

On the Dynamics of Clustering Systems

S. Kazadi*, M. Chung, B. Lee, R. Cho

24th September 2003

Jisan Research Institute; 28 North Oak Avenue, Pasadena, CA 91107

Abstract

We examine the theoretical foundations for the dynamics of puck clustering systems. Key in this investigation is the development of methods of controlling variance in cluster size, an important precursor to swarm-mediated clustering. We derive conditions under which clustering can take place in a general framework, and demonstrate two different behavioral regimes for clustering systems.

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 development 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.

Swarm studies have been increasing in number due to the astounding capability of groups of relatively simple agents. These studies have demonstrated that it is possible to accomplish high-level tasks with simple agents, even when the agents are extremely limited. Many of these studies focused on the investigation of groups of agents which demonstrated a high-level intelligence, while others focused on the development of systems capable of accomplishing engineering tasks. Thus, the studies of swarms branched off into two distinct fields: swarm intelligence, and swarm engineering. Typical swarm intelligence applications have included path-finding [1] and stick-pulling [14] to while swarm engineering has been restricted to clustering studies [2]. The potential in swarm-based applications to these tasks are enormous; interest derives from the possibility of accomplishing complex tasks in remote, hazardous, or undesirable environments economically using swarms of simple agents.

While much work was initially focused on the demonstration of the ability to do interesting things with systems developed with swarm engineering techniques [3], current work is more focused on developing methods of generating pre-specified outcomes by understanding the dynamics of the agents and agent-based systems designed by swarm engineering. This type of analysis is in its infancy, and may be found in some

*To whom correspondences should be addressed at: sanza@jisan.org.

of the convergence proofs of Marco Dorigo and collaborators [14] in working with ant-based systems along with their kinetics shown by Bonabeau, Frank, and collaborators [7][8], and in the work of Kazadi and collaborators in generating clustering predictions [2][9][10].

Most of the studies that have been undertaken thus far have lacked any theoretical basis outside of descriptive theory which tended to describe those properties of swarm systems already familiar to the researcher. This is unsatisfying because the studies themselves do not then represent more than anecdotal studies; it is difficult or impossible (without significant work) to generalize the work to other similar systems. However, in designing swarms it is advantageous to possess a fundamental and theoretical understanding of their design and operation. Thus, we wish to ground all of our design paradigms in a firm theoretical framework, a methodology that is missing in much of the swarm literature.

This paper centers around the development of theory and methodology surrounding clustering. *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]. Many 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.

In the current work, our interest lies in the understanding of differing types of clustering systems. We shall explore two differing systems and their differing dynamic properties. Two fundamental classes of systems are those in which randomness is part of the agent's behaviors and those in which randomness is part of the agent's sensory capability. These represent two very different classes of agents. Those in the first class are provided with perfect knowledge (with some degree of noise) about the system on which they are working. It has already been shown that the randomness in agent behavior drives clustering systems, and so such a system would require randomness to be embedded within the agent's behaviors. On the other hand, those systems with randomness inherent in the sensory capability of the robot would not require a random element to be embedded in the behavior of the agent, as it would be a natural consequence of any such system, even when using agents with deterministic behaviors. Thus, the driving force for clustering need not be the agent's behaviors, but rather the nature of the system, which simplifies the agent's design. However, as we shall see, this causes important differences in the dynamics of the systems.

We investigate these two different systems by examining the variance of the cluster sizes in robotic systems which generate clusters of predetermined size. We demonstrate that the effect of changes in the functional forms and ratios of f and h have different functional effects on the size of the variance in the cluster sizes. This indicates a dichotomy in the generation of predictive calculations which would seem to affect all clustering simulations. Of great import, then, is the understanding of how the probability of an agent involved in the clustering deciding incorrectly about the size of a cluster will affect the overall outcome

of the clustering system and its control. While the variance of the cluster size is our specific investigatory tool, we expect these types of considerations to generalize quite easily to other properties.

The remainder of the paper will be organized as follows. Section 2 theoretically investigates the different system properties, making appropriate definitions and theoretical predictions. Section 3 presents the experimental behavior of the behaviorally generated random systems. Section 4 gives experimental behavior of the system design-based random systems. Section 5 explores the determination of the completion of the clustering task, and its detection by the swarm. Section 6 provides a discussion of the implications of these systems and concluding remarks.

2 Reaction g vs. interaction g

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 a single cluster.

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 2.1.

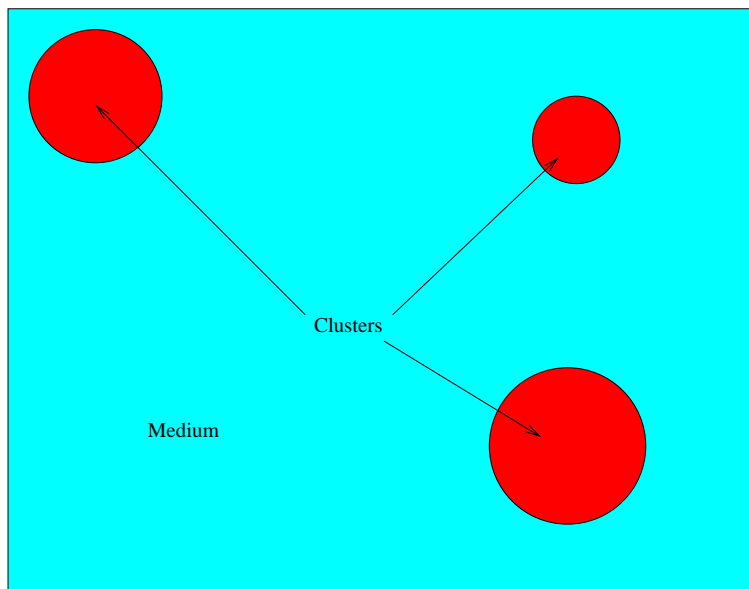


Figure 2.1: This figure represents the systemic view of clustering systems.

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.

The conditions for a single stable final state containing one cluster has already been determined in previous studies [9][10]. This condition, which has been derived in two different ways, consistent with artificial and real systems, is that

$$g(N) < g(N') \quad (1)$$

whenever $N > N'$, where g is defined by

$$g(N) = \frac{f(N)}{h(N)} \quad (2)$$

and $f(N)$ represents the likelihood of a puck removal from a cluster of size N and $h(N)$ represents the likelihood of puck deposition on a cluster of size N . This condition, as we shall see in Section 3, can be experimentally verified.

Though this theoretical framework works for many applications, it has at least two tacit assumptions: the functions f and h must be differentiable functions of N , and N must be well known. However, these assumptions are not always true. All agents' abilities to sense clusters are limited by whatever apparatus they use to identify and categorize clusters. In a real system, if the robot is small in comparison to the cluster being built, then the agent can only estimate the size of N . Thus, N represents a random variable whose mean is not necessarily the size of the cluster. Since the agent may perceive the cluster as having a size different from the actual size of the cluster, the behavior of the system can be radically different from the expected or desired behavior. In fact, it is virtually impossible to imagine building a physical robot whose perception of the cluster is so specific that it always returns the correct cluster size. As a result, in general, the number of pucks perceived by the robot N is some random variable which is parametrized by the actual number of pucks in the cluster. We can say, then, that N is more correctly thought of as N_{N_o} where the subscript N_o is added to emphasize that the number is conditioned by the number of pucks actually in the cluster. Thus, we have

$$g = \frac{f(N_{N_o})}{h(N_{N_o})} \quad (3)$$

and

$$g' = \frac{f'(N_{N_o})h(N_{N_o}) - f(N_{N_o})h'(N_{N_o})}{[h(N_{N_o})]^2} \quad (4)$$

Note that this expression only exists, and the subsequent analysis makes sense, if the functions $f(N_{N_o})$ and $h(N_{N_o})$ are well defined and differentiable. Let us now define

$$\langle f(N_{N_o}) \rangle = \int_0^\infty f(N) \rho_{N_o}(N) dN \quad (5)$$

where $\rho_{N_o}(N)$ is the probability that the agent will perceive a cluster of actual size N_o as having the size N . This is well defined for any given value of N_o , as is the analogous value for $\langle h(N_{N_o}) \rangle$.

Perfect knowledge of the cluster size implies that the probability of making a mistake is zero, though the integral of the total probability is unity. This implies that the probability would be given by

$$\rho_{N_o}(N) = \delta(N - N_o) \quad (6)$$

¹ This, in turn, recovers our previous expression for g . I.e.

$$\langle g \rangle = \left\langle \frac{f(N_{N_o})}{h(N_{N_o})} \right\rangle = \int_0^\infty \frac{f(N)}{h(N)} \delta(N - N_o) dN = \frac{f(N_o)}{h(N_o)} \quad (7)$$

¹ δ is the Kronecker-Delta Function.

