Evaluation and adaptation of swarm algorithms for operational purpose in MUM-T missions

Jannick Simon Weller[1], Sebastian Emmeram Lindner[1]

Abstract
This work focuses on using swarming algorithms to guide unmanned aerial vehicles (UAV's) in military air operations. In military operations, there are many tasks required to fulfill a mission. We want to investigate potential operational benefits using a UAV swarm to cover these tasks. Therefore, we provide insight into swarm algorithms and adapt them to be suited for military purposes.

Keywords
swarming — manned-unmanned teaming — decentralized algorithms

[1] *Institute of Flight Systems, University of the German Armed Forces, Munich, Germany*

Contents

Introduction

In a future combat scenario, a military commander will have to deal with many dynamic threats. Especially when considering that integrated air defense systems are becoming smaller and the opposing force might deploy guerrilla tactics which further challenges the attacking force one needs to find ways to meet those threats. A way would be to saturate the air space by e.g. deploying a swarm of UAVs. However, there will be plenty of other tasks that can be performed by a swarm of UAVs which has a direct operational benefit for a military commander. Those benefits will be evaluated in this work by first identifying tasks that can be performed by the UAV swarm and then secondly finding algorithms that can meet the demands of those tasks. To find algorithms for swarm control, one first has to look at the concept of a swarm and the difficulties of controlling it. Once a swarm of UAVs increases in size the computational effort to steer the drones increases exponentially. Thus one needs a decentralized controller to use large swarms of UAVs beneficially. Moreover, the algorithms need to lead to a self-organized behavior which at best requires minimal communication between the swarm members as well as the swarm and controller. To achieve a broad objective capability, different algorithms will be used for different tasks to ensure that the military commander gets the intended swarm behavior and in conclusion, the wanted results.

1. Theory

The following chapter will give an overview of the different forms of swarm control as well as definitions of the distinctive features.

The coordination of multiple UAVs can be realized in a **centralized** or **decentralized** manner, that outlines the command structure to control the swarm.

The **centralized** approach requires a planning/calculating entity that has communication with all members and therefore has full knowledge about the state of each member. Even though it is a desirable position to know everything as this is a good way of achieving the optimal solution, those controllers have the huge disadvantage of relying on communication. Moreover, due to a single controlling entity, the approach is prone to rapid changes in the environment and the loss of the controlling entity. This approach is both really powerful and exact but at the same time impractical when the members of

the swarm are too far away to communicate effectively with the controlling entity.
The **decentralized** algorithm does not need a lot of communication and is a robust architecture that allows scaling and far distances between the user and the swarm member. This is possible because each member of the swarm derives its actions on its own. However, in conclusion, each swarm member only has its local state and does not necessarily know what the other members of the swarm are doing at that particular moment. Nonetheless, to create a functioning swarm that can operate by itself, one needs decentralized algorithms that can make useful decisions with limited knowledge of the current state of the swarm.
The terms "drones", "swarm" and "swarm intelligence" have become omnipresent, therefor a brief definition will be given to ensure that the terminology is used in the right way.

1.1 Drones

Unmanned aircraft systems (UAS) are an aircraft and its associated elements which are operated with no pilot on board, whereas unmanned aerial vehicles (UAV) refers to only the flying object. These terms form the ICAO[1] circular 328-AN/190 [1] are replaced in the common language by the word drone and this document will accordingly use drones to speak of UAS/UAV and RPAS [2, chap. 2]. Moreover, this work will not have a special focus on any kind of drones such as quadrocopters, small UAV's like the Perdix drone [2] or large drones such as the MQ-9 Reaper drone [3] as the control ideas in this work should work for all platforms. Furthermore, one could argue that algorithms working for flying objects in 3D space should work for land-based systems in a 2D space as well.

