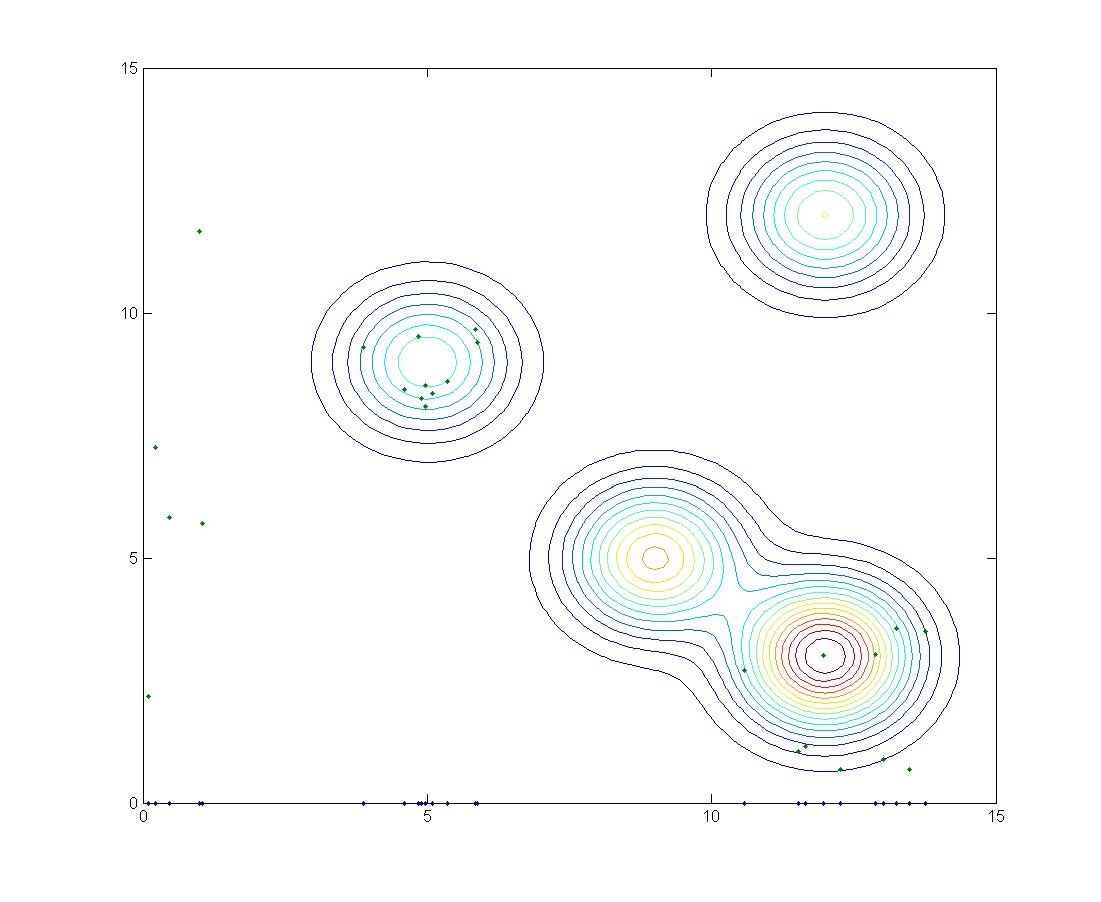


Fig 11 Robots found the signal and moving towards the target

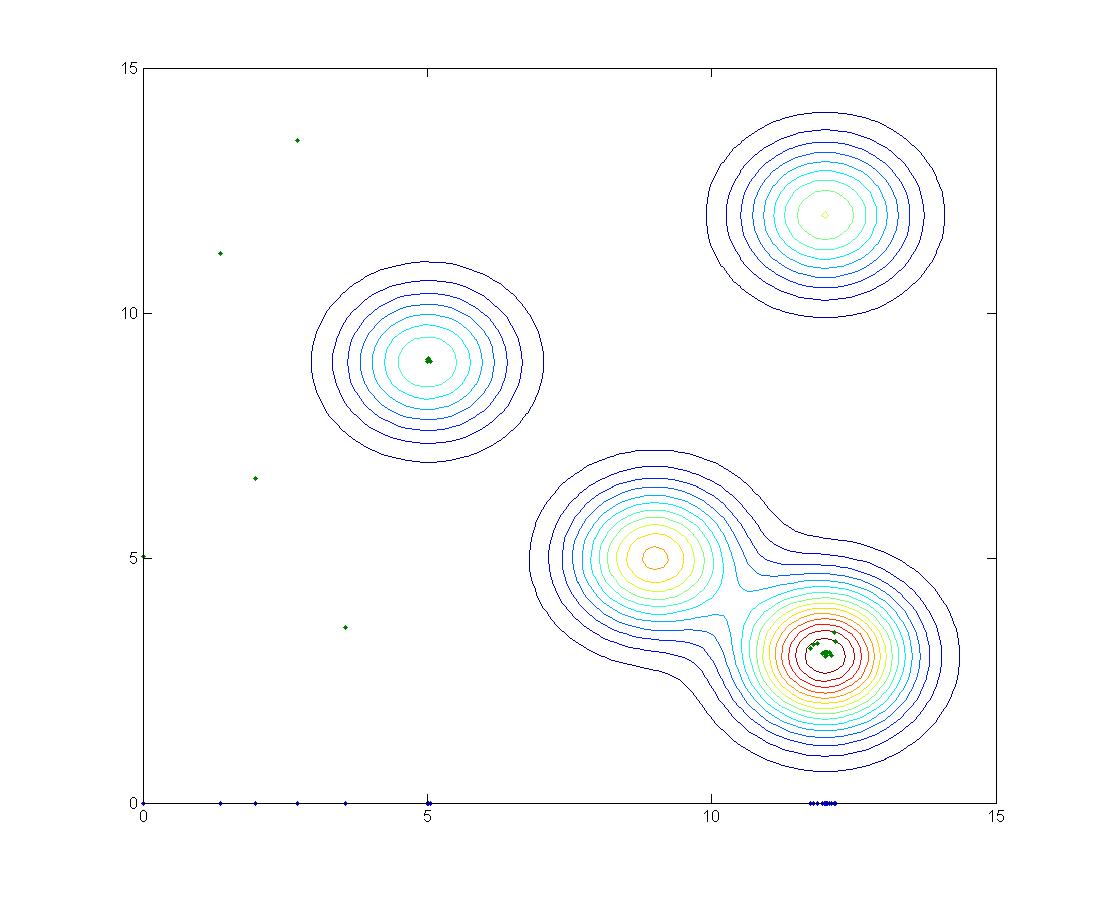


Fig 12 Robots found the target

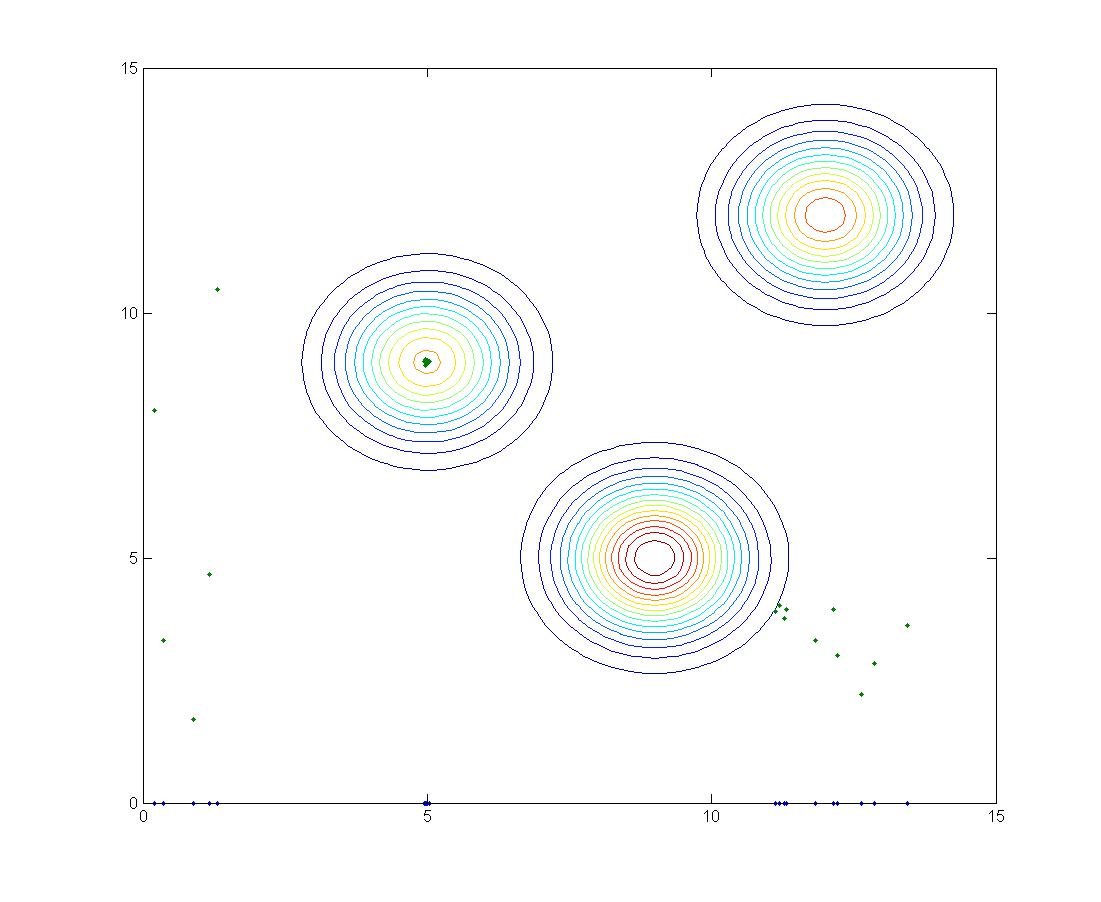


Fig 13 one of the target is shut down after detecting

The results obtained from this algorithm are good, robots were able to find all the targets in the search space without any knowledge of the number of targets and positions. This algorithm can be made much more efficient by including inter group communication, which means multiple targets converging to the same target can be prevented. This will reduce the search time by huge margin.

1. **SUMMARY AND FUTURE WORK**

The PSO algorithm requires evaluation of an objective function for the particle to start moving. In the given problem statement, there are targets located at some points in the search space and their signal diminishes with the distance from the target location. Therefore, a conventional PSO algorithm cannot be used for this problem statement. A combination of random search and swarm algorithms would help in moving the particles