We may give a specific label to the average value of g as follows.

$$G \equiv \langle g \rangle = \left\langle \frac{f(N_{N_o})}{h(N_{N_o})} \right\rangle. \quad (8)$$

which in the perfect knowledge case may be reduced to

$$G \equiv \langle g \rangle = \frac{f(N_{N_o})}{h(N_{N_o})} = g(N_o). \quad (9)$$

This last expression was used in previous studies, and we now see that it is valid only for cases in which the actual number of pucks in the cluster is very close to the expected number of pucks². Now we can see that in this case

$$G' \equiv \langle g' \rangle = \int_0^\infty g'(N_{N_o}) \delta(N - N_o) = g'(N_o) \quad (10)$$

One consequence of the randomness of the perceived size is that the randomness need not be built into the agent's controller. In this case, the control strategy of the agent can be completely deterministic. 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 .

$$f = \begin{cases} 1 & \text{if } N < N_{min} \\ 0 & \text{if } N_{min} \leq N \leq N_{max} \\ 1 & \text{if } N_{max} < N \end{cases}, \quad h = \begin{cases} 0 & \text{if } N < N_{min} \\ 1 & \text{if } N_{min} \leq N \leq N_{max} \\ 0 & \text{if } N_{max} < N \end{cases} \quad (11)$$

These functions do not have differentials at $N = N_{min}$ and $N = N_{max}$, and so the expression in equation (11) cannot be used. We may remedy this by using a similar form for the functions.

$$f = \begin{cases} \frac{1}{2} \frac{e^{-k(N-N_{min})} - e^{-k(N_{min}+N_{max})}}{e^{-k(N-N_{min})} + e^{-k(N_{min}+N_{max})}} + \frac{1}{2} & \text{if } N < \frac{N_{min}+N_{max}}{2} \\ \frac{1}{2} \frac{e^{-k(N-N_{max})} - e^{-k(N_{min}+N_{max})}}{e^{-k(N-N_{max})} + e^{-k(N_{min}+N_{max})}} + \frac{1}{2} & \text{if } \frac{N_{min}+N_{max}}{2} \leq N \end{cases}$$

$$h = \begin{cases} \frac{1}{2} \frac{e^{-k(N-N_{min})} - e^{-k(N_{min}+N_{max})}}{e^{-k(N-N_{min})} + e^{-k(N_{min}+N_{max})}} + \frac{1}{2} & \text{if } N < \frac{N_{min}+N_{max}}{2} \\ \frac{1}{2} \frac{e^{-k(N-N_{max})} - e^{-k(N_{min}+N_{max})}}{e^{-k(N-N_{max})} + e^{-k(N_{min}+N_{max})}} + \frac{1}{2} & \text{if } \frac{N_{min}+N_{max}}{2} \leq N \end{cases} \quad (12)$$

These functions have derivatives given by

$$f' = \begin{cases} \frac{-ke^{2k(N-N_{min})}}{(e^{2k(N-N_{min})}+1)^2} & \text{if } N < \frac{N_{min}+N_{max}}{2} \\ \frac{ke^{2k(N-N_{max})}}{(e^{2k(N-N_{max})}+1)^2} & \text{if } \frac{N_{min}+N_{max}}{2} \leq N \end{cases}, \quad h' = \begin{cases} \frac{ke^{2k(N-N_{min})}}{(e^{2k(N-N_{min})}+1)^2} & \text{if } N < \frac{N_{min}+N_{max}}{2} \\ \frac{-ke^{2k(N-N_{max})}}{(e^{2k(N-N_{max})}+1)^2} & \text{if } \frac{N_{min}+N_{max}}{2} \leq N \end{cases} \quad (13)$$

In the limit that $k \rightarrow \infty$, these derivatives will become infinite, as desired, at $N = N_{max}$. They are zero everywhere else. G' is given by

$$G' = \left\langle \lim_{k \rightarrow \infty} \frac{f'_k(N) h(N) - h'_k(N) f(N)}{h_k^2(N)} \right\rangle \quad (14)$$

Since $f' = -h'$ and $f + h = 1$, this further reduces to

$$G' = \left\langle \lim_{k \rightarrow \infty} -\frac{h'_k(N)}{h_k^2(N)} \right\rangle \quad (15)$$

²A similar set of results would occur if $\rho(N) = \delta(N - \alpha N_o)$.

which finally reduces to

$$G' = \lim_{k \rightarrow 0} \int_0^{N_{min}} \frac{-h'_k(N')}{h_k^2(N')} \rho_{N_o}(N') dN' + \lim_{k \rightarrow 0} \int_{N_{min}}^{N_{max}} \frac{-h'_k(N')}{h_k^2(N')} \rho_{N_o}(N') dN' + \lim_{k \rightarrow 0} \int_{N_{max}}^{\infty} \frac{-h'_k(N')}{h_k^2(N')} \rho_{N_o}(N') dN' \quad (16)$$

$$G' = \rho_{N_o}(N_{min}) - \rho_{N_o}(0) + \rho_{N_o}(N_{max}) - \rho_{N_o}(\infty) \quad (17)$$

The dynamics of such a system are certain to be different from those of a system with randomness built into the behavior.

In further discussions, we wish to differentiate between these two different forms of G . In what follows, we identify the G of the type given in equation (9) as that whose randomness in the behaviors of the agents are a result of the reaction of the agent to the information provided by the sensors. This is hereafter known as a **reaction G** or a **reaction g** whenever the meaning is clear. Other systems, whose randomness is a result of the randomness in the determination of N is hereafter referred to as an **interaction G** or an **interaction g** whenever the meaning is clear. In the following pages, we will consider the clustering dynamics of these two different systems.

3 Convergence and variance in the reaction g

In this Section, we explore puck clustering systems that employ the reaction g by using a non-embodied simulation. A *non-embodied simulation* is one in which no attempt is made to model any reasonable physics. Experiments are primarily numerical, with an emphasis on generating results meant to test a theoretical understanding of the underlying dynamics. Experiments involving the reaction g are centered around the investigation of swarm dynamics using little or no realistic physical knowledge in the exploration of these dynamics.

3.1 Simulation

We utilize a non-embodied simulation which utilizes a fixed number of agents and clusters. Agents either carry pucks or are empty, while clusters have no more structure than a number indicating how many pucks are contained within. Initially, the agents do not carry pucks and each cluster has a specific number of pucks. The agents drive the evolution of the system by changing the number of pucks in each cluster. In each cycle, each agent randomly chooses one cluster to manipulate. Each manipulation consists of either the removal of a puck or the deposition of a puck. Transport time of pucks between clusters and agents is considered instantaneous, though only using two or more successive iterations can an agent pick up and then drop off a puck. This limits the speed at which clusters can be built. The cluster sizes can increase and decrease based on agent behaviors, but once a cluster size of zero is achieved, the cluster is considered non-existent and cannot be further manipulated.

Manipulations of the clusters by the agents are conditioned on random probabilities which are themselves conditioned on the sizes of clusters. The probabilities that determine whether or not an agent will manipulate a cluster are based on two functions, as indicated in Section 2. $f(N)$ represents the probability that an agent not carrying a puck will pick up a puck from a cluster of size N , and the function $h(N)$ denotes the probability that an agent currently carrying a puck will drop it off into a cluster of size N . The ratio of these functions, $\frac{f}{h}$ is the functional g . As shown previously [9][10], we observe that the behavior of the g function can be used to predict the behavior of the system.

The main benefit of this simulation is speed and efficiency. Runs can be completed generally in under a minute on a Pentium II class personal computer. This allows for quick verification of theory and is useful in generating general simulation data.

3.2 g and the generation of clusters

As we stated in the previous Section, the g functional can be used to predict the long term evolution of the clustering system. This is useful because this allows us to generate behaviors that can be characterized as a pair of functions (f and h) and to know the detailed outcome of a system of clusters. In this subsection, we briefly recap how the g functional may be used to predict how clustering systems evolve, and we give the results of our simulations of the various situations.

In our simulations, we denote the probability of picking up a puck from a cluster of size N by

$$p_u = \int_0^\infty f(N) \rho_{N_o}(N) = \int_0^\infty f(N) \delta(N - N_o) = f(N_o) . \quad (18)$$

f is defined by the behavioral design of the agent, which in this case is leading to clustering. In our simulations, we denote the probability of dropping a puck in the cluster by

$$p_d = 1 - f(N) . \quad (19)$$

This allows us to implement a single decision controller and to still have both the pick up and drop off behavior.