1.2 Swarm

The Cambridge Dictionary defines a swarm as *a large group of insects all moving together*. In robot control, many definitions exist e.g. Campion defines a swarm as: "*a group of entities which together coordinate to produce a significant or desired result or behavior*" [3, chap. 2 para. 2] [4]. That means entities moving together are not a swarm unless they have a common objective i.e. a large number of people on a city street doing their shopping would not qualify as a swarm as they are lacking the common objective. Nonetheless, a swarm can be formed by living organisms, robots or both. In this work, the definitions given by Bruce T. Clough [5] are used. Clough points out that in essence, the idea behind building a swarm is: "*Getting a bunch of small cheap dumb things to do the same job as an expensive smart thing*". This definition illustrates that a swarm and therefore its control is not necessarily sophisticated but rather simple and effective. Nonetheless, a more formal definition is given as: "*A collection of autonomous individuals relying on local sensing and reactive behaviors interacting such that a global behavior emerges from the interactions*". Important to note is that this definition showcases that global behavior is not implemented directly but rather arises through the reactive behavior of the swarm entities. Therefore, the **centralized** approach does not qualify as a swarm as stated before, even though one could not see a difference when just looking at a group of UAVs moving in the sky. Clough derives four technical requirements from his definition which minimally need to be mastered by the swarm entities.
First, the swarm entities need to show emergent behavior. This means the entities in the swarm do not know that they are performing a defined behavior nor do they communicate directly with each other to coordinate and achieve a common behavior. The global behavior occurs through the interactions of the individual swarm entities which can be as simple as "*stay close to your neighbor but do not crash into him*" e.g. flocking birds. Moreover, Clough defines four distinctive attributes of emergent behavior. The attributes are: *decentral, implicit, resilient, scalable*. As explained before decentralized control structure means that no single entity is controlling the swarm. Implicit means that the entities are not controlled directly but rather through simple goals. The resulting behavior of a swarm is resilient to imperfections and disturbances. Last but not least scalable refers to the fact that adding or removing swarm entities does not necessarily change the emergent behavior.
The second requirement is that the entities have simple reactive behavior. The conclusion is that they do not need a plan or internal models to assess their situation and their behavior can only be manipulated by using specific triggers.
Third, one needs the behavior architecture of the swarm entities to allow switching between different reactive behaviors. Thus one needs to organize the micro behaviors to ensure that the entities can automatically switch their behavior with the sensor input. Those architectures increase the performance of the entities as the architectures make sure that the needed reactive behavior is used e.g. a subsumptive architecture that can override another behavior in an emergency or a judging architecture that grades the different behaviors constantly a chooses the best-graded behavior at any given moment.
Fourth the local sensing provides information about the environment which can trigger certain behaviors, as stated before. Therefore the reactive behavior is solely created through the local perception of the environment. One does not need a global overview of the situation which can only be achieved by communication. Another interesting fact is that the different sensor data of each swarm entity creates robustness because each entity is sensing the environment slightly differently. Therefore, when the environment changes not all entities change the behavior at the same time due to the variance in the sensing which therefore creates the possibility to have different micro behaviors at the same time. This makes the emergent behavior seem smoother rather than an array of chaotic changes.

[1] International Civil Aviation Organisation

[2] Perdix is a small drone developed by the MIT and tested by the US Government when launched from two F/A-18

[3] The General Atomics MQ-9 "Reaper" is a drone for close air support and was heavily used in Afghanistan by US Forces

1.3 Swarm Intelligence

The notion swarm intelligence is found almost everywhere nowadays ranging from scientific papers to news articles. However, it seems that each author is using one's own definition of the matter e.g. *"The term swarm intelligence is a probabilistic technique for solving computational problems which can be used to get an optimal solution"* as stated by O. Deepa and A. Senthilkumar [6]. Nonetheless, on the International Workshop from 2004 in regard to Swarm Robotics G. Beni gave an approach for a universal definition of the notion swarm intelligence [7, p. 3]. The problem seems to be the term "intelligence" as there is no satisfactory definition of intelligence. Therefore, Beni proposed to judge a system in regard to qualities of intelligence that are relevant to robotics. Hence, one needs to establish those qualities. Beni suggested that the main qualities are unpredictability and the creation of some order and ends up with the following definition: *Intelligent swarm: a group of non-intelligent robots ("machines") capable of universal material computation*. This work will use that definition when talking about swarm intelligence.

2. Potential Swarm Tasks

In this section, a few tasks are presented and explained. The tasks are rudimentary tasks that are often affiliated in the research community with tasks that can be performed by a swarm of UAVs. Moreover, German Air Force Pilots were asked what kind of tasks they would like a swarm to handle for them and those were the most frequently named tasks.

