

A Novel Relationship-based Approach to Swarm Coordination.

Neil Eliot*, David Kendall , Michael Brockway , Paul Oman , and Ahmed Bouridane

Department of Computer Sciences, Northumbria University

*Corresponding author: Dr Neil Eliot, neil.eliot@northumbria.ac.uk

Abstract—Currently, most of the models for swarm coordination are based upon fixed (single value) potential fields. Although, this approach enables predictable behaviours to be created and simplifies the underpinning model, it limits the control of an agent's movement by treating all agents equally. To allow a more configurable model the agent's perimeter status and the status of each neighbour can be considered when applying coordination parameters. This paper proposes a novel model to swarm coordination using a relationship-based approach to enable emergent behaviours to be created resulting in a much-improved structure of a swarm thus making it applicable to specific applications such as reconnaissance problems where a high-density swarm frontier and/or a reduced perimeter density may be required at the frontier. The model is based upon perimeter identification of the structure the use of an agent's perimeter status as an alternative controlling dynamic can be induced. The movement can be modified using three arrays indexed by the agent's status while the agent's vector calculations are modified by these array entries to produce the final movement vector. Extensive experiments were carried out to demonstrate how the new model can implement various behaviours such as packed and expanded perimeters to emerge for a random swarm deployment and how the new model can still operate as the traditional single value potential field model. The results show that by modifying the generated vectors of agents based upon their relationships several novel swarming behaviours can easily be produced, such as void removal, perimeter packing and expansion and can be tailored to relevant swarming applications.

I. INTRODUCTION

When cohesion and repulsion field effects (sometimes referred to as potential fields [2], [9], [12], [22], [23], [17]) are used to create a swarming effect, the stable structures that develop are limited to either straight edges or partial lattices [8]. The maintenance of a well-structured swarm is crucial to effective deployment for applications such as reconnaissance or artificial pollination, where ‘blind spots’ are best eliminated [7], and containment, where the swarm is used to surround an object or region [5]. Over time

swarms form regular shapes [19] and perimeters form of partial lattices that may contain so-called *anomalies*, such as concave ‘dents’ or convex ‘peaks’ [10]. These anomalies contribute to the disruption of an otherwise well-structured swarm. The key, therefore, is to ensure that these *anomalies* are dynamically removed from a swarm whilst maintaining a regular formation.

Perimeter packing is a technique that creates a ‘pull’ effect between perimeter agents. It is dependant upon perimeter agent identification as discussed by Eliot et. al. in [8], [9], [10] and discussed in Section IV-A in more detail.

The aim of this new algorithm is to create a flexible relationship-based coordination technique that allows new emergent behaviours to be realised. Figure 1) shows an agent and its fields. P is the perception field (The range of the sensor array). O is the obstacle field. C is the cohesion field and R is the repulsion field. The new model involves introducing three controlling arrays to the existing potential field model; k_c which modifies the magnitude of the cohesion vector. k_r which modifies the repulsion vector and R which modifies the repulsion field of an agent.

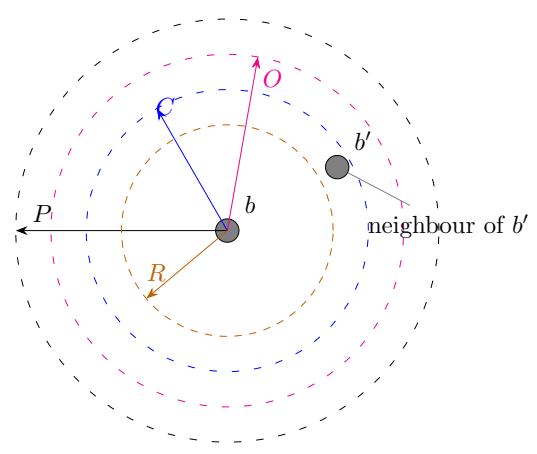


Fig. 1: Agent Fields

II. RELATED WORK

As far back as 1987 swarm theory has adopted the use of field effects/potential fields to coordinate agents [20] and this has continued since then in an attempt to improve the structure of a swarm, coordinate obstacle avoidance, and improve navigation [1], [2], [3], [4], [9], [12], [14], [22], [23]. Improvements to the basic structure of swarms has developed through the likes of a prototype framework for self-healing swarms that was developed by Dai et al. They considered how to manage agent failure in hostile environments [6]. This was similar to work by Vassey and Hinckley, who modelled swarm movement using the ASSL (Autonomic System Specification Language) [26]. This technique was employed by NASA (US National Aeronautics and Space Administration) for use in asteroid belt exploration as part of their ANTS (Autonomous Nano Technology Swarm) project. However, this work is focused towards failure of an agent's internal systems, rather than on the removal of anomalies in a swarm distribution. This need for formation control is also discussed by Speck and Bucci with respect to the diverse applications of swarms and the need to control a swarms structure [24].

In the context of swarm structure maintenance, Roach et al. focussed on the effects of sensor failure, and the impact that has on agent distribution [21]. Lee and Chong identified the issue of concave edges within swarms in an attempt to create regular lattice formations [16], and the main focus of their work is the dynamic restructuring of inter-agent formations. Ismail and Timmis demonstrated the use of *bio-inspired* healing using *granuloma formation*, a biological method for encapsulating an antigen [15]. They have also considered the effect failed agents can have on a swarm when traversing a terrain [25].

This paper proposes an alternative approach to agent coordination that can be used to induce, among other behaviours, a void reduction effect through perimeter packing. This is an extension of the work presented by Eliot et al. [10], Ismail and Timmis [15], [25], and on the work of McLurkin and Demaine on the detection of perimeter types [18]. However, perimeter type identification requires a communications infrastructure to allow the perimeter angle to be calculated. Communications within swarm formations limits swarm sizes and introduces performance problems [11]. The technique employed in this paper does not explicitly require the identification of the perimeter type as

it would limit the size of the swarm[10], [16] and is therefore a reduced perimter detection algorithm to identify *any* perimeter.

III. BASIC SWARMING MODEL

In the Original work by Eliot et. al. the resultant vector of an agent was calculated using Equation 1. Where k_c, k_r, k_d, k_o are weighting factors for the summed vectors associated with each interaction. i.e. v_c, v_r, v_d, v_o for cohesion, repulsion, direction and object avoidance respectively.

$$v(b) = k_c v_c(b) + k_r v_r(b) + k_d v_d(b) + k_o v_o(b) \quad (1)$$

Equation 1 shows the movement vector as a linear combination of a cohesion vector v_c tending to move b towards its neighbours, a repulsion vector v_r tending to move b away from its neighbours, a direction vector v_d tending to move b towards a goal, and a vector v_o tending to steer it away from obstacles. k_c, k_r, \dots are the scalar coefficients of the the linear combination.

This paper does not consider goals or obstacles so we assume $k_d = k_o = 0$ and omit the third and fourth terms.

A. Cohesion

The cohesion component is calculated based on the proximity of neighbours. Where $n_c(b)$ is the set of neighbour agents for b (Eq. 2). The inclusion of an agent from a swarm (S) in by the agent's cohesion field (C).

$$n_c(b) = \{b' \in S : b' \neq b \wedge \|b' - b\| \leq C\} \quad (2)$$

The effect of an agent being within this set is that it will generate a vector that should 'encourage' agents to maintain their proximity. i.e. generate a cohesive swarm. The general weighted formula for agents to maintain their proximity is shown in Equation 1. Equation 3 shows the technique applied to accumulating the vectors that create the cohesive effect. $|n_c(b)|$ denotes the cardinality of $n_c(b)$. This is the component of the overall vector calculation that has the k_c quotient applied to it to allow the cohesion effect to be 'balanced' with respect to other vector influences as described in [8], [9], [10].

$$v_c(b) = \frac{1}{|n_c(b)|} \sum_{b' \in n_c(b)} (b' - b) \quad (3)$$

B. Repulsion

The repulsion component of an agent's movement is calculated from interaction with its neighbours $n_r(b)$ (Eq. 4) in a swarm (\mathcal{S}) that are within the agent's (b) repulsion field (R).

$$n_r(b) = \{b' \in \mathcal{S} : b \neq b' \wedge |b' - b| \leq R\} \quad (4)$$

The repulsion is then calculated as the average of all the vectors created by the agent (b) to the neighbours (b') (Eq. 5) and its proximity ($\|b' - b\| - R$). Where $|n_r(b)|$ denotes the cardinality of $n_r(b)$. This vector is then scaled to 'balance' the effect with respect to other vector influences as shown in Equation 1 where k_c is applied.

$$v_r(b) = \frac{1}{|n_r(b)|} \sum_{b' \in n_r(b)} (|b' - b| - R) (\widehat{b' - b}) \quad (5)$$

Here, $\widehat{b' - b}$ denotes $b' - b$ normalized to unit length.