It has been shown elsewhere [9][10] that the minimal condition required for a single cluster to form is that the ratio of the probabilities is a decreasing function of the number of pucks in the cluster. That is,

$$\frac{\partial}{\partial N} \left(\frac{p_u}{p_d} \right) < 0 \quad (20)$$

for all N . This in turn implies that for these probabilities,

$$\frac{\partial f}{\partial N} < 0 \quad (21)$$

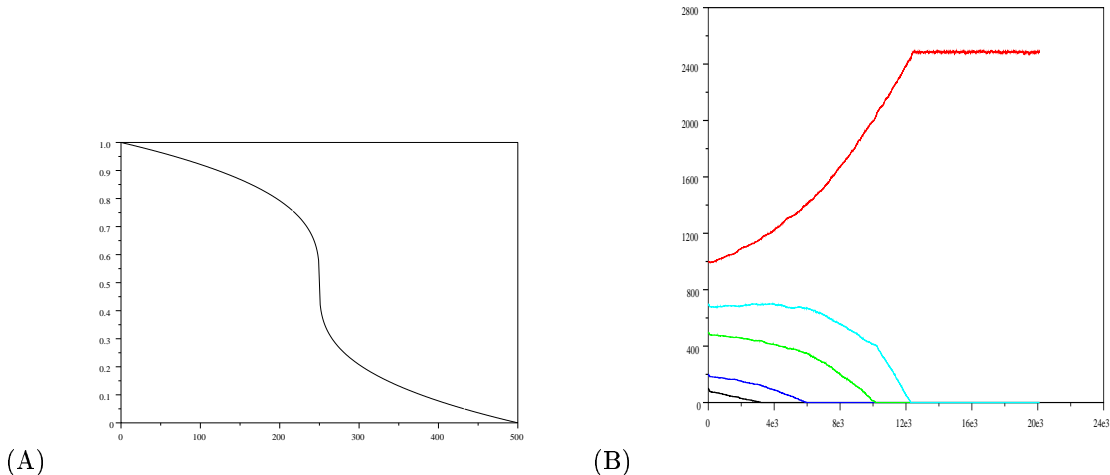


Figure 3.1: Starting with a monotonically decreasing function for g , as shown in (A), five clusters with different sizes quickly become four clusters of size zero and one of maximal size (B).

In a typical run, as shown in Figure 3.1 (B), the simulation begins with a number of different sized clusters. After several iterations the pucks from the smaller clusters are completely absorbed by the largest cluster. This is a result of a monotonically decreasing g function.

The situation is different, however, if we are interested in generating multiple clusters of the same size. It has been shown elsewhere [2] that equal sized clusters may be obtained if

$$\frac{\partial}{\partial N} \left(\frac{p_u}{p_d} \right) > 0. \quad (22)$$

In other words, we require a strongly monotonically increasing g function to create multiple identical sized clusters. This condition may be obtained from the basic behaviors if the impact of the classification of the behavior is reversed. Such a reversal requires that

$$\frac{\partial f}{\partial N} > 0. \quad (23)$$

In this case, in which g is increasing, any individual clusters initially in existence will share pucks so that the clusters become equally sized³.

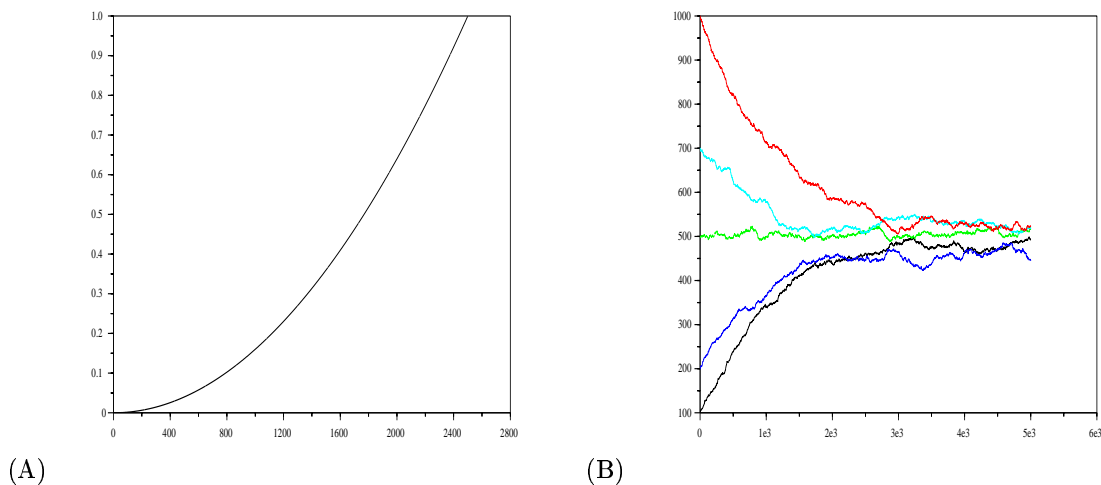


Figure 3.2: In these figures, the agent behavior is based on a monotonically increasing g function (A). This results in equal sized clusters with significant variation around the equilibrium point.

The first simulation shown starts with five clusters of different sizes. As the run progresses, four of the clusters lose their pucks significantly as the fifth one collects them. As a result, the four smaller clusters disappear, leaving only the fifth one with all the pucks. In the second example, five clusters of the same size are initialized. An arbitrary cluster manages to gain a significant advantage over the other four and ends up absorbing all the pucks. The function used to generate this behavior is also shown. This is possible because the number of clusters is fixed. In a simulation with an unfixed number of clusters, different dynamics will occur.

³This is true unless the random variation is so great that a given cluster ceases to exist.

3.3 Variance

In examining Figures 3.2, we note that while the average size of the clusters is rather predictable, it does have a fair amount of variance. As the long term goal of this work is to build systems necessary for construction, it is necessary to find methods of controlling the variance. After all, structures whose sizes vary significantly are certainly not useful to us.

We begin our investigation of the origin of the variance seen when generating multiple clusters by considering a simple system of two clusters and dynamics which produce identical sized clusters. Without proof, we assert that our results may be generalized to systems of more than two clusters, and to systems which contain clusters that evolve to differing sizes. Let n_1 equal the number of pucks in the first cluster, and n_2 equal the number of pucks in the second cluster. Also, let n_{1e} and n_{2e} equal the number of pucks of each cluster at equilibrium, which is the average size of the two clusters when generating equal sized clusters. Let r_t equal the total number of agents present, and r_c be the number of agents carrying pucks. Once again, f represents the probability of agents removing a puck from a cluster as a function of the cluster size, and h is the probability of adding a puck to a cluster. The rates of change of the number of pucks in the clusters are:

$$\frac{dn_1}{dt} = -(r_t - r_c)f(n_1) + (r_c)h(n_1) \quad (24)$$

$$\frac{dn_2}{dt} = -(r_t - r_c)f(n_2) + (r_c)h(n_2) \quad (25)$$

We may approximate the above equations using Taylor polynomials. This yields

$$\frac{dn_1}{dt} = -(r_t - r_c)[f(n_{1e}) + (n_1 - n_{1e})f'(n_{1e})] + (r_c)[h(n_{1e}) + (n_1 - n_{1e})h'(n_{1e})] \quad (26)$$

and

$$\frac{dn_2}{dt} = -(r_t - r_c)[f(n_{2e}) + (n_2 - n_{2e})f'(n_{2e})] + (r_c)[h(n_{2e}) + (n_2 - n_{2e})h'(n_{2e})] . \quad (27)$$

Rearranging we obtain

$$\frac{dn_1}{dt} = [-(r_t - r_c)((n_1 - n_{1e})f'(n_{1e})) + (r_c)((n_1 - n_{1e})h'(n_{1e}))] + [-(r_t - r_c)f(n_{1e}) + (r_c)h(n_{1e})] \quad (28)$$

and

$$\frac{dn_2}{dt} = [-(r_t - r_c)((n_2 - n_{2e})f'(n_{2e})) + (r_c)((n_2 - n_{2e})h'(n_{2e}))] + [-(r_t - r_c)f(n_{2e}) + (r_c)h(n_{2e})] . \quad (29)$$

Since n_e represents the number of pucks at equilibrium, and the rate of change at equilibrium should be 0, the equations can then be simplified to

$$\frac{dn_1}{dt} = (n_1 - n_{1e})[-(r_t - r_c)f'(n_{1e}) + (r_c)h'(n_{1e})] \quad (30)$$

and

$$\frac{dn_2}{dt} = (n_2 - n_{2e})[-(r_t - r_c)f'(n_{2e}) + (r_c)h'(n_{2e})] . \quad (31)$$

The last two equations are linear equations with restoring constants equal to $-(r_t - r_c)f'(n_{ie}) + (r_c)h'(n_{ie})$ where i represents the cluster number. Therefore, to increase the restoring constant it is necessary to increase the magnitude of $-(r_t - r_c)f'(n_{ie}) + (r_c)h'(n_{ie})$. This can be done by increasing the first derivatives of f and h at the predetermined equilibrium point.

Let us examine the behavior in practice. Suppose that we run simulations with a monotonically increasing function such as the one shown below. This particular function has a relatively small first derivative, and so its variance can be expected to be rather large.

