Mathematical Dynamics of Puck Clustering Systems

S. Kazadi
Jisan Research Institute
28 North Oak Avenue
Pasadena, CA 91107
USA
sanza@jisan.org

Abstract

Puck clustering involves the spatial rearrangement of building materials by manipulations of simple agents using local information. Thus far, this clustering has been described in the literature, but has little theoretical underpinning. Most imporant in the deficit in the literature is a solid theoretical description of clustering systems that can actually be built. This paper theoretically examines the long-term goal of clustering systems and provides methodology for the understanding of clustering dynamics as it pertains to the final clustering outcome and the variance in size of clusters of materials under the action of agents. We demonstrate that the dynamics of embodied systems differ from those of non-embodied systems, effectively creating two realms of clustering work.

keywords: swarm engineering, puck clustering, swarm-mediated construction

1 Introduction

In the past several years, there has been an increasing interest in *swarm engineering*, a method of developing of algorithms which use multiple, simple agents to accomplish complex high-level tasks. This approach relies on the generation of simple specific behaviors designed to satisfy a set of conditions which ultimately lead to the accomplishment of the primary objective. As long as the generated behavior set fulfills all of the global requirements, the original desired goal can be accomplished. However, to generate this behavior requires a great deal of effort and analysis, and also a clear understanding of the problem. What separates swarm engineering from the design of classic multiple-agent systems is that the focus is on the general global characterization of the task, rather than the local rules of the task.

Clustering is a process in which simple agents move building materials in a spatially limited area in a random or pseudo-random way. In moving the materials around, in a bordered or unbordered environment, the agents form clusters of the materials in a variety of different ways [4][6][11][12]. A great deal of different methods of clustering have been examined in the literature, both leading to the generation of multiple clusters [9] and leading to the generation of single clusters [2]. The first theoretical work on this appeared in 2000 [9] at which time the *single cluster condition* was derived. This condition stated that in order to produce a single cluster, the clustering system in question had to exhibit a *monotonically decreasing g functional*, where g is the ratio of the likelihood of an agent for puck pickup (f) and the likelihood for puck drop off (h) as a function of the cluster size. In this study, along with most of the previous work, the generation of single clusters has been the primary focus, along with the generation of conditions under which single and multiple clusters would emerge. We view this as the first step in a larger cluster-based construction series in which clusters will come to be used as markers for construction and as construction elements. Cluster-based construction can be achieved if the clusters are created of particular predetermined sizes, placed in specific relative positions, and used as a first step in generating more complex structures. While this would seem to have been completed by [2], the work there was limited to clusters and swarms in which perfect information was available. It is not clear that the same type of behavior would necessarily occur for systems of imperfect information; little is known of the dynamic properties of clustering systems in the presence of puck clustering swarms whose designs do not provide perfect cluster size data. It is not clear, therefore that the variance control exhibited in [2] exists in such systems.

The remainder of the paper is organized as follows. In Section 2, we describe theory behind clustering and multiclustering systems. Section 3 examines the design of a simple agent protocol which can be used to accomplish clustering and is extremely simple and decentralized. Section 4 explores the use of the agent design protocol in an embodied clustering model. Finally, Section 5 offers some concluding remarks.

2 Clustering Systems

We are concerned with minimalist engineering in which a specific global goal is achieved using the simplest possible agent. In puck clustering systems, this translates to using simple agents bereft of global knowledge, processing, memory, and sophisticated sensory capability. The minimal design is exceedingly simple, though the simplicity can force the driving force of the system to be other than rationality. In our clustering systems randomness is the driving force behind the clustering of building materials. That is, systems that generate clusters of pucks do so because their current configurations are perturbed, and the perturbations are caused by random behaviors taken by agents within the system. These disturbances continually occur, and their overall effect is to cause the generation of one or more clusters.

One can imagine examining a system of this type by whittling down the basic agent behavior to a set of systemic properties. These properties govern how the clusters interact with one another. Viewed in this way the agents reduce to carriers of interactions in much the same way that force carriers cause interactions in current theories of basic forces. Viewed at the systemic level, a clustering system may be characterized as in Figure 1.

The clustering system may be represented as a medium which contains multiple clusters (or single clusters). The interface between these clusters and the medium defines the dynamics of the system. Then, what we are investigating is the number of stable states, their configurations, and their stability.

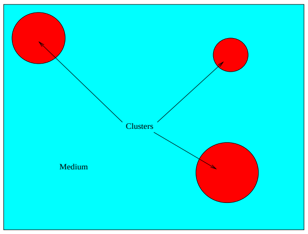

Figure 1: This figure represents the systemic view of clustering systems.

Since our agents are probabilistic entities, their interactions with clusters may be viewed probabilistically. Two sources of random behavior exist. The first source is the interaction of the agent with the cluster, while the second is the behavior of the robot once the cluster's properties have been determined. The first source is a reflection of the inability of the agent to accurately determine the cluster size due to sensor inefficiencies, behavioral limitations, or other problems. The second property is a result of the agent's internal design, and may or may not be random, as the particular design dictates. We concentrate in this paper on the second case, though we comment when appropriate on the effects of perfect information.