IV. NEW INTER-AGENT MODEL

In this paper, we propose that the behaviour of an agent should be modified depending on whether or not it is on a *perimeter*. Figure 2 shows a simple swarm. Perimeter agents are highlighted in red. Perimeter-based agents can form part of an inner boundary or an outer boundary. The swarm can also contain non-perimeter agents which are shown in black.

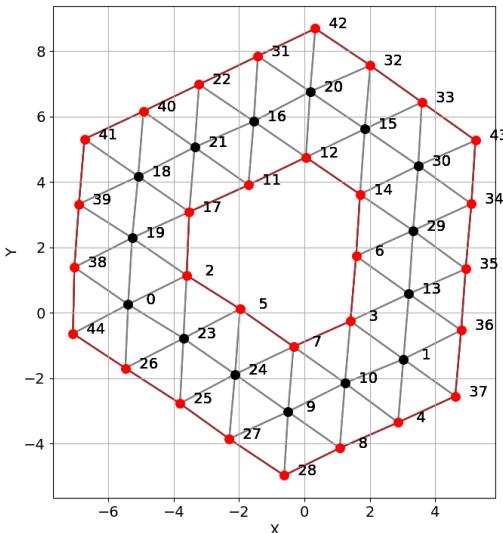


Fig. 2: Outer and inner swarm perimeters.

A. Perimeter detection

Each agent's perimeter status is identified using a cyclic analysis of its cohesion neighbours (Fig. 3). Ghrist et al. discuss a similar technique using sweep angles [13] as do McLurkin et al [18].

We order the cohesion neighbours of an agent b by their *polar angle* (α) with respect to b (Fig. 3).

The polar angle with respect to b of a neighbour, b' , $\alpha(b, b')$, is the counter-clockwise angle that vector $\vec{bb'} = b' - b$ makes with the positive x axis, shown in Figure 3 as α_i and defined by Equation 6.

$$\begin{aligned} \alpha(b, b') &= \theta \text{ where} \\ &\wedge 0 \leq \theta < 2\pi \\ &\wedge \|b' - b\|(\cos \theta, \sin \theta) = b' - b \end{aligned} \quad (6)$$

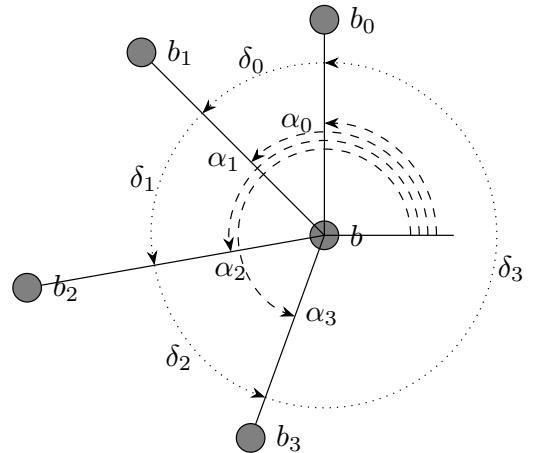


Fig. 3: Agent neighbour angles: $\alpha_0 = \alpha(b, b_0)$, $\alpha_1 = \alpha(b, b_1)$, ... and $\delta_0 = \alpha(b, b_0) - \alpha(b, b_1)$, $\delta_1 = \alpha(b, b_2) - \alpha(b, b_1)$, ...

We denote by $\langle b_0, b_1, \dots, b_{n-1} \rangle_{\leq \alpha_b}$ a permutation of the set of neighbours, $n_c(b)$, that is sorted in non-decreasing order of polar angle, i.e. $\alpha(b, b_0) \leq \alpha(b, b_1) \leq \dots \leq \alpha(b, b_{n-2}) \leq \alpha(b, b_{n-1})$.

An agent b is on a perimeter if it satisfies any one of three conditions:

- 1) consecutive neighbours are not within each other's cohesion field, or
- 2) consecutive neighbours subtend a reflex angle (shown in Figure 3 as δ_3), or
- 3) the agent has too few neighbours.

A function, $\text{prm}(b)$, specifies these conditions formally. Let b be the agent of interest and b', b'' any pair of consecutive neighbours of b in the angle-sorted list $\langle b_0, b_1, \dots, b_{n-1} \rangle_{\leq \alpha_b}$, i.e. $b' = b_i, b'' = b_{(i+1)\%n}$ for some $i \in \{0, \dots, n-1\}$. Then $\text{prm}(b)$ if any one of the following conditions is satisfied:

- 1) $b' \notin n_c(b'')$,

- 2) $\delta > \pi$, where $\delta = \alpha(b, b'') - \alpha(b, b')$ (or $\delta = \alpha(b, b'') - \alpha(b, b') + 2\pi$ if the former is negative), or
- 3) $|n_c(b)| < 3$.

B. R , k_r and k_c

This section discusses the application of the new R , k_r and k_c two dimensional (2×2) arrays structured as shown below:

$$\begin{array}{cc} & \text{False (0)} & \text{True (1)} \\ \text{False (0)} & \left[\begin{array}{cc} i \rightarrow i & i \rightarrow p \\ p \rightarrow i & p \rightarrow p \end{array} \right] \\ \text{True (1)} & \end{array}$$

Where i represents an internal agent and p is a perimeter agent. If we consider Figure 2 then agents $18 \rightarrow 21$ would be internal to internal ($i-i$), $18 \rightarrow 39$ would be internal to perimeter ($i-p$), $39 \rightarrow 19$ would be perimeter to internal ($p-i$) and $41 \rightarrow 40$ would be perimeter to perimeter ($p-p$).

The new model requires each agent to modify their inter-agent repulsion and cohesion vectors based upon their perimeter status and each neighbour's perimeter status. The basic perimeter control technique is shown in Equation 7 where the cohesion and repulsion arrays (k_c , k_r , R) are integrated into $v_c(b)$ and $v_r(b)$.

$$v(b) = v_c(b) + v_r(b) \quad (7)$$

1) Cohesion vector:

Cohesion neighbours are identified as described in Equation 2. The cohesion influence is then calculated as shown in Equation 8.

$$v_c(b) = \frac{1}{|n_c(b)|} \sum_{b' \in n_c(b)} k_c[p_b, p_{b'}](b' - b) \quad (8)$$

where $|n_c(b)|$ denotes the cardinality of $n_c(b)$, $p_b = \text{prm}(b)$, $p_{b'} = \text{prm}(b')$, and k_c is a 2×2 boolean-indexed array of constants that determine the weight of a component of the cohesion vector according to whether the interaction between b, b' is between non-perimeter agents, non-perimeter-perimeter, perimeter-non-perimeter, or perimeter-perimeter agents.

2) Repulsion vector:

The set of repellers of b are defined as Equation 9.

$$n_r(b) = \{b' \in \mathcal{S} : b \neq b' \wedge b' - b \leq R[p_b, p_{b'}]\} \quad (9)$$

where $p_b = \text{prm}(b)$, $p_{b'} = \text{prm}(b')$, and R is a 2×2 boolean-indexed array of constants that

determine the radius of the *repulsion field* for agents in the swarm, according to whether the interaction between b, b' is between non-perimeter agents, non-perimeter-perimeter, perimeter-non-perimeter, or perimeter-perimeter agents.

Now $v_r(b)$ is defined by Equation 10

$$v_r(b) = \frac{1}{\|n_r(b)\|} \sum_{b' \in n_r(b)} k_r[p_b, p_{b'}] \left(1 - \frac{R[p_b, p_{b'}]}{b' - b} \right) (b' - b) \quad (10)$$

where $p_b = \text{prm}(b)$, $p_{b'} = \text{prm}(b')$, and k_r is a 2×2 boolean-indexed array of constants that determine the weight of a component of the repulsion vector according to whether the interaction between b, b' is between non-perimeter agents, non-perimeter-perimeter, perimeter-non-perimeter, or perimeter-perimeter agents.