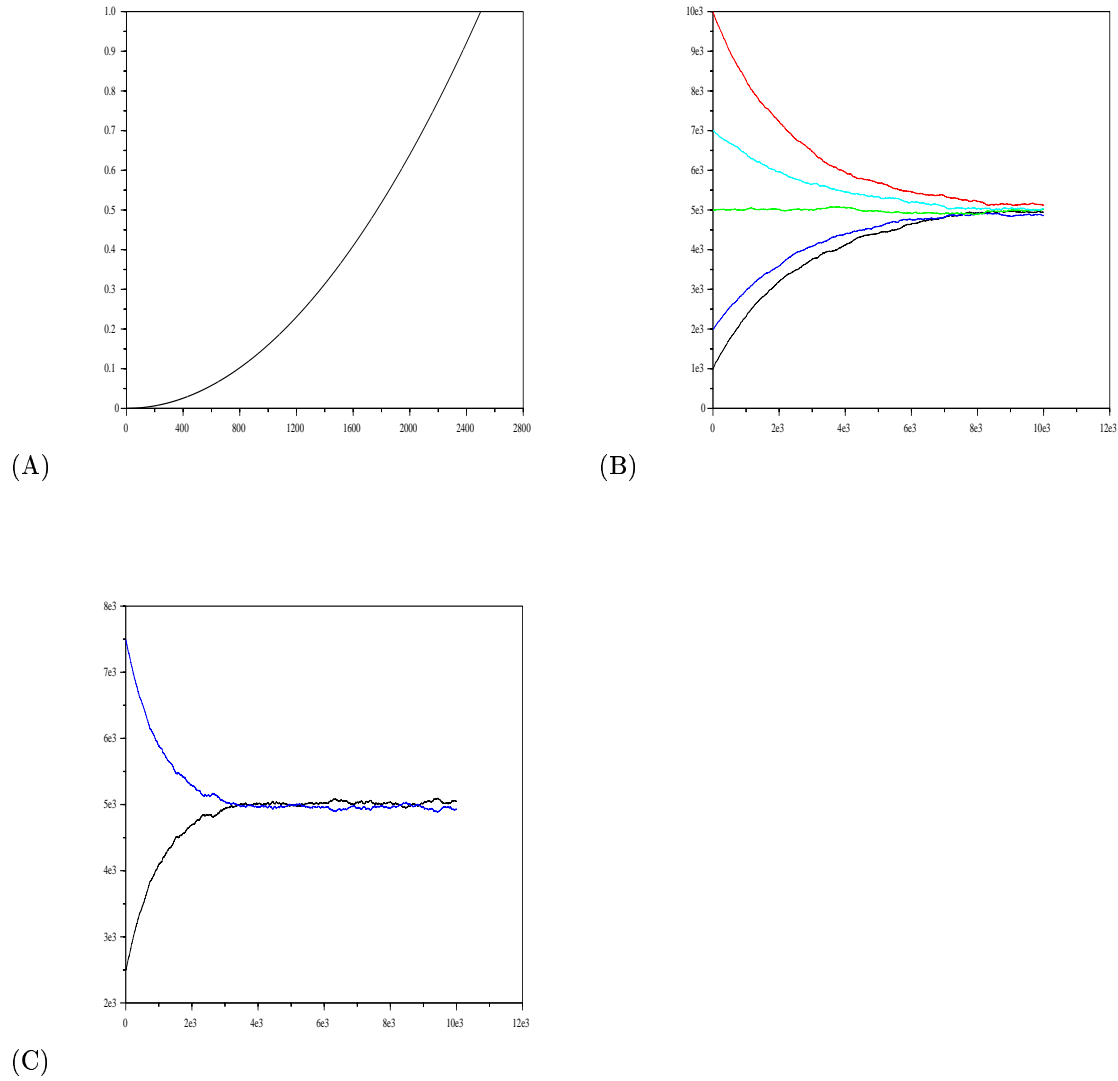


Figure 3.3: These figures depict the evolution of a standard clustering system. Figure (A) displays the function used. In (B), the system is initialized with five differently sized clusters, while in (C), the system is initialized with two differently sized clusters.

The variance in cluster size of the five clusters system is 24.590322. That of the two cluster system is 21.103539. These results were calculated using the last 100,000 iterations.

The above theory indicates that a large first derivative will equate with a greater control over the size of the cluster. Since f and h are probabilities, we are limited in their range; f and h must be from the $[0, 1]$ interval. This means that a high first derivative at the equilibrium point may be accomplished only if we make the derivative large in a restricted range, leaving the others relatively small. Such a function is depicted in Figure 3.4. In this function, the equilibrium point is in the center of the interval having a large derivative.

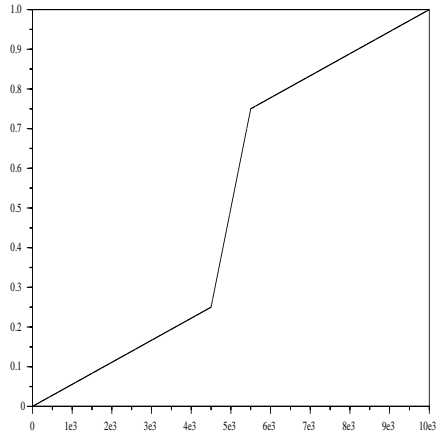


Figure 3.4: This figure depicts the g function used to reduce the variance of the cluster sizes.

When this function is applied to the same systems as those depicted in Figure 3.3, we find that the behavior is different from that found above.

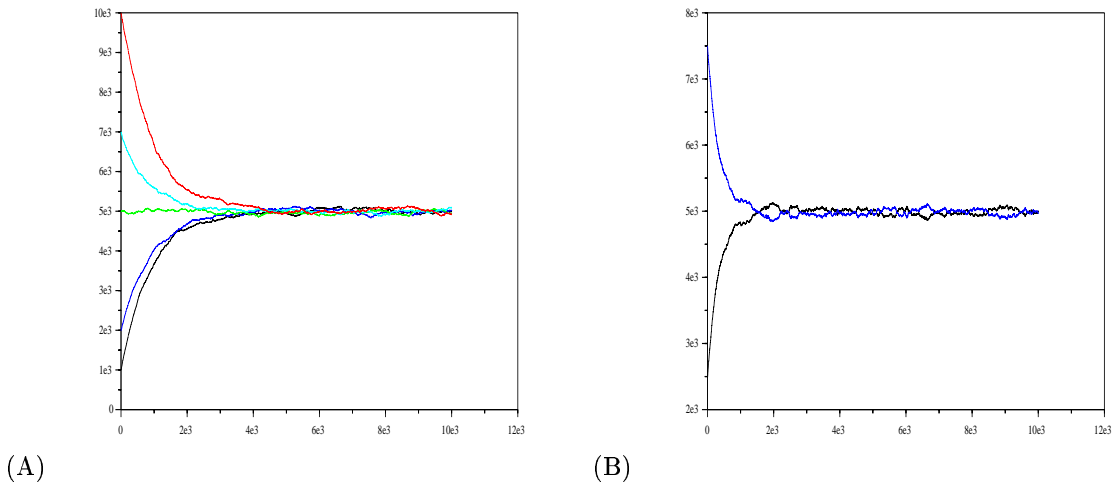


Figure 3.5: We utilize the function shown in Figure 3.4 to create five stable clusters in (A), and two stable clusters in (B).

The average variance in cluster size when utilizing five clusters is 14.298499. The average variance for two clusters is 14.583470. In both cases the variance is lower than the previous simulation, as expected.

It has been demonstrated in [9][10] that a cluster having a higher g function value than another cluster will tend to lose pucks to those with lower g function values. Since the cluster on the right side of the

equilibrium point will have a much higher g function value than the cluster on the left side, it will tend to lose pucks to the cluster on the left side, bringing both towards the equilibrium point. The loss (or gain) of pucks is proportional to the difference in g , so the greater the difference in g , the greater the likelihood that the cluster will move toward the middle. Indeed, one might expect a step function to function very nicely in this respect, as our theory predicts, and as our reasoning indicates.

We can think about this in another way. If we were to build clusters of the same size, then the probability would then be one for dropping a puck into a pile of smaller than equilibrium size, and zero of the same for dropping it in a pile greater than equilibrium. The pick up probability would be one minus the probability of drop off. Thus, the g function would be undefined up to the desired size, and zero above it. Of course, this procedure would work well *only* for clusters of a particular size. It would fail if there were either more pucks in the system or fewer clusters than planned for. The agent based system described works for any of these cases, because the probability of removal or drop off is not one.

It is important to note that the system works this way simply because the agents know, at each iteration, the precise size of the clusters they are dealing with. This allows them to make rather predictable decisions about their interactions with the clusters. Adding uncertainty in the size of the cluster removes much of the predictability. In the next section, we will revisit these issues in the case that the predictability of the cluster sizes becomes dependent on the interaction between the robot and the cluster.

4 Application of the interaction g

As we noted in Section 2, it is rarely the case that a robotic system is designed in which the robots have perfect 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 robots 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 techniques⁴, 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. For simplicity, all pucks and robots are round, though they have differing sizes (see Figure 4.1).

⁴This 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.