randomly in search space unless a target is sensed and tries to converge to the target location whenever a target is sensed. This is employed using three different algorithms. The performance of the three algorithms cannot be compared because all of them operate under different conditions and tested differently. But the main aim of the problem statement is achieved by using all the three algorithms. Firefly algorithm works well when compared to other two algortims.

**4.1 Future work**

Considering the swarm movement in 2 dimensional search space, there can be collision between two quadcopters. In order to prevent this, the predicted location of each quadcopter should be checked with that of others, so that they don’t match at an instance of time. With change in direction at every waypoint, the quadcopter has to decelerate in the previous direction and start accelerating in the new direction. This can cause jerk in the movement, so a better understanding on this can be obtained from the study of dynamics of quadcopter on such traversal.

**REFERENCES**

1) Li, Shoutao, et al. "A Hybrid Search Algorithm for Swarm Robots Searching in an Unknown Environment." PloS one 9.11 (2014): e111970.

2) Tan, Ying. "Group Explosion Strategy for Multiple Targets Search in Swarm Robotics." Fireworks Algorithm. Springer Berlin Heidelberg, 2015. 285-299.

3) Ataei, Hannaneh Najd, Koorush Ziarati, and Mohammad Eghtesad. "A BSO-Based Algorithm for Multi-robot and Multi-Target Search." Recent trends in applied artificial intelligence. Springer Berlin Heidelberg, 2013. 312-321

4) Yang, Xin-She. "Firefly algorithms for multimodal optimization." Stochastic algorithms: foundations and applications. Springer Berlin Heidelberg, 2009. 169-178.

5) Shah, M. A. "Cooperative path planning and cooperative perception for UAVs swarm." (2012).

6) Rajesh, Shah Mihir, Sanman Bhargava, and Soumya B. Mr K. Sivanathan. "Mission planning and waypoint navigation of a micro quad copter by selectable GPS co-ordinates." International Journal 4.4 (2014).

7) Derr, Kurt, and Milos Manic. "Multi-robot, multi-target particle swarm optimization search in noisy wireless environments." Human System Interactions, 2009. HSI'09. 2nd Conference on. IEEE, 2009.