C. Gap-filling

In addition to cohesion and repulsion vectors, a *gap-filling* vector can also be used to contribute to agent behaviour. Gap-filling vectors have proven useful in quickly reducing internal voids and in controlling the shape of the external perimeter.

A gap-filling vector for b contributes a motion of b towards the midpoint of a gap identified in the perimeter test for b .

Let $\langle b_0, b_1, \dots, b_{n-1} \rangle_{\leq \alpha_b}$ be the cohesion neighbours of b in polar angle order, and let $b' = b_i$ and $b'' = b_{(i+1)\%n}$ be the first pair of consecutive neighbours that satisfy either condition (1) or condition (2) of the perimeter function $\text{prm}()$, then the gap-filling vector, $v_g(b)$, for agent b is defined in Equation 11.

$$v_g(b) = k_g \left(\frac{b' + b''}{2} - b \right) = k_g \frac{b' - b + b'' - b}{2} \quad (11)$$

If there is no such pair of consecutive neighbours then $v_g(b) = 0$.

k_g is a weighting for the gap-filling vector allowing the combination of it with the other motion vectors (cohesion, repulsion, ...) to be “tuned”.

A stricter alternative to this is to choose the first consecutive neighbour pair b', b'' that satisfy condition (1), ignoring condition (2). This would then exclude any reflex angles that create a ‘gap’. Again, $v_g(b)$ is defined by eq (11) if such a pair exists, or 0 otherwise.

D. Resultant vector

The resultant vector is simply the sum of the cohesion, repulsion and gap-filling vectors as shown

in Equation 12 and a resultant swarm segment is shown in Figure 4

$$v(b) = v_c(b) + v_r(b) + v_g(b) \quad (12)$$

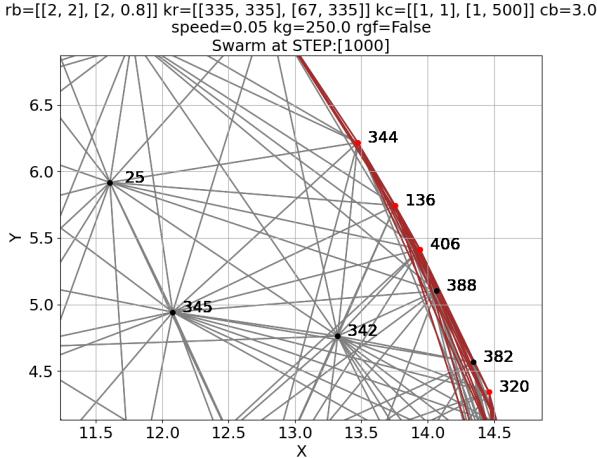


Fig. 4: Swarm Example.

E. Relationship-based swarm effects

The introduction of the arrays allows for specific relationships to effect the movement of agents. Using uniform arrays results a simple cohesion/repulsion based swarm with all agents exhibiting the same properties similar to the original model discussed in § III. However, modifying the arrays for specific relationships can induce emergent behaviours such as perimeter packing as discussed in § V-D1.

1) Cohesion model:

When using Equation 8 one array is used, k_c . This array is used to scale the cohesion vector generated between an agent pair which is proportional to their distance apart, which will be within C as shown in Equation 4. Consider the array shown in Equation 13.

$$k_c = \begin{bmatrix} 1 & 1 \\ 1 & 500 \end{bmatrix} \quad (13)$$

For a given agent pair their perimeter status will be calculated and applied to the arrays. If both agents are perimeter based then the value selected would be $k_c[P_b, P_{b'}] \Rightarrow 500$. If the agent pair were perimeter \rightarrow non-perimeter then the value selected would be $k_c[P_b, P_{b'}] \Rightarrow 1$. This configuration would cause inter-perimeter agents to tend to move towards each other more strongly than any other relationship.

2) Repulsion model:

When using Equation 10 two arrays are used k_r and R . k_r is used to scale the resultant repulsion vector that is generated. R is the radius of the repulsion field and is used to generate the proportion of the repulsion vector that is applied. Therefore consider the following two arrays (Eqs 14 and 15):

$$R = \begin{bmatrix} 2 & 2 \\ 2 & 0.8 \end{bmatrix} \quad (14)$$

$$k_r = \begin{bmatrix} 335 & 335 \\ 67 & 335 \end{bmatrix} \quad (15)$$

For a given agent pair their perimeter status will be calculated and applied to the arrays. If both agents are perimeter based then the values selected would be $R[P_b, P_{b'}] \Rightarrow 0.8$ and $k_r[P_b, P_{b'}] \Rightarrow 335$. If the agent pair were perimeter \rightarrow non-perimeter then the values selected would be $R[P_b, P_{b'}] \Rightarrow 2$ and $k_r[P_b, P_{b'}] \Rightarrow 67$.

V. EXPERIMENTAL RESULTS

The new modelling method allows for a highly configurable swarm. Each configuration will have an impact on the a swarm's structural changes. This can be analysed in several different ways. This includes the magnitude of the interactions between agents and the distances between the agents. The experimental results show the effects on the swarm in terms of inter-agent distances, magnitude metric [9], and the effect on the perimeter. The section will cover four experiments and will include a baseline swarm for comparison.

A. Distance metric

The distance metric is used by many researchers as a method of examining the structure of a swarm [1], [2], [7], [12], [22]. However, due to the new model allowing the field effects and magnitudes to be varied the distance metric will need to be adapted to analyse the agents involved in specific relationships rather than globally, therefore S will be sub-divided into the three relationship categories of S_i , S_p , S_o . Where S_i are the internal-internal relationships, S_p are the perimeter-perimeter relationships and S_o are all the internal-perimeter or perimeter-internal relationships. The distance metric is based upon the mean of a set of agents distances from its neighbours and the standard deviation between those agent sets. The mean is calculated as shown in equation 16 where $\mu_d(S)$ is the mean. The standard deviation is calculated as shown in equation 17 where $\sigma_d(S)$ is the standard deviation. The mean distance value

can be compared to the repulsion field to identify if a swarm is optimally distributed in that the mean value should be as close to the repulsion filed as possible. The standard deviation identifies the overall differences in the distances which can be caused by the swarm agents oscillating. A standard deviation of $\sigma_d(S) = 0$ would indicate that all the agents are equally spaced.

$$\mu_d(S) = \frac{\sum_{b \in S} \sum_{b' \in n_c(b)} \|b' - b\|}{\sum_{b \in S} |n_c(b)|} \quad (16)$$

$$\sigma_d(S) = \sqrt{\frac{\sum_{b \in S} \sum_{b' \in n_c(b)} (|b' - b| - \mu_d(S))^2}{\sum_{b \in S} |n_c(b)|}} \quad (17)$$

Therefore the distance metric for the distribution of a set of agents is both $\mu_d(S)$ and $\sigma_d(S)$. This can be written informally as:

$$\psi_d(S) = \mu_d(S) \pm \sigma_d(S) \quad (18)$$

An example is shown in Figure 11.

B. Magnitude metric

The magnitude metric as defined by Eliot et al. [9] is based upon the relationship between agents and as such is independent of the resultant structure in terms of distances. i.e. agents can be different distances from each other but have the same relationship magnitude. The metric is based upon the mean of a set of agent relationships and the standard deviation between those relationships. The metric is based on inter-agent reationships therefore only the cohesion and repulsion vectors involved in inter-agent magnitudes are used as defined in equation 19. The mean is calculated as shown in equation 20 where $\mu_p(S)$ is the mean. The standard deviation is calculated as shown in equation 21 where $\sigma_p(S)$ is the standard deviation. Due to the metric being based on inter-agent relationships the swarm can be analysed as a whole.

$$P(b) = \|v_c(b)\| + \|v_r(b)\| \quad (19)$$

$$\mu_p(S) = \frac{\sum_{b \in S} P(b)}{\sum_{b \in S} |nbr(b)| + \sum_{b \in S} |n_r(b)|} \quad (20)$$

$$\sigma_p(S) = \sqrt{\frac{\sum_{b \in S} (P(b) - \mu_p(S))^2}{\sum_{b \in S} |nbr(b)| + \sum_{b \in S} |n_r(b)|}} \quad (21)$$

The metric for the internal movement is the mean and standard deviation of the swarm's internal *cohesion/repulsion*. The pair $\mu_p(S)$, $\sigma_p(S)$ may therefore be written informally as:

$$\psi_p(S) = \mu_p(S) \pm \sigma_p(S) \quad (22)$$

C. Baseline Settings

For all the experiments the parameters used to create the basic swarming effect are shown in Table I. Where C is the cohesion field, k_c is the cohesion weighting, R is the repulsion field, k_r is the repulsion weighting and k_g is the weighting applied in the gap reduction algorithm discussed in [10].