2.1 Clustering and Multiclustering

First, we note that an agent interacting with a cluster can be viewed as having the perception that the cluster has a property κ which may or may not be accurate. In particular, given a particular agent's design, the agent has a well-defined probability $\rho$of perceiving any cluster of size $N_o$as a cluster of size N. We can represent this as $\rho_{N_o}(N)$ which we hereafter call the *design probability*.

As we are investigating a swarm of individual agents, the important characteristic in investigating the swarm's behaviors is the average perception of the size of the cluster across the swarm. For large swarms, this may be approximated by

$$\langle N\rangle = \int_N \rho_{N_o}(N)\,dN \tag{1}$$

We verify this in Section 4.2 for a particular form of ρ.

It has been shown elsewhere [11] that the minimal condition for single cluster clustering in any system is that the equilibrium density of puck carrying agents around smaller clusters is higher than that around larger clusters. Put mathematically, this may be illustrated by

$$\delta(N_1) > \delta(N_2) \tag{2}$$

whenever $N_1 < N_2$. Since the density is a function of the perception, we may assume that the perception is a function of the design probability. Thus, we have

$$\langle\delta(N_o)\rangle = \int_0^{N_{max}} \delta(N)\,\rho_{N_o}(N)\,dN \tag{3}$$

It has also been shown elsewhere [11] that

$$\delta_f = D\frac{1}{1+\frac{H}{F}} \tag{4}$$

where $H(N)$ represents the probability that a puck will be placed in a cluster, given the size, and $F(N)$ represents the probability that a puck will be picked up from a cluster of size N. This density is valid when using the reaction g and uniform dispersion of the robots. This can be rewritten as

$$\delta_f = D\frac{F}{F+H} \tag{5}$$

In general the robots do not know the actual size of the cluster, and have some probability of thinking that a given cluster of a particular size is a particular size. Thus, the average density of a puck holding robots around any cluster is

$$\langle D\rangle = D\int_0^\infty \frac{F(N)}{F(N)+H(N)}\rho_{N_o}(N)dN \tag{6}$$

where $\rho_{N_o}(N)$ is the probability that a cluster of size N_o will be mistaken as a cluster of size N. Kazadi et. al. [10] found that the condition leading to a single cluster is that the density of puck carrying robots around large clusters should be smaller than the density of puck carrying robots around smaller clusters. This translates to

$$\begin{array}{c} D\left(\overrightarrow{x_{N_1}}\right)\int_0^\infty \frac{g(N)}{g(N)+1}\rho_{N_1}(N)dN \\ < D\left(\overrightarrow{x_{N_2}}\right)\int_0^\infty \frac{g(N)}{g(N)+1}\rho_{N_2}(N)dN \end{array} . \tag{7}$$

whenever $N_1 > N_2$. If $D\left(\overrightarrow{x_{N_1}}\right) = D\left(\overrightarrow{x_{N_2}}\right)$[1] this condition reduces to

$$\int_0^\infty \frac{g(N)}{g(N)+1}\rho_{N_1}(N)dN < \int_0^\infty \frac{g(N)}{g(N)+1}\rho_{N_2}(N)dN \ . \tag{8}$$

This condition may be known as the *generalized clustering condition*. It is easy to see that if $\rho_{N_o}(N) = \delta(N-N_o)$[2] the condition reduces to a much simpler equation, in agreement with that found in [11]. The calculation is as follows:

$$\frac{g(N_1)}{g(N_1)+1} < \frac{g(N_2)}{g(N_2)+1} \tag{9}$$

which reduces to

$$g(N_1) < g(N_2) \ . \tag{10}$$

In essence what the generalized clustering condition means is that the g functional must decrease fast enough so that the linear combination of g functionals of a larger cluster are smaller than the corresponding linear combination from a smaller cluster. This is always the case if the probability is a delta function and the g function is monotonically decreasing.

This last expression gives the general condition that a single cluster will result from dynamics of a swarm of agents of a given design.

On the other hand, we might wish to limit the size of the clusters, making multiple identically sized clusters. We may do this if

$$\left\{\begin{array}{cc} \begin{array}{c} \int_0^\infty \frac{g(N)}{g(N)+1}\rho_{N_1}(N)dN \\ < \int_0^\infty \frac{g(N)}{g(N)+1}\rho_{N_0}(N)dN \end{array} & if\ N_1 < N_o \\ \begin{array}{c} \int_0^\infty \frac{g(N)}{g(N)+1}\rho_{N_1}(N)dN \\ > \int_0^\infty \frac{g(N)}{g(N)+1}\rho_{N_0}(N)dN \end{array} & if\ N_1 \geq N_o \end{array}\right. \tag{11}$$

which of course, in the case that $\rho = \delta$, reduces to

$$\left\{\begin{array}{cc} g(N_1) \leq g(N_0) & if\ N_1 \leq N_o \\ g(N_1) \geq g(N_0) & N_1 \geq N_o \end{array}\right. \tag{12}$$

which together mean that g is monotonically decreasing up until $N = N_o$and then increasing for $N > N_o$. This gives the minimal condition for multiclustering to occur.

---

[1] This is generally true if the robots are able to move freely from one cluster to another.

[2] δ is the Kronecker-Delta Function.

2.2 Variance in Cluster Size

The clusters, once made, must be controlled in size. In general, this means that the variance in size should be minimized. We would like an expression for the variance in size in order to determine a method of controlling the size.

