REMOVING DEGENERACY FROM SWARM-MEDIATED CLUSTER-BASED CONSTRUCTION

Sanza T. Kazadi
Jisan Research Institute
28 N Oak Ave
Pasadena, CA, 91107, USA
email: sanza@jisan.org

Olga I. Koroleva
Jisan Research Institute
28 N Oak Ave
Pasadena, CA, 91107, USA
e-mail: olga@jisan.org

ABSTRACT

In this paper we design swarm clustering algorithm to build, move and place clusters of building materials on the desired place on $2D$ plane. Such algorithm consists of three phases: building clusters, moving them radially and then orbitally. We present an example that demonstrates desired cluster placement. In particular, it is shown that with proposed algorithm capable of building clusters, moving and placing them with respect to each other in accordance with given requirements.

KEY WORDS

swarm engineering, cluster, puck, agent.

1 Introduction

There are, in the natural world, swarms of insects which have evolved over long periods of time and which possess the capability to construct large structures, many times larger than the individual insect in the swarm. Such structures can range in design from those with remarkable regularity, such as honeycombs of bees and wasps, to the somewhat irregular structures of termites, ants, or communal spiders. Swarms in the natural world are not only capable of building the homes in which they live, but they also interact with them, providing repair services when needed. Moreover, the swarms' homes can be constructed in such a way that the swarm can use them to maintain atmospheric conditions with a control that rivals systems built by Mankind.

The abilities of these insect swarms have prompted scientists to propose the design of agent-based systems that carry out similar tasks. Such systems would enjoy many of the same benefits one finds in the natural world:

1. The individuals may be rather simple in design and capability.
2. The swarm may be capable of carrying out tasks that are beyond the capability of individual agents.
3. The swarm can be extremely robust in comparison to the robustness of the individual agent.

Among these systems are puck clustering systems, in which agents are involved in the movement of generalized construction materials known as pucks. Pucks are typically moved between clusters by autonomous agents, with each agent moving one or more pucks at a time. The general goal of clustering systems is to create some specific number of clusters in a predetermined organization. Typically, each agent carries out a series of actions which is stochastic in nature, resulting in random movements between or around clusters.

Puck clustering systems have the potential to lead to more complex construction systems. Since it is now possible to generate a particular number of clusters of a predetermined size, the next step is to place the clusters in particular relative positions. Once this step is accomplished, the positions of the clusters can be used as markers for subsequent construction tasks. As an example, wall structures may be placed between clusters and columns may be built on top of them. This potentiality indicates a unique architecture, and potential method for generating more complex structures than clusters.

Since the early nineties, extensive work has been done on clustering systems [1, 2, 3, 5, 11]. Initial studies examined the use of small robotics systems in the cooperative collection of pucks. These efforts centered around physically instantiating the clustering robot. As a rule, these studies are either successful in generating a single cluster [4, 6] or in generating several clusters. The success or failure of the clustering systems are determined by the geometric properties of the system. These properties can include the size of the arena, the arrangement (or absence) of grippers on the robot, and other detailed parameters of the system. However, what is lacking in these studies is a general description of just what it takes to create a system that successfully clusters pucks.

In contrast to earlier studies, Kazadi et. al. in [7] investigates several of the properties of clustering systems, leading to a general description of what it takes for a clustering system to yield a single cluster. The work allows for the development robust clustering as a component of larger systems capable of carrying out multiple tasks (either in parallel or sequentially). Thus, a researcher needs only to build a robot which satisfies the general condition and the system will produce one cluster.

We consider randomly placed robots and pucks on $2D$ plane as initial configuration of the system as depicted in

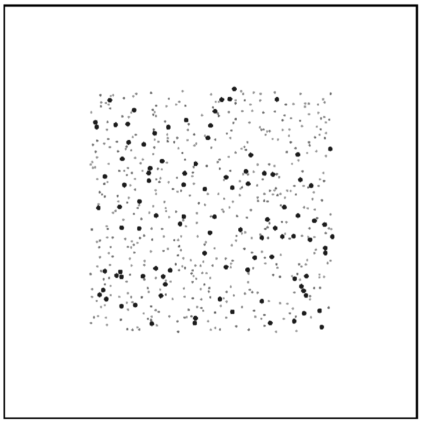

Figure 1. Initial configuration of a clustering system consisting of robots (large) and pucks (small circles).