Swarming Variable	Value
C	3.0
k_c	[[0.15, 0.15] [0.15, 0.15]]
R	[[2.0, 2.0] [2.0, 2.0]]
k_r	[[50.0, 50.0] [50.0, 50.0]]
k_g	0.0

TABLE I: Swarming effect parameters

D. Baseline

The baseline swarm consists of 400 agents which are distributed over an area of 20×20 units ($-10 \rightarrow +10$) as shown in Figure 5. This is to simulate a randomised drop of agents into an area of interest for the deployment. e.g. A field. Once deployed the agents will be allowed to stabilise. The resultant position for the baseline swarm is shown in Figure 6.

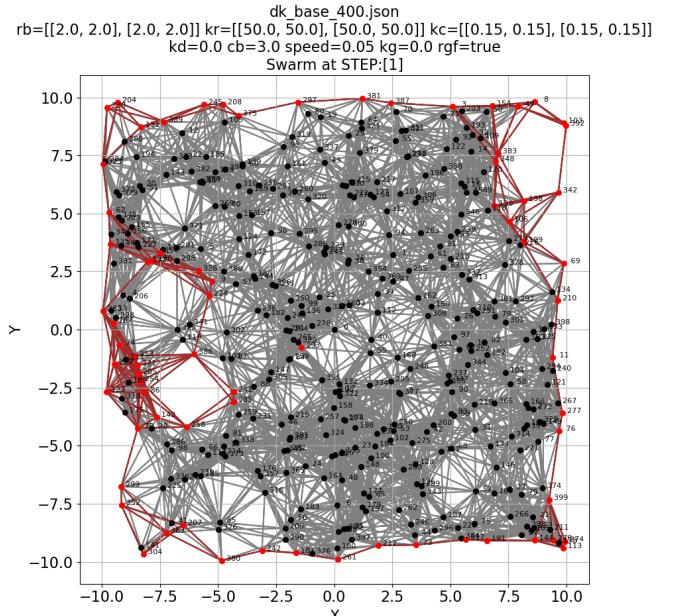


Fig. 5: Baseline swarm.

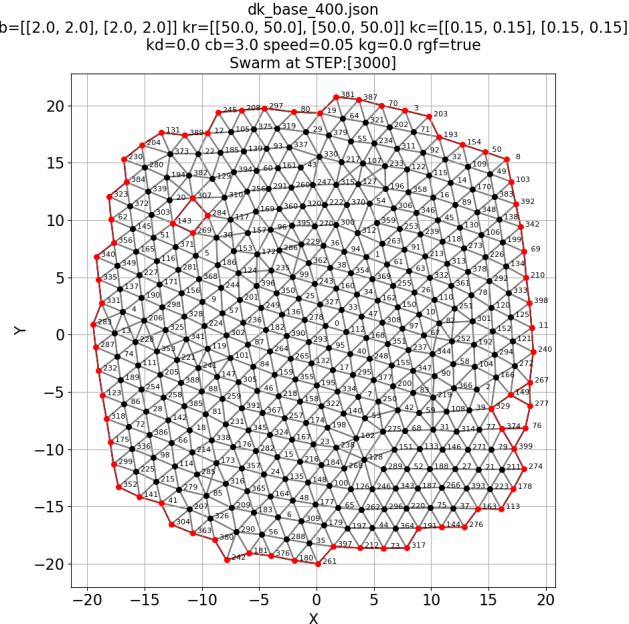


Fig. 6: Baseline swarm.

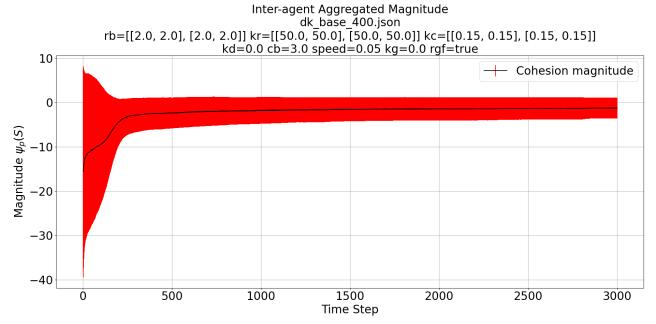


Fig. 7: Baseline swarm (Magnitude).

The distance graph shows that perimeter→perimeter agents and internal→perimeter and perimeter →internal agents settle to a similar average distance of around 2.07 units and the internal agents settle to around 2.04 units. Given that the repulsion field is set to 2 units the swarm looks very stable and is able to form a lattice based structure that changes very little and has a small amount of jitter. Jitter is the slight variation in position that agents exhibit as they move to more optimum positions.

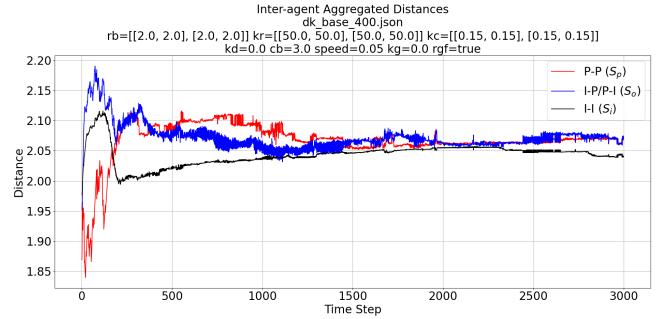


Fig. 8: Baseline swarm (Distance).

When the simulation is ran with no relationship differences i.e. all array values are equal, the changes are identified using a magnitude-based metric [9]. The resultant magnitudes generated are shown in figure 7. The swarm is also analysed based on the inter-agent distances as shown in figure 8. The distance graph shows the different agent relationships types split into S_i , S_p and S_o to allow a comparison with the new model. This state is used as the baseline for the experiments to measure the effects of changing the new arrays. The baseline configuration is equivalent to the conventional swarming algorithms using single value potential fields.

The magnitude graph (Fig. 7) shows that the swarms is relatively stable in that the overall magnitude is around -1 , which indicates it is highly expanded to the repulsion range and there is a standard deviation of around 2 which means the swarms internal magnitude ranges between -3 and 1 so the agents are all oscillating at the repulsion limit.

1) Perimeter Packing: The first experiment with the new model is to create a swarm that has a densely packed perimeter and by default exhibits a self-healing behaviour. This is achieved by modify the perimeter→perimeter agents relationship and the perimeter→non-perimeter/non-perimeter→perimeter relationship. The perimeter→perimeter agent repulsion field is reduced (Eq. 23) and the cohesion weighting is increased (Eq. 25), next the repulsion weighting of the perimeter→non-perimeter/non-perimeter→perimeter agents is reduced to allow the perimeter agents to pull closer together without the next layer of agents reducing the effect (Eq. 24).

$$R = \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 1.8 \end{bmatrix} \quad (23)$$

$$k_r = \begin{bmatrix} 50.0 & 10.0 \\ 50.0 & 50.0 \end{bmatrix} \quad (24)$$

$$k_c = \begin{bmatrix} 0.15 & 0.15 \\ 0.15 & 75.0 \end{bmatrix} \quad (25)$$

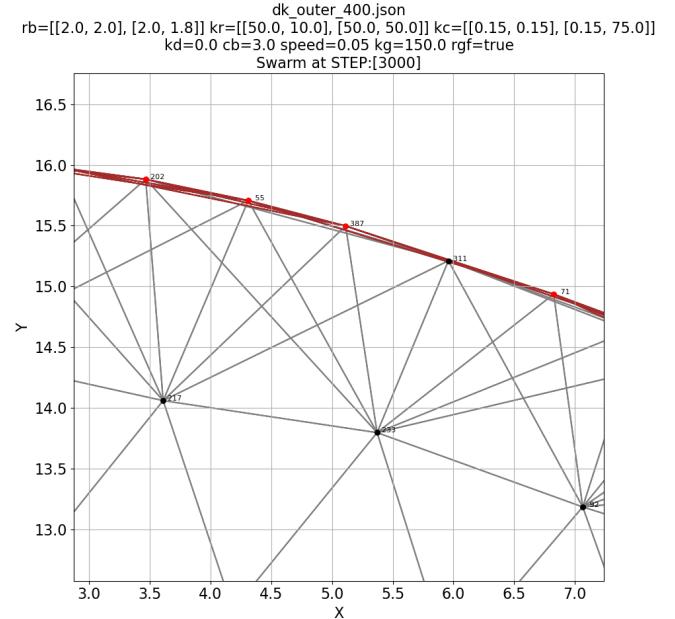


Fig. 10: Packed Perimeter 2.