Let us examine how this will evolve in terms of our previous theory. We have already seen that those clusters that have large densities tend to become smaller on average over time. This is because the swarm tends to try to equalize the density of pucks throughout the swarm, and so will move pucks from high density areas to areas with lower densities. Thus, if a cluster greater than the minimum point of the g function produces a higher density than that at the equilibrium point, then the clusters will tend to have the size of the minimum point.

First, suppose that N_1 is at the equilibrium point, and N_2 is above the equilibrium point. The density of the pucks around cluster of size N_1 is

$$\langle D\rangle_1 = D\int_0^{\infty} \frac{F(N)}{F(N)+H(N)}\rho_{N_1}(N)\,dN \qquad (13)$$

while that around the cluster of size N_2 is

$$\langle D\rangle_2 = D\int_0^{\infty} \frac{F(N)}{F(N)+H(N)}\rho_{N_2}(N)\,dN\ . \qquad (14)$$

Thus, the density around the clusters of size $N_1 < N_o$ must satisfy

$$D\int_0^{\infty} \frac{F(N)}{F(N)+H(N)}\left(\rho_{N_o}(N)-\rho_{N_1}(N)\right)dN > 0 \qquad (15)$$

with, suprisingly, the same condition no matter whether $N_1$is greater or less than N_2.

Again, if the agents have perfect information ($\rho = \delta$), this condition reduces to

$$\frac{F(N_o)}{H(N_o)} > \frac{F(N_1)}{H(N_1)} \qquad (16)$$

which is the condition found in [11].

3 Thresholding agent dynamics

A particularly simple agent one might imagine building is one which makes a decision to pick up or drop off a puck based on a simple threshold on the perceived cluster size. Such an agent is remarkably simple, and so can be robustly constructed in both robotic systems and virtual systems. The simplicity of the agent can, in some cases, be justified in terms of clustering systems because the interaction with the cluster provides the randomness required for the system. Several researchers have used such a deterministic approach in their design of clustering robot controllers [6][10][13]. Suppose, as an example, that a controller was designed using a simple thresholding behavior. This might be facilitated by having the following forms for f and h.

$$\begin{gathered} f = \begin{cases} 1 & \text{if } N < N_{min} \\ 0 & \text{if } N_{min} \le N \le N_{max} \\ 1 & \text{if } N_{max} < N \end{cases}, \\ h = \begin{cases} 0 & \text{if } N < N_{min} \\ 1 & \text{if } N_{min} \le N \le N_{max} \\ 0 & \text{if } N_{max} < N \end{cases} \end{gathered} \qquad (17)$$

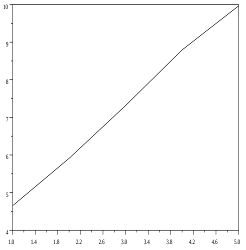

Figure 2: The variance of the cluster increases as N_{min}increases.

In this case, we have that in the case of the densities as given above in the discussion of Section 2, if the first cluster is at the equilibrium point, and not larger than the maximum size, and the robots are in the threshold model, the integral reduces to

$$\langle D\rangle_1 = D\int_0^{N_{min}} \rho_{N_1}(N)\,dN \qquad (18)$$

while the second becomes

$$\langle D\rangle_2 = D\left(\int_0^{N_{min}} \rho_{N_2}(N)\,dN + \int_{N_{max}}^{N_2} \rho_{N_2}(N)\,dN\right)\ . \qquad (19)$$

In general, the larger cluster will lose pucks to the smaller cluster if

$$\begin{gathered} \int_0^{N_{min}} \rho_{N_2}(N)\,dN + \int_{N_{max}}^{N_2} \rho_{N_2}(N)\,dN \\ > \int_0^{N_{min}} \rho_{N_1}(N)\,dN \end{gathered} \qquad (20)$$

assuming, of course, that the densities are equal. This condition will be satisfied by some forms of the probability of correct (and incorrect) characterization.

A strategy one might use to guarantee that the cluster size is controlled would be to reduce N_{min} to zero. In this case, the integral becomes

$$\int_{N_{max}}^{N_2} \rho_{N_2}(N)\,dN > 0. \qquad (21)$$

This will be satisfied no matter how small the increase in N_2 is over the desired size, making the variance minimal. The only way to reduce the variance further would then be to make more precise measurements of cluster sizes, which would in practice increase the likelihood of generating nontrivial structures in the clusters. This trade-off is an important part of practical agent design.

Put succinctly, this indicates that the greater that N_{min} is, the greater the variance, and reduction of N_{min} to zero removes an entire term from the variance calculation. This leads us to quite a different prediction than that given in [11] which indicates that a highly sloped g reduces variance. The discrepancy disappears when $\rho = \delta$ in the reaction g. In this model, however, the flatter we make g, the more controlled is the variance. Finally, there would seem to be a limit as to how much we can reduce the variance, and this limit is dependent only on the way the robot evaluates the cluster size. This is illustrated in Figure 2.