Figure 1. Similar systems are developed in Kazadi et. al. [9] in which several clusters are produced in a predictable way.

One side effect of the system, however, is that clusters produced in this way appear in random places. This is an unavoidable consequence of the system design as the individual agents are not supposed to have access to detailed global information. As a result the clusters must be moved from their original positions to new relative positions if they are to be used as a basis for generating structures of predetermined design. In the study by Kazadi et. al. [10], methods of generating multiple clusters are coupled with methods of moving clusters into specific relative positions. However, in this study, it is determined that the exact placement of several clusters of different designs suffers from the existence of degenerate states. These stable states result from cluster movement away from their points of origin. Occasionally, and with increasing frequency as the number of clusters in the design increases, the clusters move to a formation which cannot be altered by the swarm of agents. This is a significant problem for consideration of the potential use of swarm-based clustering systems as construction tools. It is hardly reasonable to develop building tasks in which one of a multitude of degenerate states might occur.

As a result, we explore a method of removing the state degeneracy which plagues current efforts in swarm-based construction. We present recent work which effectively removes the degeneracy. In Section 2, we review the major elements of puck clustering systems. Section 3 gives an account of the degeneracy problem and describes method used to remove the degeneracy problem. Its use on the construction tasks is illusrated by examples in Section 4. Finally, Section 5 offers some concluding remarks.

2 Puck Clustering Systems

Puck clustering systems are still in their infancy in regards to their ability to accomplish tasks of significance. However, even though the systems are in their infancy, a significant effort has been expended on developing their properties using methodologies based in predictive theory. This section reviews some of the results of previous studies, particularly those relevant to construction efforts.

2.1 Single Cluster Evolution

Puck clustering systems have been characterized theoretically in several ways. The simplest of these ways is the description of the requirements for individual agents which lead to the swarm's clustering of pucks in a single cluster. We assume that robots' behaviors can be characterized by two different interactions: the interaction of an agent carrying a puck and that of an agent not carrying a puck. The agent which is carrying a puck may drop the puck at the cluster, while that not carrying a puck must pick up a puck depending on what it's assessment of the situation is. Suppose that $f\left(N\right)$ represents the probability, under a particular design, that a robot not carrying a puck will pick one up from a cluster containing N pucks. Moreover, suppose that $h\left(N\right)$ is the analog for agents carrying pucks and potentially dropping off a puck at a cluster with N pucks. Then, letting $g = \frac{f}{h}$, it is the case that the swarm will generate a single cluster if

$$\frac{\partial g}{\partial N} < 0, \tag{1}$$

where the strict inequality is required. Once an agent is designed with behavior that satisfies this condition, the clustering behavior of the swarm is guaranteed [7].

Another important design concern is the speed of clustering. Once we have guaranteed that the system will lead to a single cluster, we would like to know whether or not the method we are using can be supplanted by another method that is quicker in terms of its convergence time. That is, if we have two behaviors characterized by function pairs (f_A, g_A) and (f_B, g_B), under what conditions can it be shown that swarm A will be quicker in converging than swarm B? This condition was derived in [7], where it was shown that swarm A would be more efficient than swarm B iff

$$g_A < g_B. \tag{2}$$

This design requirement may be applied to the design of agents, yielding increasingly efficient clustering.

Finally, it is interesting to ask whether or not it is possible for agents to produce several clusters instead of one, with the final clusters taking on a predetermined size and multiplicity. This is indeed possible [9], and the condition required for the development of these is remarkably simple. Rather than a strongly monotonically decreasing g functional, the condition is the existence of a multiphasic g function containing one decreasing phase followed by an increasing phase. The point at which the g function shifts from a decreasing to an increasing phase is the equilibrium point N_0, the maximum size of the cluster(s).