The resultant effect is that the perimeter agents are able to have more internal neighbours before the aggregate repulsion prevents them moving closer as shown in figure 10. As well as those changes a gap reduction effect is added ($k_g = 100$). This effect includes closing the reflex angle ($rgf = \text{True}$) to smooth the perimeter and cause a compression on a perimeter. Using a high k_g value creates a circular shaped swarm and stabilises the structure (Fig. 9).

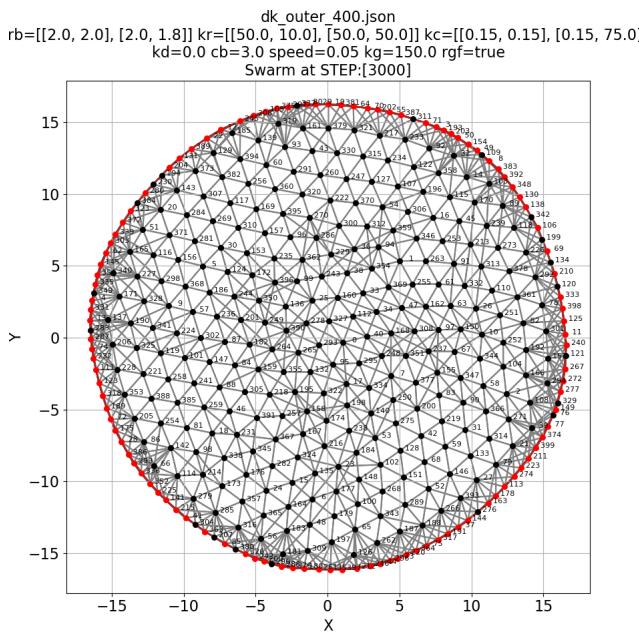


Fig. 9: Packed Perimeter 1.

The effect of the parameters effects the swarm by causing a change in the distribution of the agents compared to the baseline. Figure 11 shows that the perimeter agents (S_p) are now closer together but also shows that the distribution of the p→i and i→p agents (S_i, S_o) are now of a similar distance apart which allows the internal agents to form a regular hexagonal lattice.

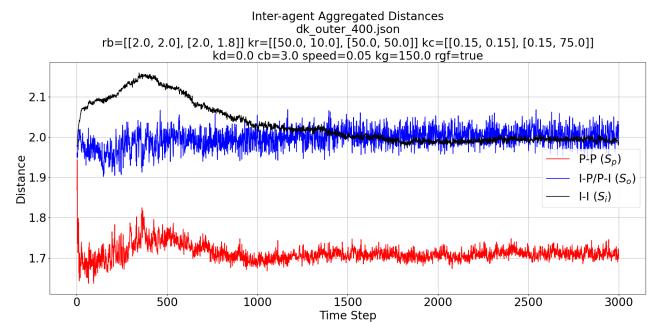


Fig. 11: Packed Perimeter (Distance).

In terms of the magnitude relationships, shown in graph 12. The agents are effected by the compression of the perimeter. This is shown by the increased average magnitude within the swarm. The metric also shows that the packing and compression on the perimeter cause more of a disturbance within the sarm which is shown by the increased standard deviation from the mean magnitude.

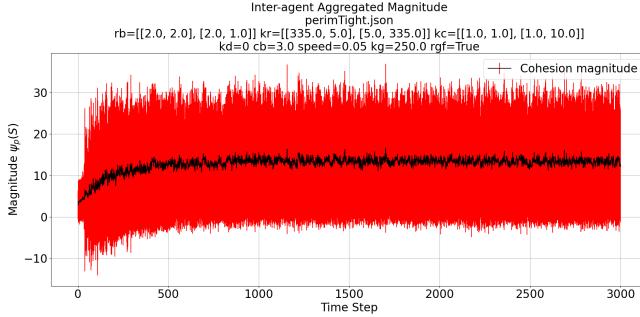


Fig. 12: Packed Perimeter (Magnitude).

2) *Perimeter Expansion*: The basis of this experiment is to use the new model to generate a swarm that allows a swarm to expand from a condensed deployment state and increase the distance between perimeter-based agents while maintaining a dense core. This would be useful when a swarm is moving into an area that could disable or harm agents to reduce the number of agents and would limit the number of agents that would be lost from the swarm. This effect has the added benefit in that the swarm will also exhibit a self-healing property due to the internal compression.

The parameters shown in equation 26, 27 and 28 are designed to create a distancing between perimeter→perimeter agents by holding them steady with an increased cohesion weighting (150.0) and at the same time creating a high degree of repulsion between non-perimeter→perimeter agents (1000.0) i.e. the non-perimeter agents repel the perimeter agents as shown in figure 14.

$$R = \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix} \quad (26)$$

$$k_r = \begin{bmatrix} 50.0 & 1000.0 \\ 50.0 & 50.0 \end{bmatrix} \quad (27)$$

$$k_c = \begin{bmatrix} 0.15 & 0.15 \\ 0.15 & 15.0 \end{bmatrix} \quad (28)$$

As well as those changes a gap reduction effect is added ($k_g = 50$). This effects, which includes closing the reflex angle ($rgf = \text{True}$), has the effect of ‘smoothing’ the perimeter to create the circular shape as shown in figure 13.

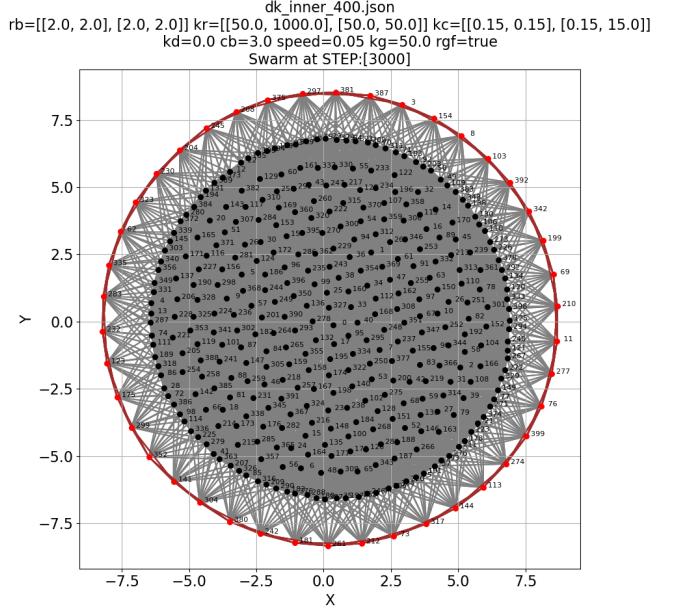


Fig. 13: Perimeter Expanded 1.

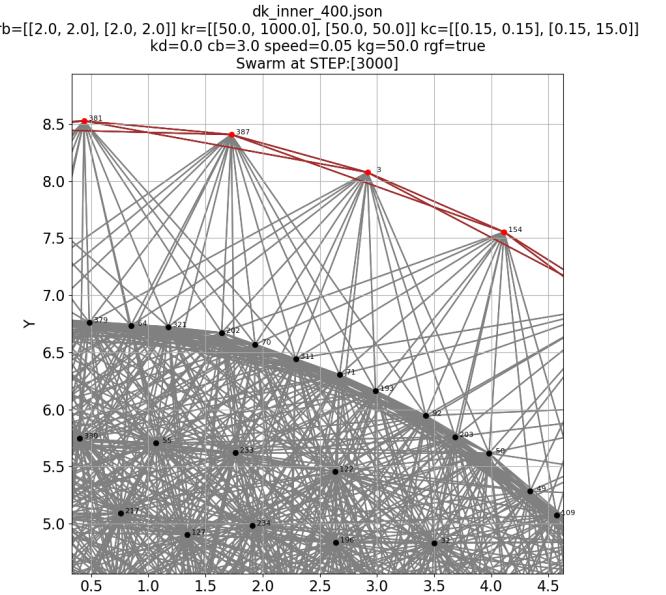


Fig. 14: Perimeter Expanded 2.