2.1 Attack Object

The main task is to attack an object, which might be a moving target like an infantry platoon or an armored vehicle, or a static target like a house, SAM [4] site, etc.. However, letting the drones fly straight to the target might make them vulnerable against countermeasures like the MANTIS[5] system. To make the swarm of drones a small target for countermeasures the drones should be flying close to each other. However, by letting the drones fly close to each other it is easier to kill multiple drones with airburst ammunition. Therefore, the idea is to let the drones attack the object in a single file. That way they appear to the object they are attacking as one drone and need to be attacked after another which should ensure mission success as the system has to identify a new threat every time it shoots down a drone. However, if the countermeasure is looking at the swarm from the side then the swarm might need to chance formation once it gets attacked.

The attack of the object could happen in different ways regarding which drones are deployed. A drone could have a weapon system which would let the drone attack the target from a certain distance and thereafter could act as a distraction, or the drone itself is the weapon with high explosives which would explode once the object is reached which obviously would mean the drone is no longer available for another task.

[4] Surface to Air Missile

[5] MANTIS is a German system designed to destroy incoming mortar rounds and small airborne vehicles.

2.2 Electronic Counter Measures

Electronic countermeasures (ECM) play a big role in the Suppression of the Enemies Air Defense (SEAD) and are defined as any measures which interfere with the enemy's ability to track airborne vehicles through radar and other sensors. However, since ECM are mostly a side task to reach a higher goal it is ideal to be performed by a swarm of UAVs. The interference with the enemies tracking ability can happen in different ways. One way is by employing stealth technology which absorbs and deflects the incoming radar waves to mask the location of the fighter. Another is to detect those radar waves and to analyze them to send back an identical radar wave that masks the location by implying to the radar site at the ground that one's current location is somewhere else. Moreover, one can jam the radar waves by emitting radar waves in all kinds of spectrums. Last but not least an effective and simple approach to block radar is by having an object close to the radar site. This approach is best for a swarm of drones, as they can block a big part of the radar. The task of the drones is to form a net-like formation that shields other aircraft from detection by flying close to the radar site and therefore blocking its view. To make it harder for the enemy to deploy countermeasures against the formation, the formation should not be static but rather dynamic.

2.3 Search/Reconnaissance

The search of an object is probably the most common objective associated with a swarm because it is intuitive to let a great number of drones search a big area in a short amount of time. Moreover, as mentioned before due to the resilient behavior a swarm of drones is a good asset to search an area with active air defense as the search task is not ended by shooting down single swarm entities in contrast to a teaming approach. Furthermore, the drones used for the task should be cheap to let the enemy waste anti-aircraft missiles against them which are far more valuable. Often it is assumed that the individual drones of the swarm communicate with each other. However, to increase the operation time of the drones and to make it harder for enemies to disturb the drone swarms, the drones should be able to search an area with as little communication as possible. In conclusion, one loses efficiency. Therefore, the task is to search an area in the fastest way, with almost no communication and without flying a predictable pattern. A compromise will need to be found to ensure a successful and efficient search. The best way would be to assess the different criteria regarding the current situation.

2.4 Protect Jet Fighter

The task is to protect the jet-fighter from enemies. Protection, however, has many forms. For example, the jet-fighter could be protected from another jet-fighters radar or a direct attack. However, both scenarios require the drone swarm to be placed somewhere between the enemy aircraft and the own jet fighter. The distance between those, however, determines the instantaneous reaction capability. If one wants to block the enemy from detecting one's position it would be best to place the drone swarm close to the enemy aircraft to block a bigger area and to enable oneself to detect more area without being restricted by the drone swarm as the drone swarm blocks the radar capability of both sides. However, since the drones can sense their environment as well the blocking of the radar capability is a bigger problem for the enemy. If one wants to protect the aircraft from an attack it is more useful to have the drone swarm close by as it is easier for incoming missiles to evade the drone swarm when the target is far away. Moreover, the drone swarm could be used in a dog fight[6] to force the enemy to change its intended course which will lead to a superior position of the jet fighter. Last but not least the swarm could be used as last resort Chaffs when it is not deployed yet.

3. Algorithms

Nowadays there exist hundreds of algorithm which create different swarm behavior. The algorithms range from modeling insects or chemically reacting particles to multi-robot swarms which can move objects together while being guided by an airborne drone. In this chapter, a few algorithms will be introduced and briefly explained to give an understanding of which algorithms were used and adapted to solve the required swarm objectives.