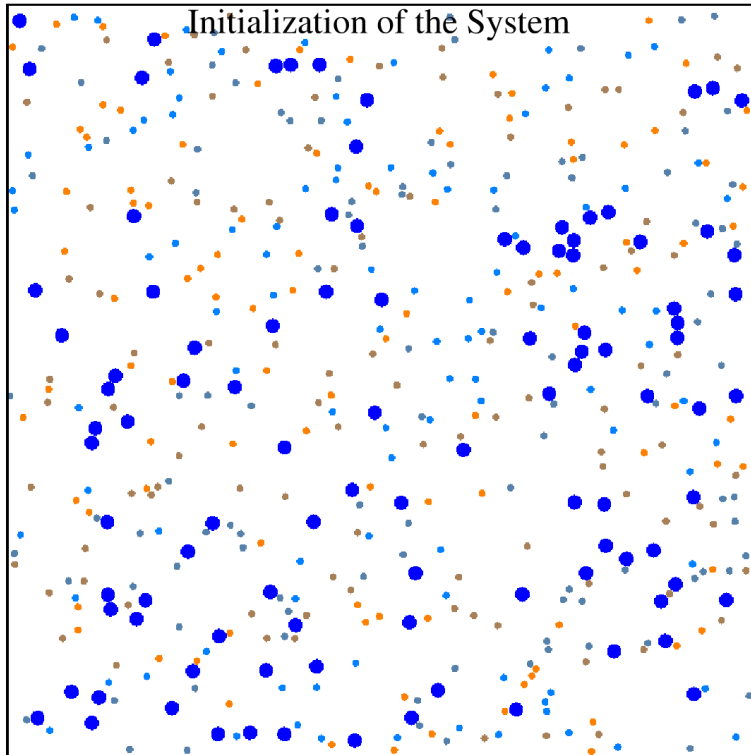


Figure 4.1: A typical initialization of the embodied simulation. Note that the large circles represent robots and the small dots represent pucks. Walls surround the robots and pucks and indicate the boundaries of the robot activity. In simulations involving robots that are not walled in, these walls have simply been removed.

Our intention is to examine the minimal condition under which clustering occurs, and to apply this to simulations of the type illustrated in Figure 4.1.

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. It is also assumed that the camera's position is above the plane in which the pucks reside, thus providing a depth measurement to the cluster. This sensor allows the robot to discern whether or not there are robots or 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.

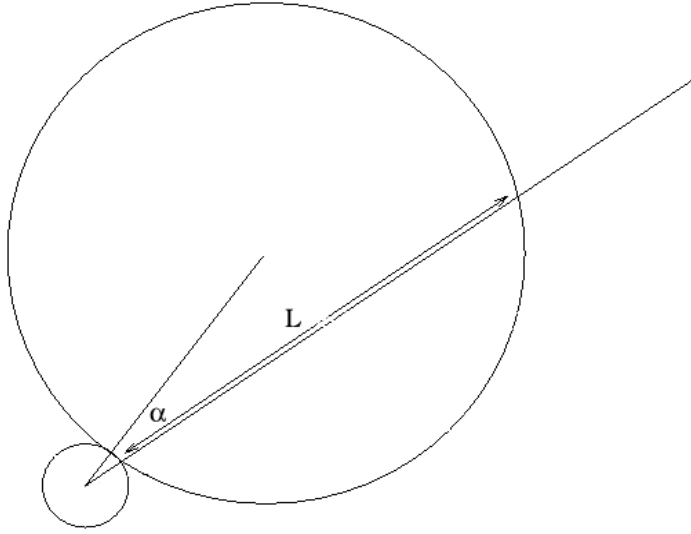


Figure 4.2: This figure gives a typical approach of a robot to the cluster. As the robot may approach from any direction, the perceived distance may range between size 0 and the diameter of the cluster. The robot would travel along the length indicated by L , and α is the respective robot's angle of approach.

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 robot⁵.

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 circular⁶.

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 a

$$\alpha = d(L) . \tag{32}$$

In our model, this indicates that

$$d(L) = \sin^{-1} \left(\frac{R}{R+r} \sin \left(\cos^{-1} \left(\frac{L}{2R} \right) \right) \right) \tag{33}$$

⁵It is important to note that this is a specific design paradigm, and not an accidental flaw. Other methods, such as scanning the cluster, finding the cluster's center might be used to more accurately find the cluster's size. However, the errors in the cluster size are, as we shall explain, the *driving force behind the clustering action*, and so these errors are left in the system deliberately. The methods just described can contribute to the building of large, brittle clusters whose dynamics can be unpredictable, as robots removing random pucks may also break the clusters into one or more separate units.

⁶While this is at this moment, and for the purpose of this work, an unimportant property of the system outside of the simplification of the following analyses, this is an important property when moving clusters, which is a necessary part of swarm-based construction.

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

$$1 = \int_{\alpha_{min}}^{\alpha_{max}} \rho(\alpha) d\alpha = \int_{\alpha_{min}}^{\alpha_{max}} \frac{1}{\Delta\alpha} d\alpha \quad (34)$$

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

$$p = \frac{L}{2R} \quad (35)$$

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

$$D(p) = \sin^{-1} \left(\frac{R}{R+r} \sin(\cos^{-1}(p)) \right) \quad (36)$$

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

$$1 = \int_{\alpha_{min}}^{\alpha_{max}} \frac{1}{\Delta\alpha} d\alpha = - \int_0^1 \left(\frac{1}{\Delta\alpha} \frac{dD}{dp} \right) dp \quad (37)$$

Therefore,

$$\rho(p) = - \frac{1}{\Delta\alpha} \frac{dD}{dp}. \quad (38)$$

ρ is thus given by

$$\rho(p) = \frac{R}{\Delta\alpha (R+r) \sqrt{(1-p^2) - \frac{R^2(1-p^2)^2}{(R+r)^2}}}. \quad (39)$$

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

$$\rho(p) = \frac{1}{1.1 \sin^{-1} \left(\frac{1}{1.1} \right) \sqrt{(1-p^2) - \frac{(1-p^2)^2}{1.1^2}}}. \quad (40)$$

which has the graph shown in Figure 4.3.

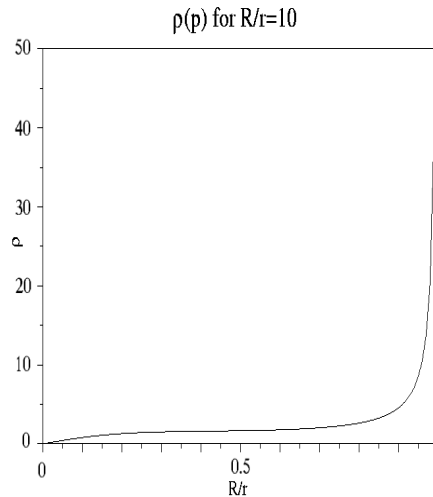


Figure 4.3: This gives the probability density of a robot approaching a cluster ten times larger than the robot and observing a cluster of size p , where p is a percentage of the cluster size.

If $R \gg r$, this reduces to

$$\rho(p) = \frac{1}{\Delta\alpha\sqrt{1-p^2}}. \quad (41)$$

With this probability density,

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

$$\langle p \rangle = \frac{2}{\pi} \int_0^1 \frac{p}{\sqrt{1-p^2}} dp = -\frac{2}{\pi} \sqrt{1-p^2} \Big|_0^1 = \frac{2}{\pi} = 0.63661979 \quad (42)$$

which gives

$$\langle L \rangle = 1.2732395R. \quad (43)$$

This last equation gives us the average ratio of the observed size to the actual size observed by the robot.

Moreover, for clusters whose radius is much greater than the robot, this ratio 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.

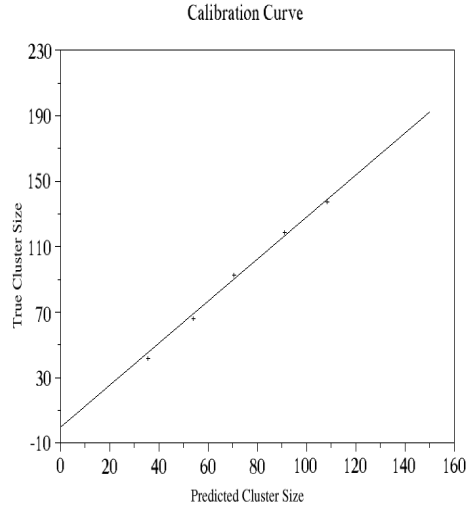


Figure 4.4: When using our metaphysical simulation, the calibration curve is as given. This curve is “nearly linear” and deviates very slightly from linear because the ratio of robot and cluster sizes changes over this range. The least squares regression line of this function is defined as $y = -0.16997 + 1.284195x$.

Figure 4.4 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

4.2 Clustering condition

It has been shown [2][10] that the minimal condition for generating a single cluster employing the reaction g is a monotonically decreasing g function. However, the same conclusion does not necessarily apply to the interaction g . In this section, we examine the conditions under which a single cluster will always be produced.