The effect on the inter-agent magnitude is shown in figure 15. The initial deployment is in a highly compressed state and the deviation of the magnitude is very high. The swarm quickly expands to produce a swarm that is quite stable with $\mu_p(S)$ being steady but due to the multi-model effect the magnitude spread is wide.

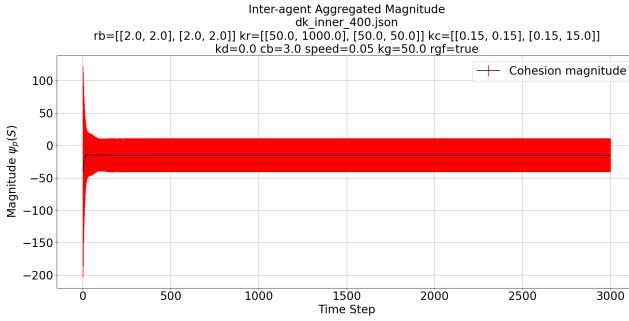


Fig. 15: Perimeter Expanded (Magnitude).

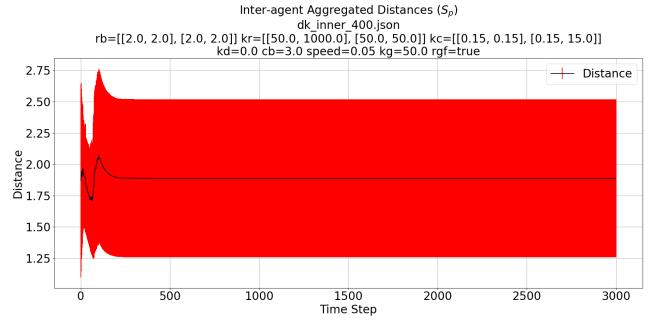


Fig. 17: Perimeter Expanded PP (Distance).

The effect on the inter-agent distances is shown in figure 16. It shows that the perimeter agents (S_p) are well distributed and almost on the limits of the repulsion field (R). However, s_i has a high degree of modality indicated by the large ψ_p of the agents and S_o shows that the internal→internal agents are, on average, holding steady well beyond the repulsion field, again due to the modality.

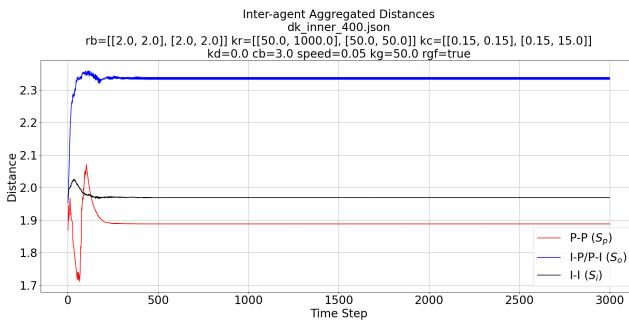


Fig. 16: Perimeter Expanded (Distance).

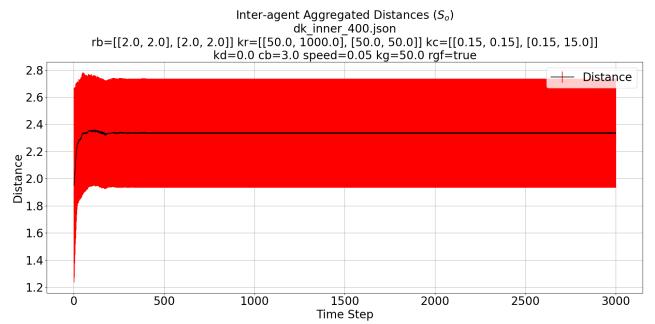


Fig. 18: Perimeter Expanded IPPI (Distance).

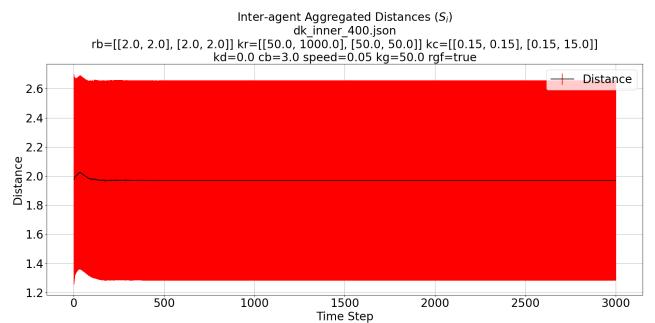


Fig. 19: Perimeter Expanded II (Distance).

Examining the distances more closely it can be seen that the standard deviation is high which indicates that the swarm is multi-modal. A multi-modal swarm is created when agents are able to see multiple agents within the cohesion field beyond the regular hexagonal distribution. The perimeter→perimeter distribution (Fig. 17) is less modal than the perimeter→non-perimeter non-perimeter→perimeter group (Fig. 18) and the internal agents are highly modal (Fig. ref:fig:perimExpandDistanceII).

3) k_g gap reduction: The aim of this experiment is to show the effect of using the gap reduction mechanism on its own to see the part it plays in the new model. The reduction method is an extension of the method used by Eliot et al. [9]. The mechanics are the same but the reflex angle is identified as a gap to add additional control to a perimeter structure. The R , k_r and k_c values are all set to the baseline and $k_g = 100$.

Figure 20 shows the final (step 3000) of the simulation. It shows that the swarm has expanded from its initial state and due to the effects of k_g the swarm has formed a compressed circular swarm.

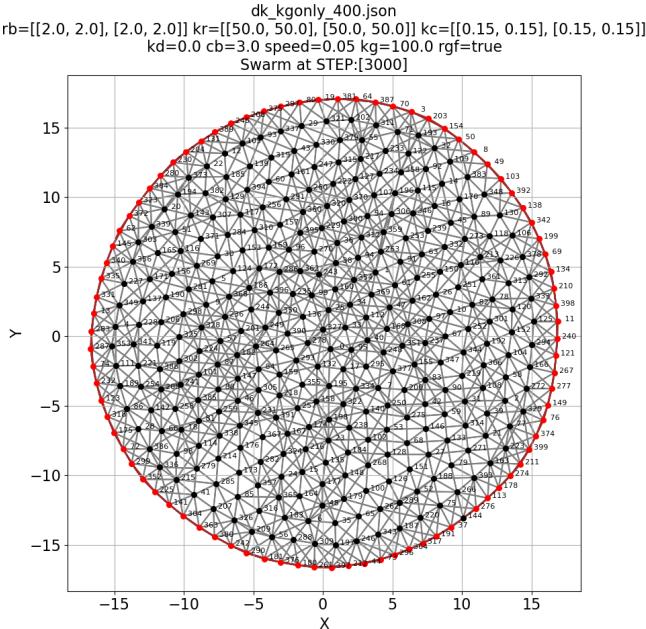


Fig. 20: Gap reduction 1.

Figure 21 shows that although the swarm structure has been brought together as a circular swarm the perimeter is only slightly compressed. This is due to the reflex angle ‘pulling’ the agents in but the repulsion between the perimeter agents prevents them from moving more closely together.

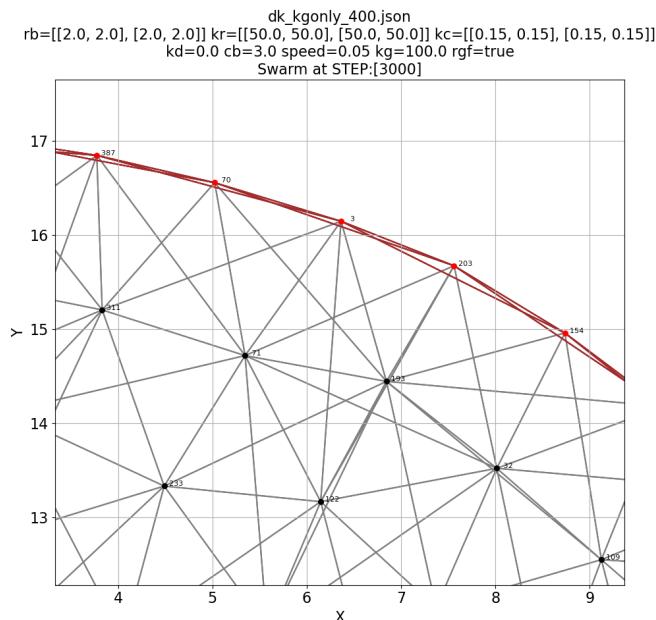


Fig. 21: Gap reduction 2.