4 Embodied agent simulations

As we noted in Section 2, it is rarely the case that an agent-based system is designed in which the agents have perfect

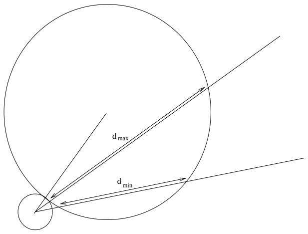

Figure 3: This figure gives a typical approach of a robot to the cluster. As the robot may approach from any direction, the perceived distance may range between size 0 and the diameter of the cluster. The minimal and maximal sizes are marked in the figure.

knowledge of the system of clusters with which they are working. It is more often the case that the system is imperfect in the sense that the agents have only estimates of the size of the clusters they are working with. Moreover, While it is often times possible, with significant effort, to gain very specific measurements of the clusters using estimate and integrate techniques[3] , it can be disadvantageous to utilize the exact size of the cluster, as the detailed cluster structure might become difficult to work around.

In these cases, the actual size of the cluster is either not wanted or impossible to obtain accurately. The behavior of the system becomes very different from that in the perfect knowledge case. In this Section, we observe several consequences of this, clearly examining differences in the behavior of the system. Our approach is both theoretical and experimental, and we illustrate the effects of these differences using a realistic simulation of a robotic system, designed with the intent to utilize sensory capabilities and behaviors that could realistically be implemented on a simple autonomous robot.

4.1 Embodied simulations and sensory accuracy

We use an embodied simulation which simulates a two-dimensional metaphysical world in which simulated agents have the freedom to move and interact with other objects. These agents are designed with great realism, with sensory, motion, and interaction capabilities strongly grounded in what is possible for real autonomous robots. For this reason, we refer to these agents as robots. The other objects in the simulation are inanimate objects known as pucks, which may be picked up, carried, and placed by robots. The robots are able to interact with objects in their environment, which may consist of pucks and walls. Walls serve as boundaries for robot interaction.

In our simulation, robots are endowed with sensors (these might be small cameras carried by real-world robot analogous to our simulated robots) which give visual information about the robot's surroundings. The camera is fixed and directed in the robot's direction of motion, oriented with the direction of motion passing through the center of the visual field. This view is limited both in width and length, though the length may be many times longer than the robot's size. This sensor allows the robot to discern whether or not there are robots in front of it, there are pucks in front of it, and approximately how far away each item in the visual field is from the robot. Using this information, the robot can calculate the diameter of the part of the cluster in its visual field, and estimate its size.

As a practical matter, this always underestimates the cluster's size in number of pucks when the robot approaches the cluster from an angle not perpendicular to the cluster's edge (assuming the cluster is circular). This method is hereafter known as the *delta estimation method (DEM)* as it assumes that the width of the robot's view is as small as possible. We may examine the average effect of the viewing strategy on the size of the cluster experienced by the robot.

One important quality for later work in this area is that the DEM is useful in helping the cluster stay circular. This may be understood in the following way. Any deviation from circular structure forces the robots to regard the cluster as a smaller cluster when approaching from multiple directions. If the natural tendency is to add to smaller clusters, the clusters are then padded from the sides. If the natural tendency is to remove pucks from small clusters, these cluster irregularities will be reduced in size until they have been removed entirely. Thus, the cluster will stay circular.

4.2 Agent perception of cluster size in the embodied simulation

In what follows, we assume that the probability of approaching the cluster from any given angle is equal. Let L be the length of a chord through the circle along the robot's direction of motion. Then let α be the angle of approach, and

$$\alpha = d(L)\ . \tag{22}$$

In our model, this indicates that

$$d\,(L) = sin^{-1}\left(\frac{R}{R+r} sin\left(cos^{-1}\left(\frac{L}{2R}\right)\right)\right) \tag{23}$$

If $\rho\,(\alpha)$ is the probability of the robot approaching from any angle, then it is the case that

$$1 = \int_{\alpha_{min}}^{\alpha_{max}} \rho\,(\alpha)\,d\alpha = \int_{\alpha_{min}}^{\alpha_{max}} \frac{1}{\Delta\alpha} d\alpha \tag{24}$$

where $\Delta\alpha = \alpha_{max} - \alpha_{min}$. Now suppose that

$$p = \frac{L}{2R} \tag{25}$$

where R is the radius of the cluster. This defines a new function of p. Thus,

$$D\,(p) = sin^{-1}\left(\frac{R}{R+r} sin\left(cos^{-1}\,(p)\right)\right) \tag{26}$$

where p runs from zero to one. Substitution and rearrangement gives us

$$1 = \int_{\alpha_{min}}^{\alpha_{max}} \frac{1}{\Delta\alpha} d\alpha = -\int_0^1 (\frac{1}{\Delta\alpha}\frac{dD}{dp})dp \tag{27}$$

Therefore,

$$\rho(p) = -\frac{1}{\Delta\alpha}\frac{dD}{dp}. \tag{28}$$

ρ is thus given by

$$\rho\,(p) = \frac{Rp}{\Delta\alpha\,(R+r)\sqrt{(1-p^2) - \frac{R^2\left(1-p^2\right)^2}{(R+r)^2}}}\ . \tag{29}$$

As an example, if we assume the radius of a cluster is 1 and the radius of a robot is 0.1 then

$$\rho\,(p) = \frac{-p}{1.1 sin^{-1}\left(\frac{1}{1.1}\right)\sqrt{(1-p^2) - \frac{\left(1-p^2\right)^2}{1.1^2}}}\ . \tag{30}$$

---

[3] This might be possible if the robot, for instance, swept its sensor across the cluster and integrated the size of the cluster as it swept across it.