The theory presented by Kazadi et. al. [10] on densities indicates that for clustering to occur, the density of puck carrying robots should be smaller around larger clusters than it is around smaller clusters. We investigate the conditions under which this behavior may result from a monotonically decreasing g function, which is analogous to the theory presented in the reaction g section. The robots in the system interact with the cluster in accordance with the previously defined thresholding functions F and H of equation (12), which are analogous to f and h , but now include the geometric properties of the arena and cluster position(s). D is the time averaged density of robots in a region, and δ_f is the density of robots around a cluster that are currently holding pucks. Then it is the case that [10]

$$\delta_f = D \frac{1}{1 + \frac{H}{F}} \quad (44)$$

This density is valid when using the reaction g and uniform dispersion of the robots. If the robot density itself is a function of position, this may be rewritten as

$$\delta_f(\vec{x}) = D(\vec{x}) \frac{1}{1 + \frac{H}{F}}. \quad (45)$$

This can be rewritten as

$$\delta_f(\vec{x}) = D(\vec{x}) \frac{F}{F + H} \quad (46)$$

In general the robots do not know the actual size of the cluster, instead they 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

$$\langle D(\vec{x}) \rangle = D(\vec{x}) \int_0^\infty \frac{F(N)}{F(N) + H(N)} \rho_{N_o}(N) dN \quad (47)$$

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

$$D(\vec{x}_{N_1}) \int_0^\infty \frac{g(N)}{g(N) + 1} \rho_{N_1}(N) dN < D(\vec{x}_{N_2}) \int_0^\infty \frac{g(N)}{g(N) + 1} \rho_{N_2}(N) dN. \quad (48)$$

whenever $N_1 > N_2$. If $D(\vec{x}_{N_1}) = D(\vec{x}_{N_2})$ ⁷ this condition reduces to

$$\int_0^\infty \frac{g(N)}{g(N) + 1} \rho_{N_1}(N) dN < \int_0^\infty \frac{g(N)}{g(N) + 1} \rho_{N_2}(N) dN. \quad (49)$$

This condition may be known as the *generalized clustering condition*. It is easy to see that if $\rho_{N_o}(N) = \delta(N - N_o)$ the condition reduces to equation (52), our value in the reaction g . I.e.

$$\int_0^\infty \frac{g(N)}{g(N) + 1} \delta(N - N_1) dN < \int_0^\infty \frac{g(N)}{g(N) + 1} \delta(N - N_2) dN \quad (50)$$

⁷This is generally true if the robots are able to move freely from one cluster to another.

$$\frac{g(N_1)}{g(N_1) + 1} < \frac{g(N_2)}{g(N_2) + 1} \quad (51)$$

which reduces to

$$g(N_1) < g(N_2) . \quad (52)$$

In essence what this 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.

4.3 Metaphysical robot clustering

In swarm engineering, our primary goal is to determine minimal conditions required to obtain a global outcome and then to build robot systems exhibiting these minimal conditions. In equation (48) we have the general clustering condition. We now build a robotic system and demonstrate the conditions under which this system satisfies the generalized clustering condition. In Section 4.4, 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 robots are meant to be extremely simple. This is to facilitate the generation of real robots in the laboratory capable of doing what our simulated robots are. With this in mind, we design our robot controllers as follows:

1. The robot will move around according to simple random or semi-random behaviors.
 - (a) In a bounded arena, in which the simulation limits the range of the robot to a rectangular arena, robots move in a straight line until encountering a wall or another robot, at which point the robot will execute a random turn away from the object, and continue moving.
 - (b) In an unbounded arena, the robot is given an unlimited arena to move in, and thus a spiraling behavior is implemented. When the robot is unable to see any pucks, it begins to spiral in a random direction, thus allowing the robot to return to the clusters.
2. When the robot approaches a cluster, it generates an estimate of the size of the cluster as a result of its approach angle and position. This behavior has been explained in Section 4.1.
3. If the estimate is that the cluster size is below a minimal threshold, the robot will deposit a carried object with probability p or pick up an object from the cluster with a probability $1 - p$. This facilitates pseudoclustering, and usually is subject to a small probability p .
4. If the estimate is that the cluster size is above a minimal threshold and below a maximal threshold, the robot will deposit a carried object.
5. If the estimate is that the cluster size is above a maximal threshold, the robot will pick up an encountered object. However in generating single clusters, the maximum threshold is disregarded.

In our simulations, the robot to robot interactions as well as robot to puck interactions are simulated with as much realism as possible. As previously described, robots travel in straight lines at a given speed until they encounter other objects. Based on the object it encounters, the robot 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 robot 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 robot can either drop or pick up a puck, or simply ignore the

cluster. When a robot encounters another robot, both robots execute a random turn away from each other. When a robot encounters a wall in the bounded simulation, the robot executes a random turn away from the wall.

In the unbounded simulation, the robots are allowed significant freedom of movement, which makes it possible for the robot to wander away from any pucks or robots. In order to avoid the loss of robots due to wandering, we implement a spiraling behavior that causes the robots to begin spiraling in a random direction once the robot is unable to see any pucks. This eventually causes the robot to return to the pucks if they are temporarily lost. Other than the spiraling effect, the other robot 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 robot 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 robot which correctly assigns it a small cluster designation.

Runs generated by the embodied simulation are much more complex and realistic than from the non-embodied simulation, providing much more reliable results that may be verified in later, real-world experiments. Because of the realism in the simulation, significantly more time is required to undertake these simulations than the previous simulation. This limits our ability to test conditions with more than a few hundred robots or pucks.

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 robot approaching a cluster (Figure 4.2). We may determine the minimal approach angle resulting in the robot's placing a puck in the cluster as a function of the robot size r , the minimal distance d_{min} , and the cluster size R . We note that the maximal angle between the direction of robot motion and the connector between diameters is formed when

$$\Theta_{max} = \sin^{-1} \frac{R}{R+r} . \quad (53)$$

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

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

$$\alpha_{min} = \sin^{-1} \left(\frac{R}{R+r} \sin \left(\cos^{-1} \left(\frac{d_{min}}{2R} \right) \right) \right) . \quad (54)$$

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

$$p_d = \frac{\alpha}{\theta_{max}} = \frac{\sin^{-1} \left(\frac{R}{R+r} \sin \left(\cos^{-1} \left(\frac{d_{min}}{2R} \right) \right) \right)}{\sin^{-1} \frac{R}{R+r}} . \quad (55)$$

This also puts the probability of pickup at

$$p_u = 1 - \frac{\alpha}{\theta_{max}} = 1 - \frac{\sin^{-1}\left(\frac{R}{R+r} \sin\left(\cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right)}{\sin^{-1}\frac{R}{R+r}}. \quad (56)$$

Together, these produce a g equal to

$$g = \frac{p_u}{p_d} = \frac{\sin^{-1}\frac{R}{R+r}}{\sin^{-1}\left(\frac{R}{R+r} \sin\left(\cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right)} - 1 \quad (57)$$

This g function is monotonically decreasing, and satisfies at least part of the clustering condition. Clearly equation (48) 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 .

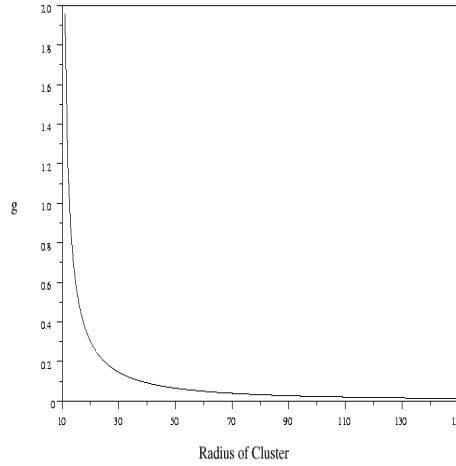


Figure 4.5: This figure illustrates the form of g as a function of R . g is a decreasing function of R , which is the first condition of clustering.

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.

Note that increasing the minimum size in a thresholding model will increase the sharpness of this graph. That is, tightening the constraints on the size of the cluster makes g more sharply sloped. The effect of this will be explored later and compared with the effects that we observed in Section 3.

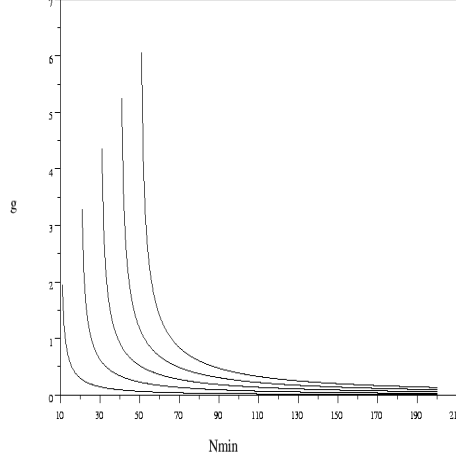


Figure 4.6: This figure illustrates the effect on g of increasing the N_{min} . Note that the greater N_{min} is, the greater the slope is of g around the equilibrium point.