Figure 22 shows that the swarm is compressed due to the large standard deviation in the agents.

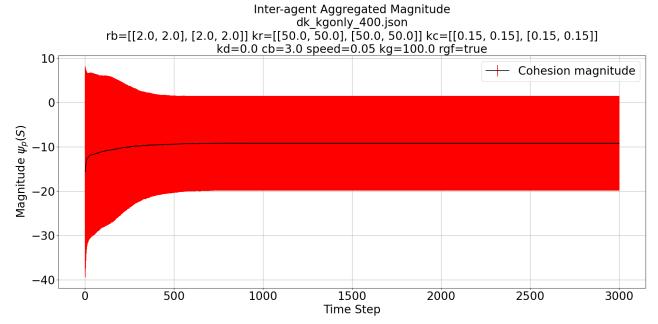


Fig. 22: Gap Magnitude

Figure 23 shows that the swarm perimeter is slightly compressed as the distance is just below the repulsion field, where as the internal and perimeter→non-perimeter/non-perimeter→perimeter agents have an average distance that is beyond the repulsion field which indicated a multi-model/compressed arrangement with the internal→internal agents being slightly less compressed.

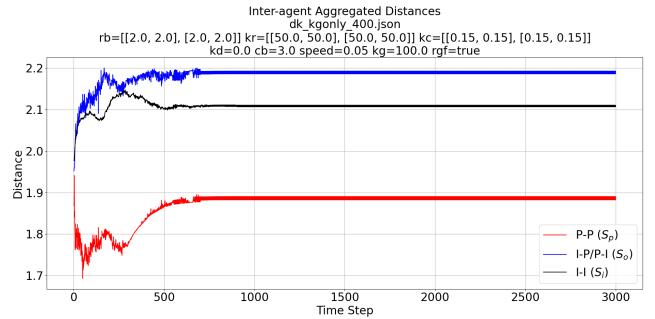


Fig. 23: Gap Distance

4) Void Removal and Perimeter Packing: In this experiment the swarm consists of 400 agents which are distributed over an area of roughly 25 units as shown in Figure 24 and consists of agents that are relatively evenly spaced with large void in the centre of the swarm. The basis for this experiment is to show that the new algorithm can be tuned to heal the swarm by removing the void and also create a packed perimeter formation. The parameters used to create the effect are based upon the baseline swarm parameters with the changes to R , k_r and k_c shown in equations 29, 30 and 31.

$$R = \begin{bmatrix} 2.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix} \quad (29)$$

$$k_r = \begin{bmatrix} 25.0 & 10.0 \\ 50.0 & 50.0 \end{bmatrix} \quad (30)$$

$$k_c = \begin{bmatrix} 0.15 & 0.15 \\ 0.15 & 150.0 \end{bmatrix} \quad (31)$$

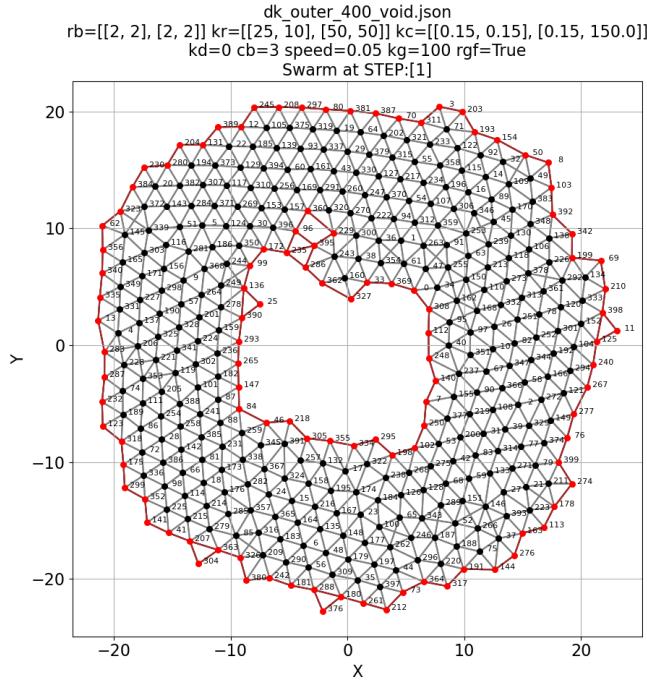


Fig. 24: Void removal and Perimeter Packing 1

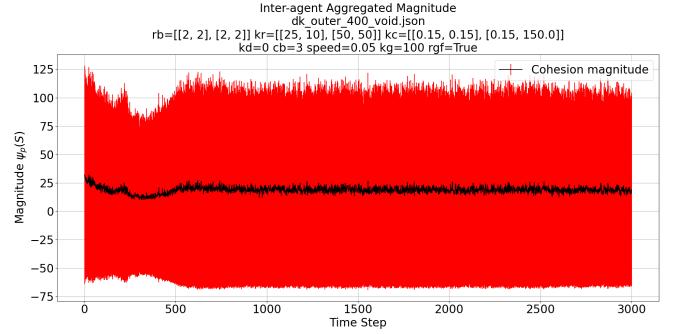


Fig. 26: Void and Perimeter Packing (Magnitude)

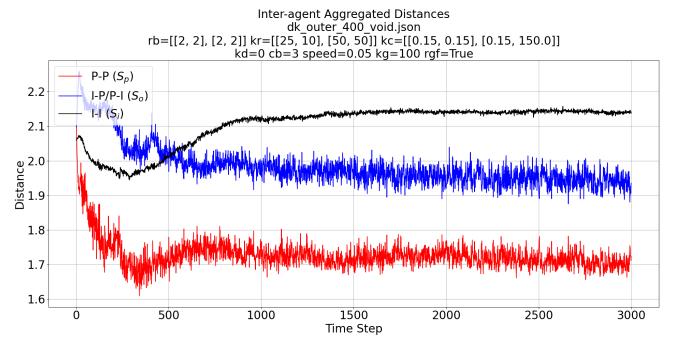


Fig. 27: Void and Perimeter Packing (Distance)

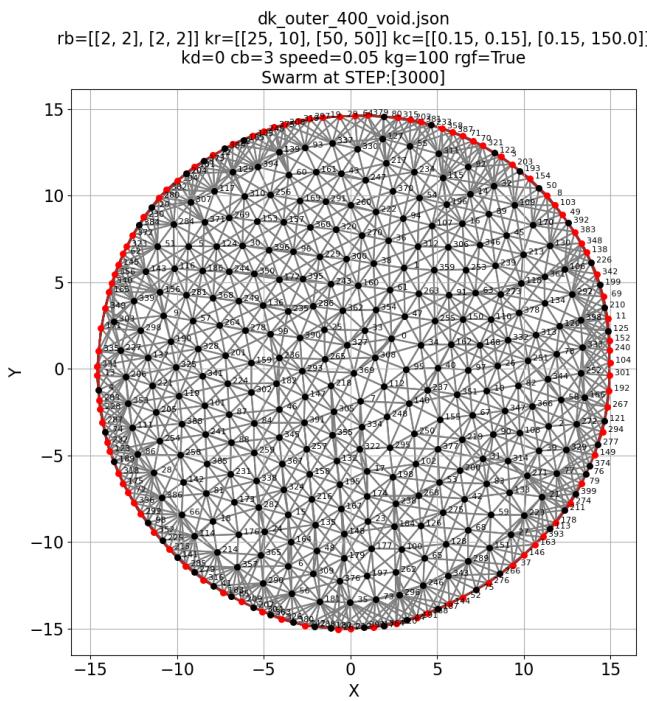


Fig. 25: Void removal and Perimeter Packing 2

VI. CONCLUSIONS AND FUTURE WORK

From the initial simulations it is possible to show that the technique is able to successfully restructure swarms into usable configurations based upon the requirements of 4 distinct relationships with a swarm. Also, by adjusting the gap reducing vector to not use the reflex angles it is possible to allow the perimeter agents to circulate around areas that form naturally on the perimeter. This requires more analysis to fully realise its potential and application. The effect can be seen in the video at <https://youtu.be/E4Q4hk4KrWA>. Additionally it is possible to remove voids and therefore surround obstacles. The metrics show that the algorithm does have an impact on swarm stability to exhibit these new features but the impact is consistent throughout the swarms lifetime as it migrates into different structures.