3.1 Boid

The most commonly known algorithm is the Boids algorithm by Craig Reynolds [8] which emulates the behavior of flocking birds. The algorithm consists of three rules and requires position and velocity. The three rules which generate the behavior are separation, alignment, and cohesion. Separation is the rule which pushes entities away from each other. Alignment means that the entities should alter their direction into the overall heading of nearby entities. And last but not least cohesion works directly against separation as it lets the entities move towards the center of all entities in their proximity. These three rules can be used with different scaling factors to affect the behavior of the swarm. The restricting factor of this algorithm is the perception of the entities as all three rules require other entities nearby to generate the overall behavior. Without entities in the perception area, the entities will move in a straight line. The advantage of the algorithm is that it is easy to understand and implement. Research regarding the Boid algorithm ranges from computer game animation of animals to the control of robots and satellites.

3.2 Voronoi Cells

Voronoi Cells are part of the Voronoi-Diagram [9] which is named after the Russian mathematician Georgy Feodosevich Voronoi. The idea is to divide a space into cells according to some fixed points which are known as center points. Each cell has one central point. The cell is defined as all points which are closer to the center point of that specific cell than to any other central point. There is also a more graphical explanation for the emergence of the Voronoi Cells. If one would send a nondirectional signal at the same time from all center points the cells would be created when two signals meet each other. What that means physically is that if one has a moving object at each center position that very object can reach any point in its Voronoi Cells faster than any other object. Therefore, the Voronoi-cells show the area in which a specific object can reach any desired point the fastest. The Voronoi-Diagram is based on complex mathematical functions and therefore not very intuitive. Moreover, the position of the center points needs to be defined for the Voronoi Cells to be calculated. Another drawback is that center points which are close to each other will create smaller cells or cells where there is a big difference in the distance to the edges of the cell. The algorithm is used by the research community to model the biological growth of cells or to divide an area clustered with drones into cells regarding each drone member to increase search performance.

[6] a dog fight is the synonym for a close distance air battle between aircraft which often includes the use of guns rather than rockets

3.3 Pheromone

Pheromone is a chemical substance that is used by animals and plants to communicate with each other. The small molecules are carried through the air and can be used for orientation and tracking. Animals e.g ants use a pheromone to find the best way to a new food source. In the beginning, all entities run around and emit pheromone in the search for something e.g. in the example of ants for food. Once the food is found the entity returns home using the direct route which might intersect with routes other ants have used before. This leads to a clustering of pheromone which increases the strength of the pheromone on that route. Each ant using that route will increase the level of pheromone on that route and therefore tell ants in the proximity that an ideal route from the home to a food source has been established. The disadvantage of using this approach in the digital world is that memory needs to be created where the pheromone is stored digitally. This requires all entities to have access to that memory and therefore requires communication. However, due to the simple idea, a lot of information can be communicated with little effort which increases performance. The idea to use a pheromone approach for the guidance of a drone swarm is fairly popular and is found in many algorithms e.g. in the *ant colony optimization (ACO)* [10] algorithm. This approach can be used to find the best route through a terrain but also to improve the search time of an area by marking a part of the area as searched by adding pheromone. This helps to avoid searching an area twice which therefore increases efficiency and performance. An important factor is the decay rate as the pheromone does not need to stay in a place indefinitely. The decay rate enables the pheromone approach to search an area again at certain intervals. However, one needs to pay attention when setting the decay rate. If the decay rate is too high the pheromone is gone too quickly and the algorithm has no information to work with. The same goes for a too slow decay rate as the whole area is covered in pheromone and therefore the algorithm can not make any deductions and is useless again. So an important task is to scale the decay rate according to the task at hand without making the algorithm useless.

4. Adaption of Algorithms

In the following section, each algorithm will be explained in-depth and the prior requirements will be mentioned. Each algorithm is either triggered by a signal of the control station which requires the swarm to be in signal reach with the station or a preprogrammed trigger e.g. a GPS coordinate. Thereafter the swarm can achieve its task without further communication with the station which increases the tactical capability of the swarm. However, communication is required for status updates and therefore should be kept if feasible.