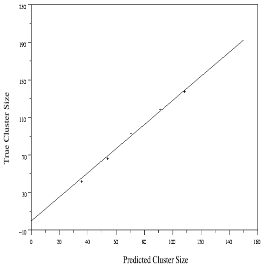

Figure 4: When using our metaphysical simulation, the calibration curve is as given. This curve is "nearly linear" and deviates very slightly from linear because the ratio of robot and cluster sizes changes over this range. The least squares regression line of this function is defined as $y = -0.16997 + 1.284195x$.

If $R \gg r$, this reduces to

$$\rho(p) = \frac{1}{\Delta\alpha\sqrt{1-p^2}} \ . \tag{31}$$

With this probability,

$$\Delta\alpha = \frac{\pi}{2}$$

$$\langle p \rangle = \frac{2}{\pi}\int_0^1 \frac{p}{\sqrt{1-p^2}}dp = -\frac{2}{\pi}\sqrt{1-p^2}|_0^1 = \frac{2}{\pi} \tag{32}$$

which gives

$$\langle L \rangle = 1.2732395R \ . \tag{33}$$

This last equation gives us the probability of the robot thinking that a cluster is a particular ratio of the cluster size.

Moreover, for clusters whose radius is much greater than the robot, this probability is nearly constant. This indicates that the size estimated by the robot should be linearly dependent on the size of the cluster, increasing linearly with cluster size.

Figure 4 illustrates the increase in observed cluster size as a function of the actual cluster size. The least squares regression line of these data is $L = -0.16997 + 1.284195R$, in good agreement with the expected value of the average ratio[4].

4.3 Metaphysical agent clustering

In swarm engineering, our primary goal is to determine minimal conditions required to obtain a global outcome and then to build agent systems exhibiting these minimal conditions. In equation (48) we have the general clustering condition. We now build a virtual system and demonstrate the conditions under which this system satisfies the generalized clustering condition. In Section 4.3, we extend this investigation to provide predictions about the cluster variance, and contrast the predictions of this system with those of the previous system.

Our agents are meant to be extremely simple. This is to facilitate the generation of real agents in the laboratory capable of doing what our simulated agents are. With this in mind, we design our agent controllers very simply.

1. The agent will move around according to simple random or semi-random behaviors.
    (a) In a bounded arena, agents move in straight lines until encountering wall or another agents, at which point the agent executes a random turn away from the object, and continues moving.
    (b) In an unbounded arena, the agent is given an unlimited arena to move in, and thus a spiraling behavior is implemented. When the agent is unable to see any pucks, it begins to spiral in a random direction, thus allowing the agent to return to the clusters.
2. When the agent approaches a cluster, it generates an estimate of the size of the cluster as a result of its approach angle and position.
3. If the cluster is classified small, the agent will deposit a carried object with probability p or pick up an object from the cluster with a probability $1 - p$.
4. If the estimate is that the cluster size is above a minimal threshold and below a maximal threshold, the agent will deposit a carried object.
5. If the estimate is that the cluster size is above a maximal threshold, the agent will pick up an encountered object. However in generating single clusters, the maximum threshold is disregarded.

In our simulations, the agent to agent interactions as well as agent to puck interactions are simulated with as much realism as possible. As previously described, agents travel in straight lines at a given speed until they encounter other objects. Based on the object it encounters, the agent can choose one of these courses of action: dropping or picking up a puck, or making a random turn away from the object. When a agent encounters a puck, it assumes that the puck is part of a cluster, and then estimates the size of the cluster. Based on its interpretation of cluster size, the agent can either drop or pick up a puck, or simply ignore the cluster. When a agent encounters another agent, both agents execute a random turn away from each other. When an agent encounters a wall in the bounded simulation, the agent executes a random turn away from the wall.

When the boundaries are removed the agents are allowed significant freedom of movement, which makes it possible for the agent to wander away from any pucks or agents. In order to avoid the loss of agents due to wandering, we implement a spiraling behavior that causes the agents to begin spiraling in a random direction once the agent is unable to see any pucks. This eventually causes the agent to return to the pucks if they are temporarily lost. Other than the spiraling effect, the other agent movement behaviors are the same for both embodied simulations.

The realistic nature of the embodied simulation allows clusters to dynamically form and break up. Clusters are not predefined, with a specific number of clusters which is preserved throughout the simulation. Rather, clusters are allowed to be form and to be broken up under the influence of agent actions. This yields significantly more complex behavior as the clusters then can be much more difficult to control. We find empirically, that control of clusters with a minimal internal structure is significantly advantageous. Clusters made up of diffuse puck strands can be easily broken if the strand is approached by a agent which correctly assigns it a small cluster designation.

4.3.1 Single cluster development

Of course, our first goal is to understand how single clusters can be formed in this model of clustering. We are armed now with our understanding of the size estimation capability of the swarm and with the general clustering condition. Let us consider again the agent approaching a cluster (Figure 3). We may determine the minimal approach angle resulting in the agent's placing a puck in the cluster as a function of the agent size r, the minimal distance d_{min}, and the cluster size

---

[4] It is interesting to note that this represents the first *quantitative* prediction of behavior in the puck clustering literature.

R. We note that the maximal angle between the direction of agent motion and the connector between diameters is formed when

$$\Theta_{max} = sin^{-1} \frac{R}{R+r} \ . \tag{34}$$

Considering the probabilistic nature of these agent behaviors, we find that any uncertainty in the behavior results from the uncertainty that agents will approach clusters from any given direction. This uncertainty creates the behavioral probabilities that ultimately drive the clustering behaviors. This helps determine the probability of entering any behavior.