Going forward the new model will be examined based upon the introduction of direction and obstacles. Initial testing shows that the model holds up well including the improvement in self-healing as demonstrated in figure 28 which shows an obstacle being introduced and removed in a packed perimeter swarm. Figure 29 shows the impact on the Magnitude and figure 30 shows the impact on the inter-agent distances.

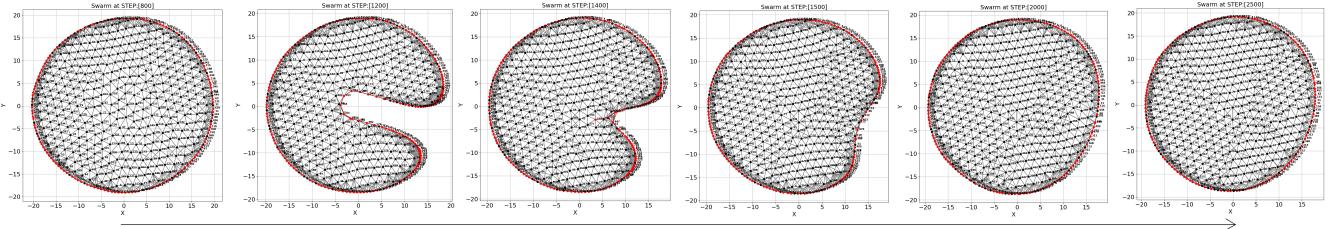


Fig. 28: Simulation of a packed perimeter demonstrating self-healing properties over time

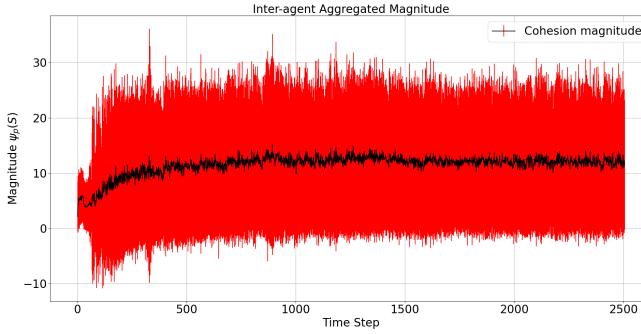


Fig. 29: Perimeter packed - self Healing (Magnitude)

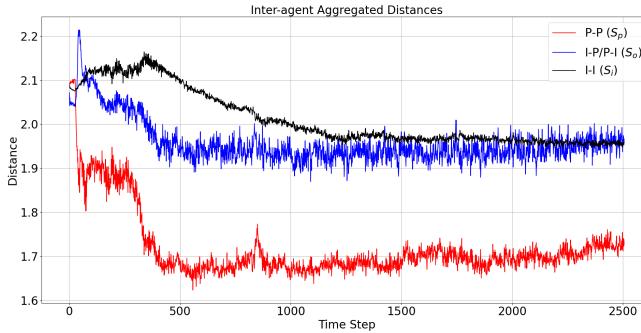


Fig. 30: Perimeter packed - self Healing (Distance)

REFERENCES

- [1] L. Barnes, W. Alvis, M. Fields, K. Valavanis, and W. Moreno. Heterogeneous swarm formation control using bivariate normal functions to generate potential fields. In *Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on*, pages 85–94. IEEE, 2006.
- [2] L. Barnes, W. Alvis, M. Fields, K. Valavanis, and W. Moreno. Swarm formation control with potential fields formed by bivariate normal functions. In *Control and Automation, 2006. MED'06. 14th Mediterranean Conference on*, pages 1–7. IEEE, 2006.
- [3] L. Barnes, M. Fields, and K. Valavanis. Unmanned ground vehicle swarm formation control using potential fields. In *Control & Automation, 2007. MED'07. Mediterranean Conference on*, pages 1–8. IEEE, 2007.
- [4] D. Bennet and C. McInnes. Verifiable control of a swarm of unmanned aerial vehicles. *Journal of Aerospace Engineering*, 223(7):939–953, 2009.
- [5] Y. Cao, W. Ren, and M. Egerstedt. Distributed containment control with multiple stationary or dynamic leaders in fixed and switching directed networks. *Automatica*, 48(8):1586–1597, 2012.
- [6] Y. Dai, M. Hinckley, M. Madhusoodan, J. Rash, and X. Zou. A prototype model for self-healing and self-reproduction in swarm robotics system. In *2006 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*, pages 3–10, Sept 2006.
- [7] K. Elamvazhuthi and S. Berman. Optimal control of stochastic coverage strategies for robotic swarms. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1822–1829. IEEE, 2015.
- [8] N. Eliot. *Methods for the Efficient Deployment and Coordination of Swarm Robotic Systems*. University of Northumbria at Newcastle (United Kingdom), 2017.
- [9] N. Eliot, D. Kendall, and M. Brockway. A new metric for the analysis of swarms using potential fields. *IEEE Access*, 6:63258–63267, 2018.
- [10] N. Eliot, D. Kendall, A. Moon, M. Brockway, and M. Amos. Void reduction in self-healing swarms. In *Artificial Life Conference Proceedings*, pages 87–94. MIT Press, 2019.
- [11] X. Fu, J. Pan, H. Wang, and X. Gao. A formation maintenance and reconstruction method of uav swarm based on distributed control. *Aerospace Science and Technology*, 104:105981, 2020.
- [12] V. Gazi. Swarm aggregations using artificial potentials and sliding-mode control. *IEEE Transactions on Robotics*, 21(6):1208–1214, Dec 2005.
- [13] R. Ghrist, D. Lipsky, S. Poduri, and G. Sukhatme. Surrounding nodes in coordinate-free networks. In *Algorithmic Foundation of Robotics VII*, pages 409–424. Springer, 2008.
- [14] S. P. Hou and C. C. Cheah. Multiplicative potential energy function for swarm control. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4363–4368, Oct 2009.
- [15] A. R. Ismail and J. Timmis. Towards self-healing swarm robotic systems inspired by granuloma formation. In *Engineering of Complex Computer Systems (ICECCS), 2010 15th IEEE International Conference on*, pages 313–314. IEEE, 2010.
- [16] G. Lee and N. Y. Chong. Self-configurable mobile robot swarms with hole repair capability. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1403–1408, Sept 2008.
- [17] X. Liang, X. Qu, N. Wang, Y. Li, and R. Zhang. Swarm control with collision avoidance for multiple underactuated surface vehicles. *Ocean Engineering*, 191:106516, 2019.
- [18] J. McLurkin and E. D. Demaine. A distributed boundary detection algorithm for multi-robot systems. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4791–4798. IEEE, 2009.
- [19] T. Rätz. On the application of the honeycomb conjecture to the bee's honeycomb. *Philosophia Mathematica*, page nkt022, 2013.
- [20] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *ACM SIGGRAPH computer graphics*, volume 21, pages 25–34. ACM, 1987.
- [21] J. H. Roach, R. J. Marks, and B. B. Thompson. Recovery from sensor failure in an evolving multiobjective swarm. *IEEE*

- Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):170–174, Jan 2015.
- [22] F. E. Schneider and D. Wildermuth. A potential field based approach to multi robot formation navigation. In *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on*, volume 1, pages 680–685 vol.1, Oct 2003.
 - [23] J. Son, H. Ahn, and J. Cha. Lennard-jones potential field-based swarm systems for aggregation and obstacle avoidance. In *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, pages 1068–1072, 2017.
 - [24] C. Speck and D. J. Bucci. Distributed uav swarm formation control via object-focused, multi-objective sarsa. In *2018 Annual American Control Conference (ACC)*, pages 6596–6601, 2018.
 - [25] J. Timmis, A. Ismail, J. Bjerknes, and A. Winfield. An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems. *Biosystems*, 146:60 – 76, 2016. Information Processing in Cells and Tissues.
 - [26] E. Vassey and M. Hinchev. Assl specification and code generation of self-healing behavior for nasa swarm-based systems. In *2009 Sixth IEEE Conference and Workshops on Engineering of Autonomic and Autonomous Systems*, pages 77–86, April 2009.