4.1 Attack

To force the enemy to identify and attack each UAV successively, and as a result gain time, the swarm should attack a target in a single file. To achieve that goal the Boid algorithm 3.1 has been adapted and modified. The algorithm requires the position of static targets once at the beginning and for moving target the algorithm needs to get updated with the current target position. The original separation law of the Boid algorithm is used in this approach. However, the rest of the algorithm logic is different from the original Boid algorithm. The algorithm first calculates the average position of itself

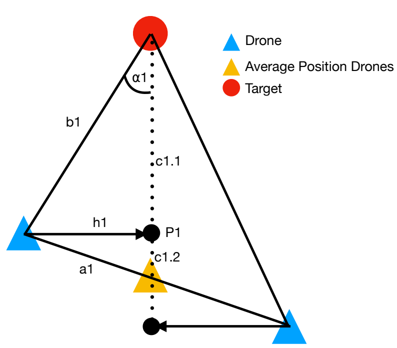

Figure 1. Vector Overview Attack

and all the other drones it can sense. Thereafter the vector b1 and c1 are calculated. The vector b1 is the link between drone and target and c1 is the link between the average position of the drones and the target as can be seen in figure 1. If one knows the vectors one also knows the magnitude of those vectors. The angle between those two vectors is $\alpha 1$ and can be obtained by using equation (1).

$$\alpha 1 = \cos\left(\frac{(-b1)*(-c1)}{|-b1|*|-c1|}\right)^{-1} \tag{1}$$

The length h1 can be calculated by using equation (2).

$$h1 = b1 * \sin(\alpha 1) \tag{2}$$

One can now calculate the length of the vector c1.1 by using the Pythagorean theorem (3) to get the position off the point P1.

$$c1.1 = \sqrt{b1^2 - h1^2} \tag{3}$$

P1 is obtained by setting the magnitude (4) off the vector from the target to the average drone position to c1.1 and adding the position of the target. Thereafter the desired vector h1 can be calculated as the link between the drone position and the position of P1.

$$set\ Magnitute\ to\ Limit:\ divisor = \frac{\sqrt{x^2+y^2}}{Limit} \tag{4}$$

$$x_{new} = \frac{x}{divisor} \quad y_{new} = \frac{y}{divisor}$$

As mentioned before this approach showcases that the drone swarm only needs the information that it now has to switch to that logic and the position of the target. Thereafter, the drone swarm can perform autonomously. However, same as the Boid algorithm this approach is limited by the perception range of the used drones. If the drones are too far away from each other the algorithm will not work.

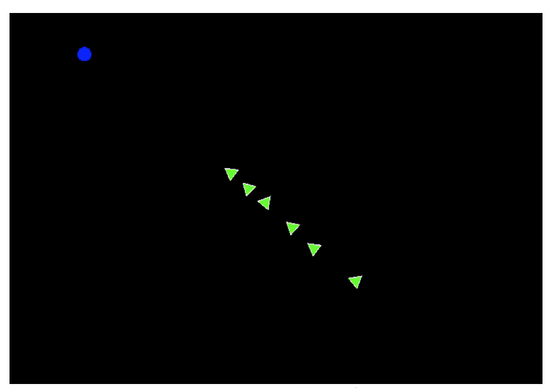

Figure 2. Attack

4.2 ECM

The Electronic Counter Measures (ECM) task is to create a belt to block the biggest area possible of an enemy radar site to ensure that friendly aircraft can act behind the swarm without enemy detection. This task is achieved through an algorithm that is also inspired by the Boid algorithm 3.1 and is similar to the Attack algorithm 4.1. Again the original separation law of the Boid algorithm is used. And again the alignment rule is modified and adapted to create the wanted behavior.

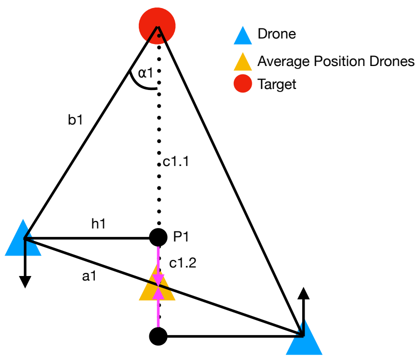

Figure 3. Vector Overview ECM

Furthermore, the algorithm also needs the target position and the drones performing the algorithm need to be able to sense their neighbors and calculate their velocity. As explained in section 4.1 and in equation (1), (2) , (3) and (4) one needs to calculate the point P1 again. Thereafter, the desired vector is the vector from point P1 to the average drone position which is added to the position of the drone itself as can be seen in figure 3. The problem of this algorithm is also that it does not work if the drones of the swarm are out of their sensing range.