When approaching a cluster the angle at which the agent perceives the cluster at its minimal size for dropping the object off is given by

$$\alpha_{min} = sin^{-1}\left(\frac{R}{R+r} sin\left(cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right) . \tag{35}$$

This may be verified geometrically. Assuming that the maximal distance d_{max} is infinite, this puts the probability of dropping off the object at

$$p_d = \frac{\alpha}{\theta_{max}} = \frac{sin^{-1}\left(\frac{R}{R+r} sin\left(cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right)}{sin^{-1}\frac{R}{R+r}} . \tag{36}$$

This also puts the probability of pickup at

$$p_u = 1 - \frac{\alpha}{\theta_{max}} = 1 - \frac{sin^{-1}\left(\frac{R}{R+r} sin\left(cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right)}{sin^{-1}\frac{R}{R+r}} . \tag{37}$$

Together, these produce a g equal to

$$g = \frac{p_u}{p_d} = \frac{sin^{-1}\frac{R}{R+r}}{sin^{-1}\left(\frac{R}{R+r} sin\left(cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right)} - 1 \tag{38}$$

This g function has is monotonically decreasing, and satisfies at least part of the clustering condition. Clearly equation (?) cannot be satisfied if g is an increasing function of the cluster size R. This function is a decreasing function of R, which indicates that it is also a decreasing function of N. Using this behavior, the agent must perform a binary decision each time it encounters a cluster. This decision determines whether or not the agent considers the cluster large. If the agent considers the cluster to be large, then it will not pick up pucks, but will drop off pucks. On the other hand, if it considers the cluster small, it will only pick up pucks.

Let us now examine the probability ρ in order to understand how we may predict the outcome of the system. Let us suppose that we have a regular form of ρ. That is, suppose that $\frac{d^2\rho}{dN^2} < 0$ for all N. It is easy to show that this requires that there be at most one peak in the function. This means that there must exist a point of intersection between any two different regular probabilities ρ_{N_1} and ρ_{N_2}. We can see this by noting that if $\rho_{N_1}(0) < \rho_{N_2}(0)$, since $\int_0^1 \rho_{N_1} = \int_0^1 \rho_{N_2} = 1$ either $\rho_{N_1}(N) = \rho_{N_2}(N)$ or there exists a point a such that $\rho_{N_1}(a) < \rho_{N_2}(a)$ and $\rho_{N_1}(b) > \rho_{N_2}(b)\ \forall b > a$. The Intermediate Value Theorem provides the rest of the proof. Thus, we may rewrite the generalized clustering condition, we have

$$\begin{aligned} &\textstyle\int_0^a \left(\rho_{N_2}(N) - \rho_{N_1}(N)\right) \frac{g(N)}{g(N)+1} dN \\ > &\textstyle\int_a^\infty \left(\rho_{N_1}(N) - \rho_{N_2}(N)\right) \frac{g(N)}{g(N)+1} dN \end{aligned} \tag{39}$$

In our simulations, we utilize agents whose actions are based on a threshold in the cluster size perceived, as explained above. The functions F and H are as given in (11). For $N < N_{min}$, $\frac{F(N)}{F(N)+H(N)} = 1$ while for $N \geq N_{min}$ $\frac{F(N)}{F(N)+H(N)} = 0$. This allows us to simplify (14) to

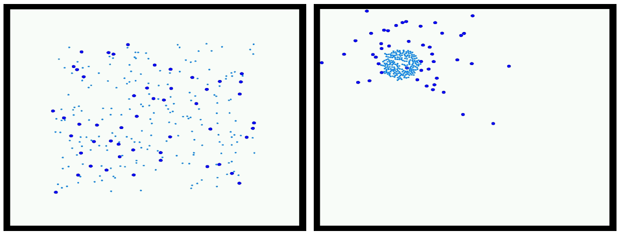

Figure 5: This simulation is unbounded and initialized with arbitrarily scattered pucks. Clusters form, one of which is considerably more dominant than the other. As in the non-embodied simulation, we find that the dominant cluster absorbs all the pucks.

$$\langle D \rangle = D \int_0^{N_{min}} \rho_{N_o}(N)\, dN \tag{40}$$

and (15) to

$$\int_0^{N_{min}} \left(\rho_{N_2}(N) - \rho_{N_1}(N)\right) dN > 0 \tag{41}$$

providing that $a > N_{min}$. If $N_{min} > a$ then (15) becomes

$$\left(\int_0^a + \int_a^{N_{min}}\right) \left(\rho_{N_2}(N) - \rho_{N_1}(N)\right) > 0 dN \ . \tag{42}$$

In the case that $R \gg r$ the probability is as given in equation (31). Moreover, the probability of a given value for p is equal to the probability of a given value for N where $N = \kappa\pi\,(pR)^2$. This allows us to rewrite equation (42) as

$$\int_0^{N_{min}} \left(\frac{1}{\sqrt{2N_2N - 2N^2}} - \frac{1}{\sqrt{2N_1N - 2N^2}}\right) dN > 0 \tag{43}$$

$$2\sqrt{2}\left(\int_{-1}^{\frac{2N_{min}}{N_2}-1} - \int_{-1}^{\frac{2N_{min}}{N_1}}\right) \frac{1}{\sqrt{1-q^2}} dq > 0 \tag{44}$$

$$sin^{-1}\left(1 - \frac{2N_{min}}{N_1}\right) > sin^{-1}\left(1 - \frac{2N_{min}}{N_2}\right) \tag{45}$$

Note that since arcsine is a positive increasing function, this will always be true when N_1 is greater than N_2 as long as N_1 is large enough that random fluctuations do not reverse this condition. Thus, in our system, the simple behavior will lead to a single cluster. In practice, this is precisely what is found, as can be seen in Figure 5.