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

$$\int_0^a (\rho_{N_2}(N) - \rho_{N_1}(N)) \frac{g(N)}{g(N)+1} dN > \int_a^\infty (\rho_{N_1}(N) - \rho_{N_2}(N)) \frac{g(N)}{g(N)+1} dN \quad (58)$$

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 (46) to

$$\langle D \rangle = D \int_0^{N_{min}} \rho_{N_o}(N) dN \quad (59)$$

and (57) to

$$\int_0^{N_{min}} (\rho_{N_2}(N) - \rho_{N_1}(N)) dN > 0 \quad (60)$$

providing that $a > N_{min}$. If $N_{min} > a$ then (59) becomes

$$\int_0^a (\rho_{N_2}(N) - \rho_{N_1}(N)) dN > \int_a^{N_{min}} (\rho_{N_1}(N) - \rho_{N_2}(N)) dN . \quad (61)$$

In the case that $R \gg r$ the probability is as given in equation (39). 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 (59) as

$$\int_0^{N_{min}} \left(\frac{1}{\sqrt{2N_2N - 2N^2}} - \frac{1}{\sqrt{2N_1N - 2N^2}} \right) dN > 0 \quad (62)$$

$$\frac{1}{\sqrt{2}} \int_0^{N_{min}} \left(\frac{1}{\frac{N_2}{2} \sqrt{1 - \left(\frac{2N}{N_2} - 1\right)^2}} - \frac{1}{\frac{N_1}{2} \sqrt{1 - \left(\frac{2N}{N_1} - 1\right)^2}} \right) dN > 0 \quad (63)$$

$$2\sqrt{2} \int_{-1}^{\frac{2N_{min}}{N_2} - 1} \frac{1}{\sqrt{1 - q^2}} dq - 2\sqrt{2} \int_{-1}^{\frac{2N_{min}}{N_1}} \frac{1}{\sqrt{1 - q^2}} dq > 0 \quad (64)$$

$$\sin^{-1} \left(1 - \frac{2N_{min}}{N_1} \right) > \sin^{-1} \left(1 - \frac{2N_{min}}{N_2} \right) \quad (65)$$

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 4.7.

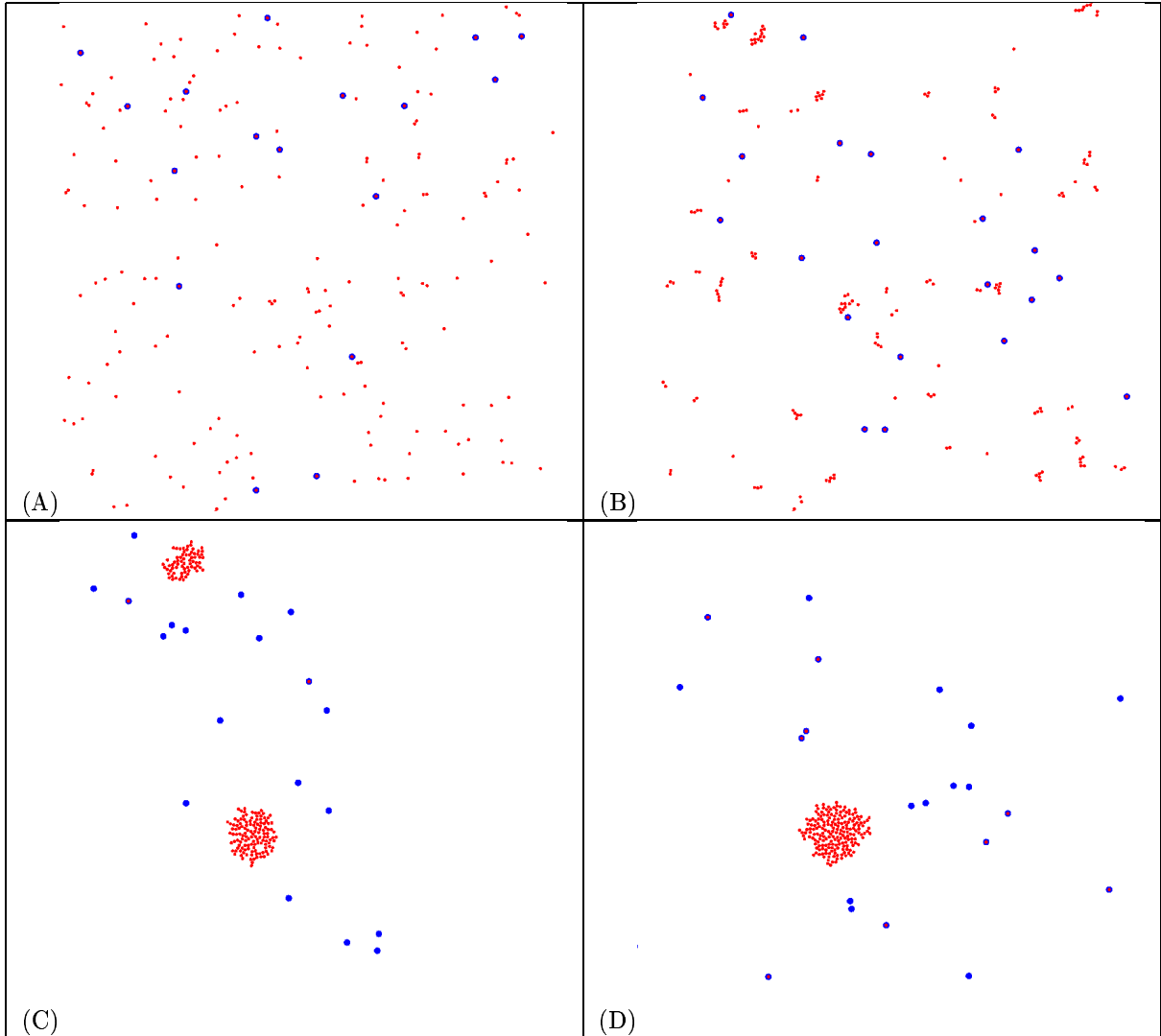


Figure 4.7: This simulation is unbounded and initialized with arbitrarily scattered pucks. Clusters form, one of which is considerably more dominant than the other. As in the non-embodied simulation, we find that the dominant cluster absorbs all the pucks.

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 robot 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

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

$$= \begin{cases} \frac{\sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right)}{\sin^{-1}\frac{R}{R+r}} & \text{if } \frac{d_{max}}{2} > R > \frac{d_{min}}{2} \\ \frac{\sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right) - \sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{d_{max}}{2R}\right)\right)\right)}{\sin^{-1}\frac{R}{R+r}} & \text{if } \frac{d_{max}}{2} \leq R \end{cases} \quad (66)$$

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

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

$$= \begin{cases} 1 - \frac{\sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right)}{\sin^{-1}\frac{R}{R+r}} & \text{if } \frac{d_{max}}{2} > R > \frac{d_{min}}{2} \\ 1 - \frac{\sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right) - \sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{d_{max}}{2R}\right)\right)\right)}{\sin^{-1}\frac{R}{R+r}} & \text{if } \frac{d_{max}}{2} \leq R \end{cases} \quad (67)$$

In this case, the ratio g is given by

$$g = \frac{p_u}{p_d} = \begin{cases} \frac{\sin^{-1}\frac{R}{R+r}}{\sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right)} - 1 & \text{if } \frac{d_{max}}{2} > R > \frac{d_{min}}{2} \\ \frac{\sin^{-1}\frac{R}{R+r}}{\left(\sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{d_{min}}{2R}\right)\right)\right) - \sin^{-1}\left(\frac{R}{R+r}\sin\left(\cos^{-1}\left(\frac{d_{max}}{2R}\right)\right)\right)\right)} - 1 & \text{if } \frac{d_{max}}{2} \leq R \end{cases} \quad (68)$$

This function 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.

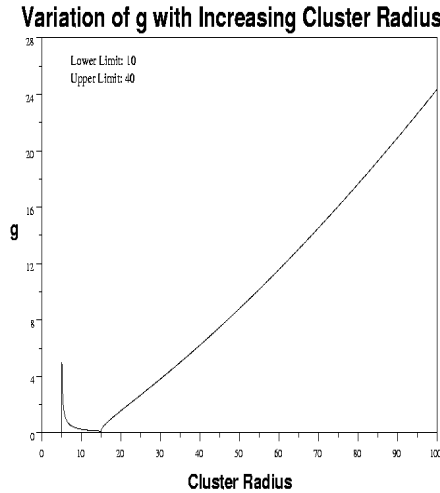


Figure 4.8: This gives the behavior of g as a function of the cluster radius. In this case, the function has a minimum, and increases monotonically after this minimum. This minimum represents the equilibrium size for the cluster that results from this behavior. d_{max} is 30 and d_{min} is 90. The minimum is at 45.