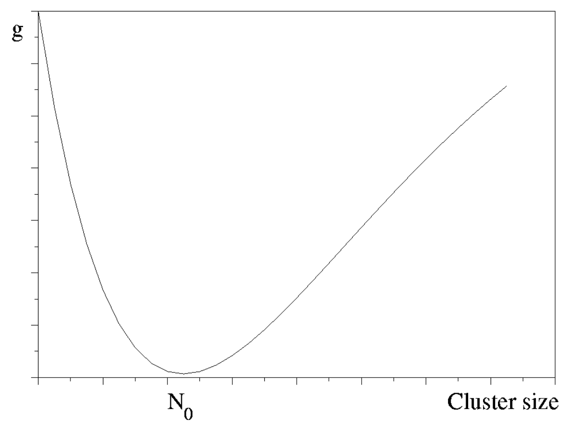

Figure 2. This figure illustrates a potential multiphasic g function with a minimum at N_0. Such a multiphasic g function yields multiple clusters with size N_0 pucks.

2.2 Cluster Motion

In general, the clusters in random clustering systems appear in unpredictable locations. The process of building the clusters from individual elements is stochastic in nature, and any single element may be used as a starting point for what may eventually be the final cluster. As a result, the cluster may start in any location. Of course, if someone is to build a structure using clusters as initial markers of structural design, it must be possible to place the clusters in specific positions relative to one another. This requires that movement of the cluster be part of the capability of the swarm.

In order to move a cluster without changing its shape or disassembling it and reassembling it in a new location, it is necessary to generate a net movement of pucks from one side of the cluster to the other. The situation is depicted in Figure 3.

On the right side of the cluster, the robots are more likely to pick up pucks while on the left side, they are more likely to drop off pucks. The situation causes a gradient in g values from right to left, an indicator that a puck movement is occurring between the areas of high g values and low g values.

This can generally be accomplished in a number of ways. One method that has been employed in [10] involves differential bias of the likelihood of drop off as the angle of the robot encountering the cluster varies. The method is a variation on a thresholding model of cluster size maintenance. In the thresholding model, the probability of pick up of a puck is given by

$$f(N) = \begin{cases} 1-p, & N < l_b, \\ 0, & l_b < N < u_b, \\ 1, & u_b < N, \end{cases} \tag{3}$$

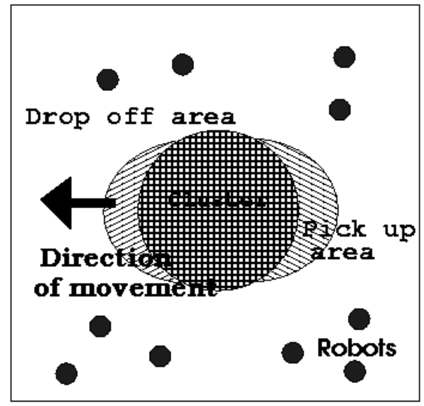

Figure 3. This figure illustrates the creation of a g-gradient. Such a gradient indicates a net movement of pucks from high g value areas to low g value areas. In this case, the areas are on opposite sides of a cluster, and so lead to a movement of the cluster.

where p is some small number, l_b is the lower threshold size of the cluster, and u_b is the upper threshold of the cluster. The threshold model is altered by choosing u_b as follows

$$u_b' = u_b \left(1 - e\cos(\phi)\right), \tag{4}$$

where ϕ represents the angle between the desired direction of motion and the angle that the robot is facing. By increasing the upper threshold opposite the desired direction of motion and decreasing the upper threshold in the desired direction, the effect is that on the side opposite the direction of motion, the robots attempt to decrease the cluster size while on the other side the robots attempt to increase the cluster size. The effect is a creeping motion in the desired direction of motion *provided that the cluster size is at least as close to the upper threshold as* e. If the cluster is not large enough, then there will be no effect from the changing of the upper threshold.

This method of motion generation has interesting consequences. Among these is the functional form of the speed of the cluster. This is given by

$$v_{max} = \frac{4\alpha r_t}{\pi d_p \sqrt{N}} \frac{f(N)h(N)}{f(N)+h(N)} \tag{5}$$

where f and h are as defined above, r_t is the total number of robots, α is the density of robots at the cluster, d_p is the diameter of a puck, and N is the number of pucks in the cluster. Interestingly, this speed is inversely proportional to the square root of the number of pucks in the cluster. This reflects the fact that smaller clusters are much easier to move in this manner than large clusters.