In this run the simulation is initialized with the pucks arbitrarily scattered. As expected, all the clusters eventually merge to form a single cluster.

4.3.2 Multi cluster formation

The behavior for generating multiple clusters results from the same set of agent behaviors for the system. However, the upper limit is placed at some positive number. This results in a significant change in robot behavior. In this case, the probability of drop off is given approximately by

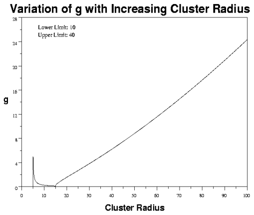

Figure 6: This gives the behavior of g as a function of the cluster radius. In this case, the function has a minimum, and increases monotonically after this minimum. This minimum represents the equilibrium size for the cluster that results from this behavior. d_{max} is 30 and d_{min} is 90. The minimum is at 45.

$$p_d = \begin{cases} \frac{\alpha}{\theta_{max}} & \text{if } \frac{d_{max}}{2} > R > \frac{d_{min}}{2} \\ \frac{\alpha-\beta}{\theta_{max}} & \text{if } \frac{d_{max}}{2} \leq R \end{cases} \tag{46}$$

where β is analogous to α for d_{max} rather than d_{min}.

$$p_u = \begin{cases} 1-\frac{\alpha}{\theta_{max}} & \text{if } \frac{d_{max}}{2} > R > \frac{d_{min}}{2} \\ 1-\frac{\alpha-\beta}{\theta_{max}} & \text{if } \frac{d_{max}}{2} \leq R \end{cases} \tag{47}$$

This ratio of $p_u$to p_d clearly does not exist at $R < \frac{d_{min}}{2}$. At such sizes, the cluster has not formed well enough for this examination to be applicable. As discussed above, we employ a pseudo-clustering approach to meet the smallest size criteria. The function has a minimal point at $\frac{d_{max}}{2}$after which point it begins increasing. This of course means that the robot will start tearing apart clusters larger than the large threshold. This is both as expected and desired.

The behavior above yields a V-shaped function. In [9], this function was associated with the development of multiple clusters of differing sizes or potentially single clusters, depending on the initial sizes of the clusters. This had the unhappy side effect of providing more than one potential outcome for the system depending on the starting point. That work assumed that the probability was a delta function, which is not the case here.

Let us examine how this will evolve in terms of our previous theory. We have already seen that those clusters that have large densities tend to become smaller on average over time. This is because the swarm tends to try to equalize the density of pucks throughout the swarm, and so will move pucks from high density areas to areas with lower densities. Thus, if a cluster greater than the minimum point of the g function produces a higher density than that at the equilibrium point, then the clusters will tend to have the size of the minimum point.

First, suppose that N_1 is at the equilibrium point, and N_2 is above the equilibrium point. The density of the pucks around cluster of size N_1 is

$$\langle D \rangle_1 = D \int_0^\infty \frac{F(N)}{F(N)+H(N)} \rho_{N_1}(N)\, dN \tag{48}$$

while that around the cluster of size N_2 is

$$\langle D \rangle_2 = D \int_0^\infty \frac{F(N)}{F(N)+H(N)} \rho_{N_2}(N)\, dN\ . \tag{49}$$

Since the first cluster is at the equilibrium point, and not larger than the maximum size, and the robots are in the threshold model, the integral reduces to

$$\langle D \rangle_1 = D \int_0^{N_{min}} \rho_{N_1}(N)\, dN \tag{50}$$

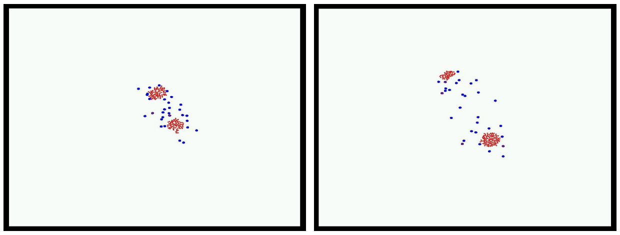

Figure 7: This simulation shows a run in which two clusters of equal sizes are formed. Both the net change of each of the cluster is balanced, so no cluster maintains a clear dominance over the other.

while the second becomes

$$\langle D \rangle_2 = D \left(\int_0^{N_{min}} \rho_{N_2}(N)\, dN + \int_{N_{max}}^{N_2} \rho_{N_2}(N)\, dN \right)\ . \tag{51}$$

In general, the larger cluster will lose pucks to the smaller cluster if, after rearrangement,

$$\rho_{N_2}(N)\, dN > \int_0^{N_{min}} \left(\rho_{N_1}(N) - \rho_2(N) \right) dN \tag{52}$$

assuming, of course, that the densities are equal. This condition will be satisfied by some forms of the probability of correct (and incorrect) characterization, it is clear that others will not be able to satisfy this in general or precisely at the equilibrium point. In general, the integrand on the right is expected to be positive, but there is no reason to believe that this must be the case. In actuality, N_2 must be large enough so that the integral on the left side is larger than that on the right, and the loss of this condition can cause an increase in the size of the cluster, which in turn could cause a significant variance in the cluster sizes, depending on how sharply peaked the integral is.