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

$$\langle D \rangle_1 = D \int_0^\infty \frac{F(N)}{F(N) + H(N)} \rho_{N_1}(N) dN \quad (69)$$

while that around the cluster of size N_2 is

$$\langle D \rangle_2 = D \int_0^\infty \frac{F(N)}{F(N) + H(N)} \rho_{N_2}(N) dN . \quad (70)$$

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

$$\langle D \rangle_1 = D \int_0^{N_{min}} \rho_{N_1}(N) dN \quad (71)$$

while the second becomes

$$\langle D \rangle_2 = D \left(\int_0^{N_{min}} \rho_{N_2}(N) dN + \int_{N_{max}}^{N_2} \rho_{N_2}(N) dN \right) . \quad (72)$$

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

$$\int_0^{N_{min}} \rho_{N_2}(N) dN + \int_{N_{max}}^{N_2} \rho_{N_2}(N) dN > \int_0^{N_{min}} \rho_{N_1}(N) dN \quad (73)$$

assuming, of course, that the densities are equal. Rearranging, we obtain

$$\int_{N_{max}}^{N_2} \rho_{N_2}(N) dN > \int_0^{N_{min}} (\rho_{N_1}(N) - \rho_{N_2}(N)) dN . \quad (74)$$

While, 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

$$\int_{N_{max}}^{N_2} \rho_{N_2}(N) dN > 0. \quad (75)$$

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 4.9, we illustrate this process by creating two clusters of equal size. Initially, the cluster material is randomly placed. Then 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.

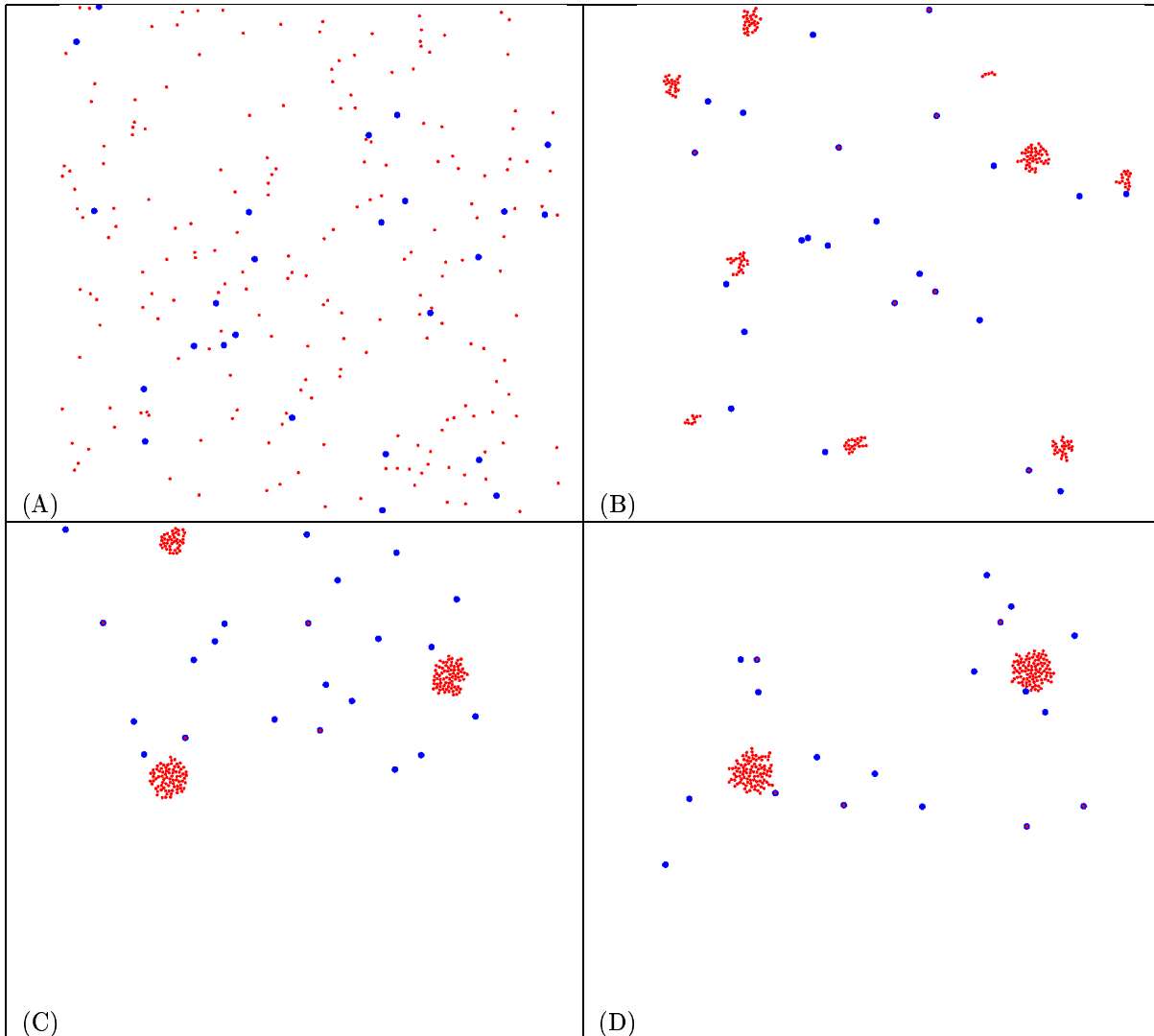


Figure 4.9: This simulation shows a run in which two clusters of equal sizes are formed. The net change of each of the cluster is balanced, so no cluster maintains a clear dominance over the other.

4.4 Variance

Of course the long term goal in these theoretical and experimental efforts is to generate a method of building structures of both a predetermined size and of a limited variance. We have investigated the variance in the reaction g and have found that the variation in systems under the reaction g can be reduced by increasing the first derivatives of f and h at a predetermined equilibrium point. However, this situation is not necessarily the case in the interaction g . In Section 2, we showed that the interaction g forces the variance G' to be given by

$$G' = \rho_{N_o}(N_{min}) - \rho_{N_o}(0) + \rho_{N_o}(N_{max}) - \rho_{N_o}(\infty) \quad (76)$$

This has a very different behavior than the reaction g . First, it is usually the case in most controllers that $\rho_{N_o}(\infty) = 0$. We therefore neglect this term. As we have stipulated that these are normal probabilities ($\frac{d^2\rho}{dx^2} < 0$), there can be only one maximum. Since the probabilities are likely to be peaked around the cluster size, $\rho_{N_o}(N_{min}) > \rho_{N_o}(0)$. This means that the first two terms make a positive term. The last term, that of $\rho_{N_o}(N_{max})$ is always positive or zero, and is zero only if the probability of miscalculating the size of a cluster smaller than N_{max} is zero. If the cluster is of size N_{max} , the probability is positive, and is due to the random nature of puck drop off and pick up.

This also leads us to the same conclusion that we had before. 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 in the reaction g . In this case, we find that as g increases, the variance increases. Moreover, 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.

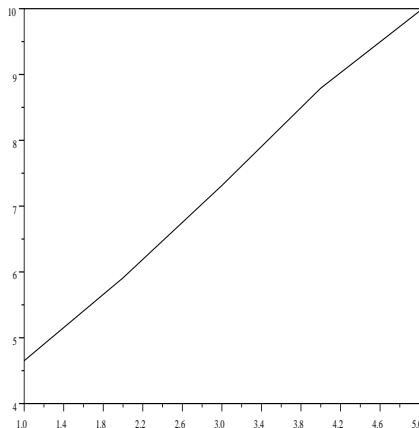


Figure 4.10: The variance of the cluster increases as N_{min} increases.

5 Discussion and conclusions

The importance of this work becomes apparent when analyzing the two different types of simulations. 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 .

The differences between the two systems hinges on their ability to determine cluster size. While one determines the cluster size by an estimation based on the overall size, the other relies on probabilities to drive agent behavior. This discrepancy leads to different uses for either simulation. If one is concerned with clustering that employs the agent's location to determine behavior, such as the two dimensional clustering employed in the embodied simulation, one would need an understanding of the interaction g . On the other hand, if one is concerned with clustering that is based on cluster sizes rather than agent location, such as stacking, then one would need an understanding of the reaction g because stacked pucks can be easily measured. An understanding of the fundamental theory behind each of the systems is necessary to accomplish certain tasks.

We are but in the first steps toward 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 distinct behavioral designs of agents for puck clustering systems.

6 Acknowledgements

R. Cho, M. Chung, and B. Lee would like to thank their parents for providing much of the financial backing for their work at the Jisan Research Institute. S. Kazadi would like to thank the men and women of the Jisan Research Institute, without whom the Institute would not function, and this study would not have been possible.

References

- [1] W. Cohen. Adaptive mapping and navigation by teams of simple robots. *Robotics and Autonomous Systems*. 1996.
- [2] A. Zhang. et. al. *Variance in Converging Puck Cluster Sizes*. **First International Joint Conference 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 cemetery 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.