2.3 Relative Placement

The motion of clusters is necessary for the correct placement of the clusters, but by itself, it isn't sufficient for their

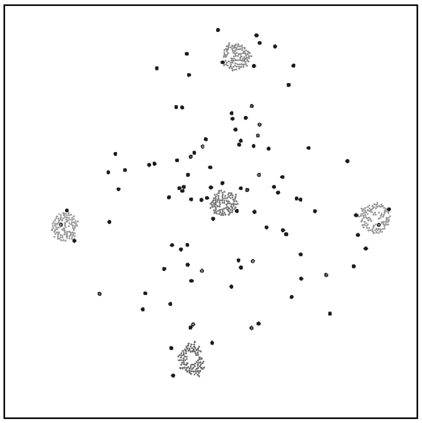

Figure 4. This figure illustrates the degenerate pentagon form. Clusters postions are not as desired, though they are at the final states.

correct placement. Not only must the clusters move, but their motion must be in a particular direction in order to be useful. The direction must be consistent with the generation of the desired design. That is, it must be possible for the individual autonomous agent to determine a direction that the cluster should be moving in, and then react accordingly.

This question is addressed in [10]. In that study, each agent is assumed to have access to accurate current positions of all of the clusters relative to the robot. The positions can then be used to calculate accurate current distances and directions between clusters. This information is then used to determine the direction of motion utilizing the following direction calculation:

$$\overrightarrow{\delta d} = 2\Sigma_{i \neq j} \widehat{u_{ij}} \left(d_{ij} - D_{ij} \right), \tag{6}$$

where $\overrightarrow{\delta d}$ represents the direction of motion, $\widehat{u_{ij}}$ represents the unit direction between clusters i and j, d_{ij} is the actual distance between clusters i and j, and D_{ij} is the desired distance between clusters i and j.

The effect of this is that any two clusters that are too close to one another will be pushed apart, while those too far apart will be pulled together. The aggregate effect of all of the other clusters thereby defines the direction of motion of any given cluster.

This methodology is capable of generating a movement that accurately places pairs and triplets of clusters relative to one another.

3 Cluster Placement with Minimal Cluster-Cluster Impedance

In swarm engineering, we determine minimal conditions required for the global outcome to occur and then design agent behavior that provably leads to the global outcome. In this case, the global outcome is the correct relative placement of the clusters. The minimal condition is the following: *each cluster moves along a path that has a finite length, ends at the desired relative position of the cluster, and cannot be impeded by constrains caused by other clusters.* This is a provable minimal condition in the sense that there is a finite time after which the cluster will be located at the end of the path, and this time will define the total design time. How these paths are generated, and how the clusters are moved along them is the subject of this study.

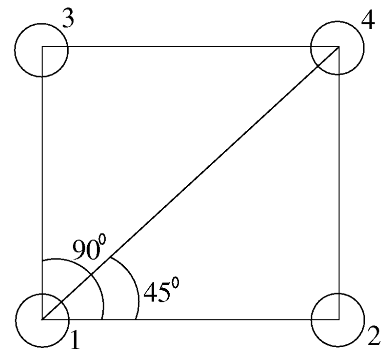

Figure 5. Placement of clusters in corners of square. Cluster 1 is “central” and clusters 2, 3, 4 are “secondary” clusters. Map provided to each agent includes distances between clusters 2, 3, 4 and “central” cluster 1, and angles $\angle 312$ and $\angle 412$.

First, we simplify our discussion by viewing each cluster as a single point. This point may be obtained by averaging the position of the individual pucks and using this centroid as the position of the cluster. We assume that the agents which are carrying out this action are capable of assessing the centroid of a group of pucks, a restriction on the capabilities of the agents. In realistic experiments, this may be accomplished using passive cameras and simple image processing.

Once the puck positions have been determined, each robot must determine how to bias the pick up and drop off probabilities so that the clusters are moved in the proper direction. We assume also that a map is provided to each of the robots in the form of a list of distances from a “central” cluster to each of the other clusters, along with the angle from a “secondary” cluster’s radial connector to the central cluster to that of each other cluster. That is, in order to position clusters we need to have data similar to those in a table below.