Figure 4. Blocking Enemy Radar Site

4.3 Search/Reconnaissance

Search and Reconnaissance is a complex task because there are many factors e.g. size of the searching area, number of swarm members, etc. which have an effect on the success and the efficiency of the task. Therefore, one needs to distinguish between the different factors and find an ideal solution. For example, one can distinguish between a static or a moving target or even simpler a single or multiple targets. Each of these factors affects on the situation and therefore ideally requires its optimized algorithm. Nonetheless, the idea behind the algorithm is simple as it is to search an area in the fastest time by using only as much communication as required. In this section, a few algorithms will be presented which have different capabilities and therefore suit different situations in theory. However, the algorithms still need testing to prove their capabilities.

The first algorithm is derived from the pheromone algorithm 3.3. The algorithm requires at least one memory that all drones of the swarm can access as well as the location and dimensions of the area, which needs to be searched. The idea behind the algorithm is to safe pheromone as a digital unit to the memory. Each drone sets pheromone in its perception area. The behavior of the drones is to fly to spots without pheromone as these spots have not been searched yet. To increase unpredictability, the drone scans the area in a 180° angle ahead and searches for spots without pheromone. One of these spots is then selected randomly and is the new heading of the drone. However, due to statistic reasons, the drones move more or less straight as it is more common that the different headings in the 180° area sum up to close to 90°. This behavior can be observed in figure 5 when looking at the top drone which is not flying in a straight line due to that very reason. Since the algorithm uses one single memory unit a

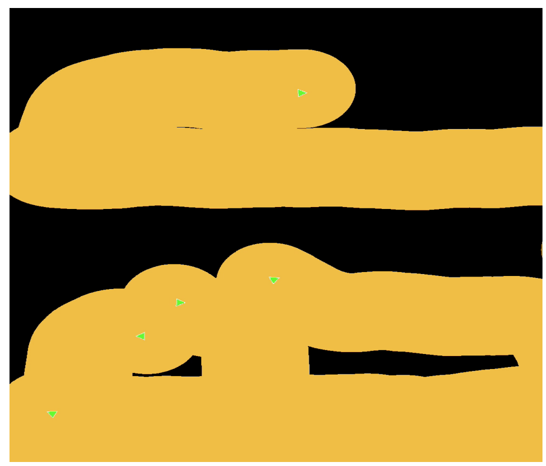

Figure 5. Pheromone aided Search

function is activated if drones can not detect any areas within the 180° which calculates the closes area without pheromone which further increases performance. Moreover, the progress is tracked in $[\%]$ to inform the pilot about the progress. However, the pheromone does not need to be static it could also have a decay rate which enables the user to define in what rates the drones should search areas where they have been before. Even though this helps to find and track moving targets one needs to keep in mind that the efficiency of the search increases with a higher decay rate as the drones can search the same area more frequently without having search the whole area before. The problems of this algorithm are mostly communication if only one global memory is used. However, this problem can be solved by using a memory unit on each drone. Each drone saves the digital pheromone to its memory unit. Once two drones of the swarm come close to each other they communicate their pheromone distribution with each other and add the areas where the other drones have been to already. Even though this approach decreases efficiency it is stable

even when no communication between the swarm members of the global memory unit e.g on the control station is possible. Moreover, the decay rate can increase efficiency when searching moving targets, however, a too high decay rate can also make the algorithm useless as the pheromone is deleted too quickly and therefore the algorithm has no information work with.
The second algorithm is inspired by the Boid algorithm 3.1 again. The idea is to create a behavior like gas molecules under pressure in a gas tank which trade impulses when hitting each other through the elastic collision law. Even though one does not want the drones to crash into each other the behavior can be created by using the separation law of the Boid algorithm. The algorithm requires the performing drones only to be able to sense drones in their proximity and to have knowledge about the area in which it is searching. If two drones come into the perception range of each other they push themselves away from each other which results in the drones flying into another direction as before as can be seen in figure 6. The same happens when a drone is in the perception of the