A strategy one might use to guarantee that the cluster size is controlled would be to reduce N_{min} to zero. In this case, the integral becomes

$$\int_{N_{max}}^{N_2} \rho_{N_2}(N)\, dN > 0. \tag{53}$$

This will be satisfied no matter how small the increase in N_2 is over the desired size, making the variance minimal. The only way to reduce the variance further would then be to make more precise measurements of cluster sizes, which would in practice increase the likelihood of generating nontrivial structures in the clusters. This trade-off is an important part of practical robot design.

In Figure 7, we illustrate this process by creating two clusters of equal size. Initially, the cluster material is randomly placed, and it begins forming stable clusters. However, the size of these clusters is limited, and does not exceed half of the pucks. The clusters are formed using a minimal size of zero.

5 Discussion and conclusions

The importance of this work becomes apparent when analyzing the dynamics of the clustering systems. We had previously thought that there was only one distinct type of puck clustering, as defined by [2]. Using a basic g function, they were able to generate results that were deemed similar in both the embodied and the non-embodied simulation. However, upon further testing, we discovered that there are in fact two distinct approaches to puck clustering: one which employs the reaction g, and one which employs the interaction

g. This paper has explored the dynamics of the interaction g, deriving different dynamic properties than those previously determined.

We are but in the first steps towards swarm based distributed construction. We expect to be able to generate controlled two dimensional shapes of multi-cluster formations, and then three dimensional shapes. Moreover, pucks can be seen as arbitrary building materials that the agents manipulate based on a certain set of behaviors. Most likely in the real world, agents would be implemented in the form of robots, and pucks in the form of bricks or other building material. The behavior of the agent would be determined by the task that needs to be accomplished, which would be different depending on whether the reaction or interaction g is employed. Our research contributes a basic fundamental theory that clearly shows two methods of puck clustering in order to determine the behavior of agents in a puck clustering system.

References

[1] W. Cohen. *Adaptive mapping and navigation by teams of simple robots.* **Robotics and Autonomous Systems**. 18: 411-434, 1996.

[2] A. Zhang. et. al. *Variance in Converging Puck Cluster Sizes.* **First International Joint Conferencee on Autonomous Agents and Multi-Agent Systems**. Bologna, Italy, July 2002.

[3] M. Mataric. *Designing and understanding adaptive group behavior.* **Adaptive Behavior**, 4 (1): 51-80, 1995.

[4] M. Mataric. *Designing emergent behaviors: from local interactions to collective intelligence.* In **Proceedings, From Animals to Animats 2, Second International Conference on Simulation of Adaptive Behavior (SAB-92)**. J-A. Meyer, H. Roitblat and S. Wilson, eds., MIT Press, 432-441, 1992.

[5] W. Agassounon, A. Martinoli, and R. Goodman. *A Scalable, Distributed Algorithm for Allocating Workers in Embedded Systems* **Proceedings of the 2001 IEEE Systems, Man and Cybernetics Conference**. Tucson, Arizona, USA, October 2001.

[6] R. Beckers, O. Holland, J. and Deneubourg. *From* R. Beckers, O. Holland, and J. Deneubourg, eds. *From local actions to global tasks: Stigmergy and collective robotics.* **Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems**, MIT Press, 1994.

[7] E. Bonabeau, G. Theraulaz, V. Fourcassie, and J. Deneubourg. *Phase-ordering kinetics of cemetary organization in ants.* **Physical Review E**, 57 (4), 1998.

[8] N. Franks, A. Wilby, B. Silverman, and C. Tofts. *Self-organizing nest construction in ants: sophisticated building by blind bulldozing.* **Animal Behavior**, 44, 357-375, 1992.

[9] S. Kazadi, A. Abdul-Khaliq, and R. Goodman. *On the convergence of puck clustering systems.* **Robotics and Autonomous Systems**, 38 (2), 93-117, 2002.

[10] S. Kazadi. **Swarm Engineering**. PhD Thesis, California Institute of Technology, 2000.

[11] M. Maris and R. Boekhorst. *Exploiting physical constraints: heap formation through behavioral error in a group of robots.* **IROS '96 IEEE/RSJ International Conference on Intelligent Robots and Systems**, 1996.

[12] A. Martinoli, A. Ijspeert, and F. Mondada. *Understanding collective aggregation mechanisms: from probabilistic modeling to experiments with real robots.* **Robotics and Autonomous Systems**, 29: 51-63, 1999.

[13] C. Melhuish and O. Holland. *Getting the most from the least: lessons for the nanoscale from minimal mobile agents.* **Proceedings of Artificial Life V**. C. Langton, K. Shimorhara, eds. MIT Press: Cambridge MA, 1996.

[14] N. Meuleau and M. Dorigo. *ACO and Stochastic Gradient Descent.* **Artificial Life**. 8(2), 103-121, 2002.