Table 1

Cluster Number	Distance	Angle
1	0	0
2	100	0
3	100	90
4	141	45

This table describes the relative placements of all clusters required to create a square arrangement of clusters as shown in Figure 5. Under an arrangement or placement of clusters we understand such positioning of clusters that they are placed in the corners of a defined geometric figure. Note that each cluster is initially randomly placed with respect to the others and has any absolute position in the plane. As a result, the mechanism requires that each cluster be carefully moved to specific relative positions via some means that are absolutely incapable of being restricted by other clusters.

Our method proceeds in two phases. In Phase I, the clusters are moved away from the "central" cluster to a distance commensurate with the desired distance. This may be accomplished via the movement protocol described in Section 2. The pick up probabilities are biased based on the robot's direction with respect to the ray directly connecting a given cluster and the "central" cluster. If the two clusters are too close, the upper threshold is altered according to

$$d = -\epsilon B cos\left(\phi\right), \tag{7}$$

where ϵ is a small positive constant, B is the current upper threshold, and ϕ is the angle between the robot's direction and the ray connecting the "central" cluster to the cluster in question. If the clusters are too far apart, this relation is negated, resulting in the cluster moving toward the "central" cluster. Note that the "central" cluster is not biased; the agents have identical interactions with it regardless of the direction of motion of the robot when interacting with the cluster. This means that although there will be a slow random walk, in general, the cluster will not move significantly compared to the deliberate motions of the other clusters.

In Phase II, the clusters are rotated around the central cluster, using the "secondary" cluster as a reference for measuring angles. The method assumes that the agents can measure the angles between the ray connecting the "central" cluster and the "secondary" cluster, and the ray connecting the "central" cluster and the cluster the agent is working with. Once the angle has been determined, the agent's interactions are biased in such a way that the induced direction of motion moves the cluster toward the desired angle by the shortest route. The bias is, as previously, done by altering the upper threshold. This means that the bias has the form

$$d = \pm\epsilon B sin\left(\phi\right), \tag{8}$$

rather than the previous one, with the sign depending on whether or not the angle between the desired location (relative to the "central" and "secondary" clusters) lies closer via a clockwise or counter-clockwise direction.

The described situation is depicted in Figure 6 for a square placement. Once four clusters formed, one of them was randomly picked to be "central" cluster. Other ones radially move away from it and then orbit to the desired positions.

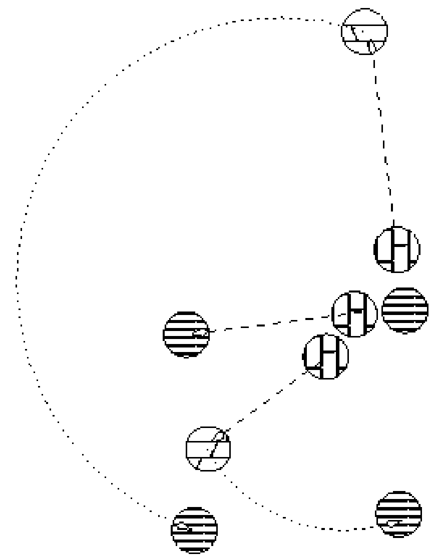

Figure 6. This figure illustrates the mechanism used to construct a square. First, three clusters are moved away from a "central" cluster. Then, two are moved around the central cluster, orbiting at the desired radius from the "central" cluster. The orbit ends at the final, desired, points, whose positions are measured against the central and first cluster.

4 Generation of Simple Geometric Forms

Let us now consider examples of right triangle, square and right pentagon arrangements of the clusters. Each cluster contains 80 pucks. All pucks are moved by 30 agents. Simulation starts from random placement of robots and pucks on $2D$ plane. Once the clusters are built, they are assigned unique numbers from 1 to N, where N is total number of clusters. Cluster 1 is a "central" cluster and it remains fixed during simulation. Other clusters move away from it on a desired distance first (radial movement) and then move away from each other (orbital movement) as shown on Figure 6.

The proposed method can be used to construct asymmetric as well as symmetric structures. In our simulations we build symmetric triangle, square and pentagon. This allows us to compare results with those Kazadi et. al. obtained in [10].