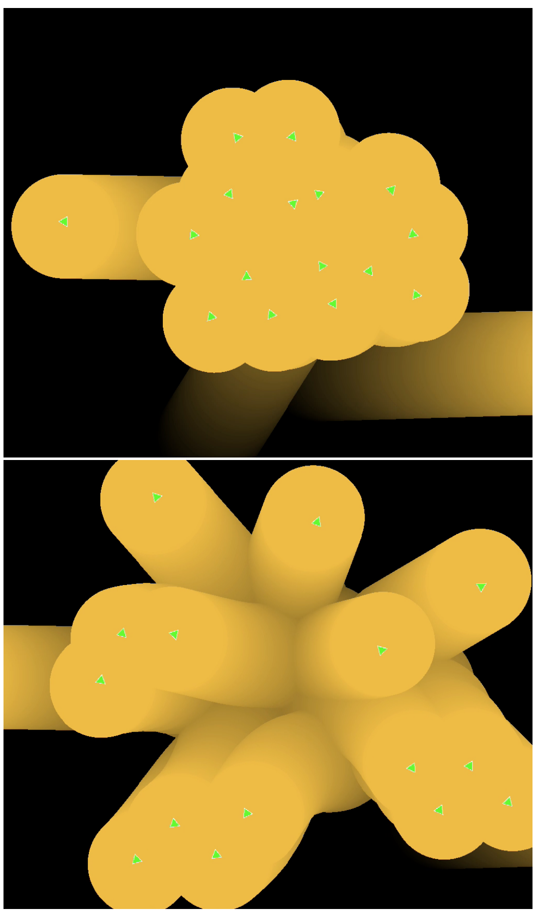

Figure 6. Separation

limits of the search area. This algorithm is very simple as it does not require a lot of information about the environment nor a lot of calculation steps and therefore is really energy efficient. However, there is no guaranteed mission success and the efficiency of the search scales with the dimensions of the area and the number of drones and their speed. Nonetheless, it is a robust algorithm that is almost unpredictable.
Last but not least the third algorithm is derived from the Voronoi cell algorithm 3.2. The algorithm requires information about the search area and a global memory which can be accessed by all swarm members. Moreover, the algorithm can calculate the cells in two ways. The first approach requires a central computing entity e.g. the control station to calculate the different cells. In the case of the control station calculating the cells, the station needs the current position of all swarm members. In the second approach, all swarm members need direct communication with a global memory. They all start registering the area around their position in a circle shape at the same time. Over time the circle shapes increase until they would access memory which is already registered to another drone. The algorithm works by first calculating the Voronoi-Cells as explained in section 3.2 by either using approach one or two. The result can be seen in figure 7. Thereafter, each

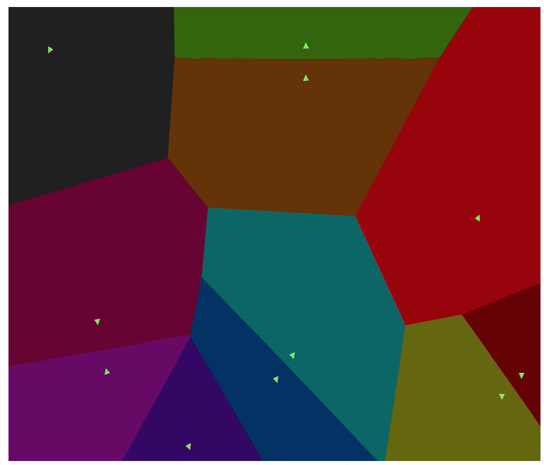

Figure 7. Voronoi Cell aided Search

drone starts searching it personal cell. This algorithm has the advantage of having small specific cells that are searched by one member which can speed up the search process. However, the algorithm has a few drawbacks. For example, if the search is conducted in a high threat environment and a drone is shot down that cell will no longer be searched. Moreover, the cell sizes can differ in size and therefore drones can be down searching their cell while other drones have not yet fully searched their cell. Furthermore, the algorithm requires a good distribution of the drones when the Voronoi cells are calculated and communication with each other and global memory. Nonetheless, under the right circumstances, the algorithm can be effective as well.

4.4 Protect Jet Fighter

The protection task implemented is no specific algorithm like one of the specific tasks presented in section 2.4 but rather the general conclusion that if the drone swarm wants to protect the jet fighter it needs to be somewhere between the enemy aircraft and the jet fighter. For this algorithm the Boid algorithm 3.1 is adapted again. The requirements are that the drone swarm can sense the jet fighter and the enemy aircraft of the jet fighter communicates its position and the position of the enemy aircraft to the swarm. Moreover, the swarm members need to be able to sense drones in their proximity. The algorithm works by calculating a vector between the position of the jet fighter and the position of the enemy aircraft. The vector this then set to half of its original magnitude and either position of the jet fighter is added which returns the position between the jet fighter and the enemy aircraft. Like the cohesion law in the Boid algorithm, this position is now the desired position of each swarm member. To prevent the swarm members from crashing into each other, the separation law of the Boid algorithm is used as well. The resulting behavior can be seen in figure 8. One problem of the algorithm is that the