First, we consider right triangle clusters placement. During simulations, after clusters were built, "secondary" clusters 2 and 3 moved from cluster 1, then cluster 2 stopped and only cluster 3 continued its orbital movement away from cluster 2. Figure 7 illustrates creation of right triangle.

In the case of higher order placements such as square, pentagon etc. the problem of cluster collision avoidance should be taken into account. For the square placement it follows from Table 1 that during second phase of movement clusters 2 and 3 are moving on the same orbit which makes it possible for them to crush into one another. We

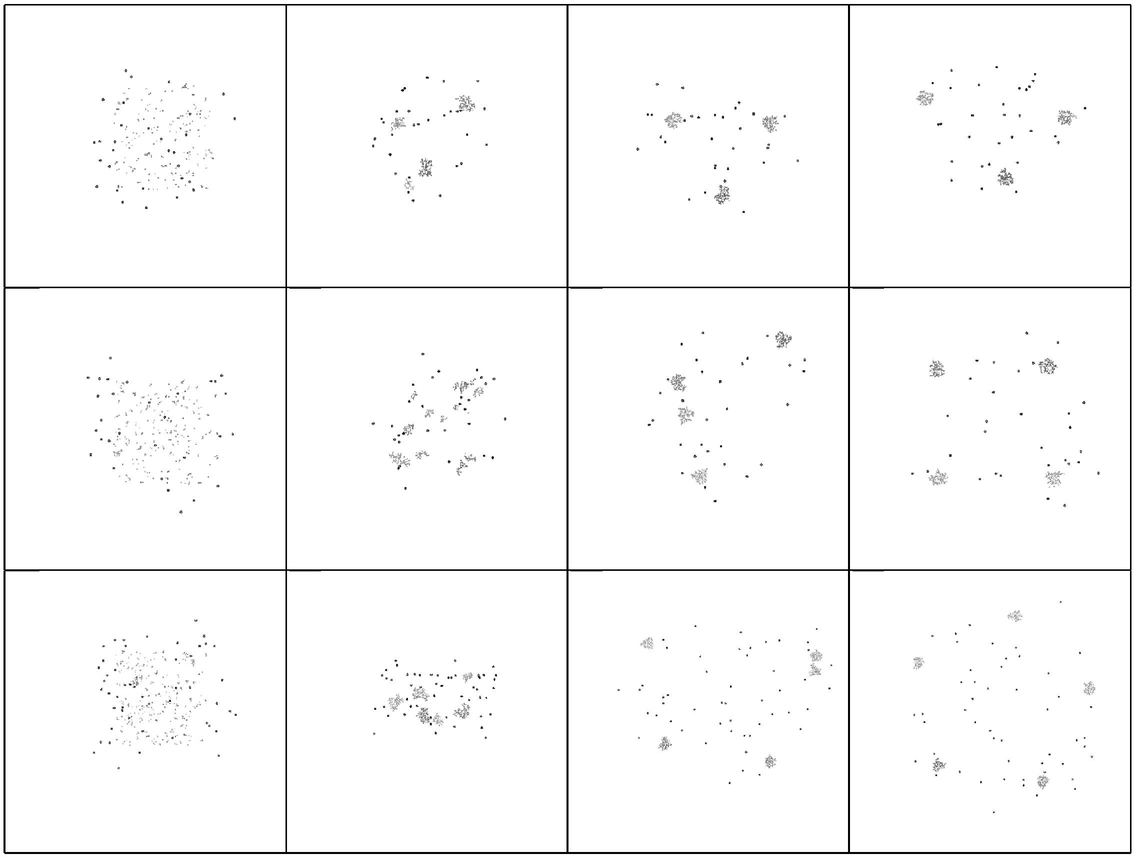

Figure 7. This figure illustrates building process of right triangle, square and pentagon.

encounter similar situation with pentagon arrangement and other high-order geometric placements.

For this purpose the following modifications were made. During radial phase of movement agents required to move clusters further than necessary and proceed to second phase. When the angle is as desired then clusters moved radially back to their orbit. This allows another clusters which are still in the second phase of movement to pass already placed clusters. In Figure 7 corresponding movement of clusters is shown.