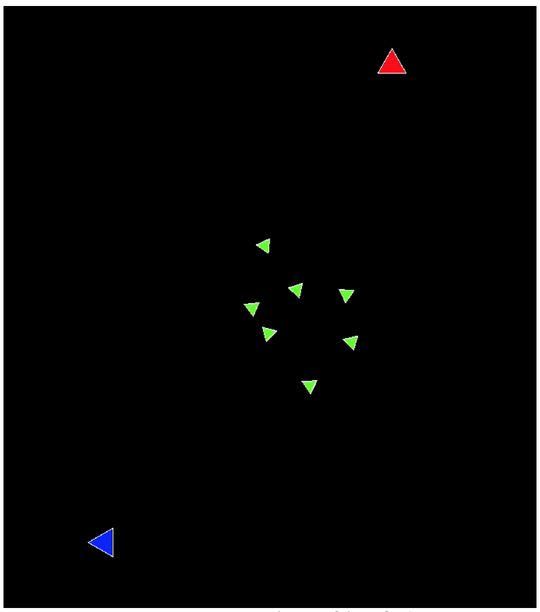

Figure 8. Protection of jet fighter

position of the enemy aircraft needs to be known to the swarm at all times. If the swarm is already between the jet fighter and the enemy aircraft the jet fighter can not sense anything beyond the swarm and is, therefore, relying on the swarm to keep track of the enemy aircraft and to communicate what the swarm is sensing in the area it is blocking for the jet fighter.

5. Conclusion and Prospect

This work investigated swarming algorithms to operate multiple UAVs in a military air mission. Tasks that were eligible for this were Attack, ECM, Search, Reconnaissance and Protect Aircraft. Each task was analyzed and the behavior was derived, which the drone swarm should perform to solve the task. To test that the algorithms work each algorithm was implemented in a simulation program that showed the different behaviors of each algorithm and allows to make deductions about the stability and performance. However, even though the algorithms produce the wanted behavior one still needs to investigate how to embed those capabilities with a military unit. A military unit commander is dealing with a complex and rapidly changing situation which induces high workloads and stress levels. For the commander to be able to control the swarm, it needs to have a self-organizing behavior that was achieved by all presented algorithms which makes them attractive for manned-unmanned-teaming missions. Nonetheless, the commander does not need to only have control over the swarm but rather needs to be able to adjust the emerging behavior of the swarm to ensure that is not causing an unintended result. Therefore, the behavior of the swarm needs to be transparent for the commander to understand. The question remains how those swarm algorithms can benefit from the tactical awareness and understanding of the commander. Further research should be regarding how the commander can tune the algorithms to suit a broad variety of situations in a simple manner.

References

[1] ICAO. *Unmanned Aircraft Systems (UAS)*. ICAO, 2011.

[2] European Aviation Safety Agency. Concept of Operations for Drones A risk based approach to regulation of unmanned aircraft. pages 1–8, 2015.

[3] Mitch Campion, Prakash Ranganathan, and Saleh Faruque. A Review and Future Directions of UAV Swarm Communication Architectures. *IEEE International Conference on Electro Information Technology*, 2018-May:903–908, 2018.

[4] E Teague and RH Kewley Jr. Swarming Unmanned Aircraft Systems. (September), 2008.

[5] Bruce Clough. UAV Swarming? So What Are Those Swarms, What Are The Implications, And How Do We Handle Them?, 2002.

[6] Deepa O. and Senthilkumar A. Swarm Intelligence from Natural to Artificial Systems: Ant Colony Optimization. *International Journal on Applications of Graph Theory In wireless Ad Hoc Networks And sensor Networks*, 8(1):9–17, 2016.

[7] David Hutchison and John C Mitchell. *Swarm Robotics - Lecture Notes in Computer Science*. 2004.

[8] Craig Reynolds. Boids Background and Update. *Http://Www.Red3D.Com/Cwr/Boids/*, page 1, 1995.

[9] Tm Liebling and Lionel Pournin. Voronoi Diagrams and Delaunay Triangulations: Ubiquitous Siamese Twins. *Documenta Methamatica*, I(Extra Volume: Optimization Stories):419–431, 2012.

[10] Marco Dorigo and Krzysztof Socha. Ant colony optimization. *Handbook of Approximation Algorithms and Metaheuristics*, (November):26–1–26–14, 2007.