Once the clusters are placed at the desired positions they stay at them with respect to the central "cluster". However, the clusters positions may deviate slightly. These deviations are additive in the sense that "central" cluster's deviations lead to deviations in other cluster's positions. In order to avoid them, agents forced to stop once information they measure coincides with desired information they were given *a priori*.

The most significant feature of proposed approach to clusters placement is that it allows to avoid degeneracy. Under degeneracies we understand such states when distances and angles between clusters are same or close to desired ones, but structure formed is different from required. This situation was described in [10] and is illustrated by Figure 4. The percentage of runs which developed correct structures in that paper is reported in table below.

Table 2

Structure	Correct Shape Percentage	
	Old Results	New Results
Equilateral Triangle	100%	100%
Square	66.667%	100%
Pentagon	12.5%	100%

This table shows that as the number of clusters increases, the number of degenerate states also increases. With the algorithm proposed in this paper all placements are non-degenerate. So, the methodology we have examined is capable of building clusters, and moving and placing them according to desired map.

5 Conclusions and future work

Historically, clustering became a basis for construction development. In order to build a structure it is necessary to obtain a drawing, or a map of this structure. Then the next step in actual building process is to mark important points such as corners of the building. After that it is possible to build walls, ceilings etc. The development of an accurate marking technique thus is a significant task.

This paper reviewed work that has been done previously on building clusters, and proposed algorithm for movement clusters such that they are placed on $2D$ plane as required by an engineer. It was shown that problem of degeneracy that wasn't solved in previous work, was overcame by moving clusters straight away from randomly chosen "central" cluster and then changing direction of their movement so that they "orbit" around center until reaching desired position. It should be noted that this position is given with respect to other clusters, particularly in this case with respect to "central" cluster. We don't take into account clusters position on a plane.

In case of large enough structures another problem arise. Since the agents are relatively small and their field of vision is bounded it might happen that they don't "see" whole construction cite, but only part of it. Robots need to identify to what part of construction they belong.

This sets a task for a future work in the area. It is necessary to design a method which allows agents to know whole map of construction and identify themselves on this map. Then once clusters of building materials positioned in required places, robots build walls between them and then ceilings on top. In other words, present results lead to building one story structures by swarm of minimalist robots.

References

[1] Beckers R., Holland O., and Deneubourg J. *From local actions to global tasks: Stigmergy and collective robotics*. H. Ritter, H. Cruse, & J. Dean (eds.), **Prerational intelligence: Adaptive behavior and intelligent systems without symbols and logic**, Vol. 2. Kluwer Academic Publishers, 2:549-563, 2000.

[2] Bonabeau E., Dorigo M., and Theraulaz G. *Swarm Intelligence: From Natural to Artificial Systems*. **SFI Studies in the Science of Complexity**. Oxford University Press, New York, NY, 1999.

[3] Deneubourg J., Goss S. *Collective Patterns and decision-making*. **Ethology, Ecology, Evolution**, 1:295-311, 1989.

[4] Holland O. and Melhuish C. *Stigmergy, self-organization, and sorting in collective robotics*. **Artificial Life**, 5(2): 173-202, 1999.

[5] Ijspeert A., Martinoli A., Billard A., and Gambardella L. *Collaboration through the Exploitation of Local Interactions in Autonomous Collective Robotics: The Stick Pulling Experiment*. **Autonomous Robots**, 11 (2): 149-171, 2001.

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

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

[8] Kazadi, S. *Position Control in Puck Clustering Systems*. **Proceedings of the Workshop on Agent and Swarm Programming, 2003**, Clevelend Ohio, September, 2003.

[9] Kazadi S., Chung M., Lee B., Cho R. *On the Dynamics on Clustering Systems*. **Robotics and Autonomous Systems**, (46), pp.1-27, 2004.

[10] Kazadi S., Grosz A., Lim A., Wigglesworth J., Vitullo D., *Swarm-Mediated Cluster-Based Construction*. Submitted to **Complex Systems**, 2004.

[11] Winfield A. and Holland O. *The application of wireless local area network technology to the control of mobile robots*. **Microprocessors and Microsystems**, 23(10): 597-607, 2000.