A Principled Approach to Swarm-Based Wall-Building

Lihan Lai, Jeff Manning, Jeannie Su, and Sanza Kazadi

Jisan Research Institute, Pasadena, CA 91107, USA

Abstract. In this paper, we apply a theoretical swarm-generating technique to a system implementing *cluster-based construction*. The technique, known as *swarm engineering* consists of two stages. In the first stage, which is top down, the global goal is expressed in such a way that specific conditions may be developed which, when satisfied, guarantees achievement of the global goal. The second step, which is bottom up, concerns the design of specific agents. These agents, once built in accordance with the conditions from the top-down step, will provably lead to the global goal. We develop parts of this theory and apply them to the cluster-based construction problem.

1 Introduction

Swarm-based systems have received an increasing amount of attention over the past few years owing to the remarkable abilities of swarms of agents to do things *en masse* that none of the individual agents can do. The phenomenon of *emergence*, in which agents carry out actions that they are not explicitly designed to do, has captured the imagination of many engineers and scientists and this has led to a flurry of work.

A number of researchers have explored the potential use of swarms to carry out construction tasks[11,12,13,14]. Much of this work has centered around trying to understand how animal systems accomplish this task and then to reproduce what animals have done. While many of these studies have produced interesting initial steps, few have actually made the transition from a proof of concept to a useful and competitive construction idea.

Our interest in swarms centers around the careful design of swarms using a reproducible methodology. To date, no standard technique exists which can be used to design swarms with specific global properties. As a result, most practitioners of swarm design must resort to using their own skill as engineers to build swarms of particular design. There is no guarantee that the desired swarm can be built at all. There is no way of knowing whether or not a particular behavior will yield the desired global behavior without running the task.

In this paper, we extend our previous work on puck clustering by applying the formal swarm engineering technique we call the ***Hamiltonian method of swarm design*** to the wall-building subproblem. This problem involves building walls between existing placed clusters. Once we've solved this problem from a theoretical standpoint, we apply it to a simulated swarm of agents.

M. Randall, H. Abbass, and J. Wiles (Eds.): ACAL 2007, LNAI 4828, pp. 306–320, 2007.

Though this paper deals only with artificial swarms, this methodology may be used to understand natural systems as well. Each natural system has global properties that must be satisfied in order to accomplish tasks that help keep it alive. Understanding the minimal requirements of that task will help the researcher identify agent behaviors that lead to specific swarm-level behaviors. Moreover, the identification of these behaviors helps to understand how the agents maintain the swarm and how each small behavior contributes to the global behavior of the swarm.

The paper proceeds in the following way. Section 2 reviews swarm-based construction work. Section 3 describes the swarm engineering methodology and its application to this problem. Section 4 gives simulation results. Finally, Section 5 offers some discussion and concluding remarks.

2 Previous Work in Swarm Construction

Swarm based construction deals specifically with the use of swarms of robots in construction. Unlike regular construction, the "workers" in this swarm based system do not use high-level reasoning. The swarm, however, does behave like the insects, exhibiting remarkably dynamic properties that make construction possible. Once developed, swarm based construction applications might range from assisting in civil disasters to remote construction problems in which human labor is impossible or dangerous; robots can be used to build replacement homes in areas struck by disaster, construct levee banks to restrain floodwaters, and build walls to retain chemical spills or nuclear radiation leaks. Swarms have also been envisioned as a solution to building underwater facilities and even structures in space. The main question in dealing with any of these problems lies in determining the role of the swarm, the various castes required, the interaction of the various castes, and the control algorithms for each of the agents in each of the castes.

It is interesting to look at the animal kingdom as a source of inspiration for the design of construction swarms. For instance, bulldozer ants have been observed to build nests by plowing material away from the nest site. These ants, behaving like little "bulldozers", ensure that the construction site is clear of rocks and other obstacles. Parker, Zhang, and Kube [12] explore this collective construction strategy, which they call "blind bulldozing". In their study, the robots, like the ants, carve a nest out of an excess material or rubble. The robots plow a nest by continuing to push the material until the robots cannot exert enough force to move the material. By pushing back the material, the robots build a circular wall structure that encircles the nested area. This version of site preparation can be implemented in the first step of construction. The robots that accomplish this task may be thought of as a specific caste of robots that perform the initial construction task, with subsequent castes required for further construction.

While the blind bulldozing methodology does not lead directly to methods for the construction of larger structures, the simplicity of the method is alluring. Not only is a single individual capable of accomplishing the entire task if given enough time, but other individuals can be added to the task seamlessly, without affecting the original agent's behavior or design. Moreover, additional individuals have a very small effect on the original individual as long as the number of new individuals is relatively small. As a result, the method is both scalable and robust to failure of a single individual. These qualities are desirable, and therefore should appear in the construction task.

This approach has already been used to construct complex structures using modified algorithms "borrowed" from the natural world. Ants can construct complex structures such as arches by first forming small piles of sand and adding onto these piles. Two piles are started near one another. Ants continually place bits of sand on top of the piles, making the overall piles grow upward. The piles grow up, and can bend over as they are buit up. When the piles are built up and curve inward towards one another, eventually meeting at the top. In [1], Bowyer implements a similar method to build arches and walls by adding blobs of polymer foam to piles of foam initially placed near one another. In this instantiation, the robots have legs, which allow them to climb the foam walls and build on the existing structure. The structures developed using this method in the laboratory resembled that found in the natural world, while illustrating one important aspect of the overall task design process - the morphological instantiation of the robot itself.

Our approach to construction is therefore based on this general approach. We are developing a methodology that can be accomplished with groups of individuals or with single individuals, with a graceful improvement in overall performance as new individuals are added to the task. On top of this, we would like predictability in our final results in the sense that we know within given tolerances what type of structure will emerge. This is very different from the requirements of a natural system, as a natural system will have differing overall structure depending on nuances of the environment, and therefore we depart from strict adherence to the designs of a natural world.

The general *puck clustering* methodology [9,10] satisfies the requirements outlined above. This methodology focuses on creating clusters of predetermined size and multiplicity. Special care is taken to provide theory that allows predictions as to the final state of the system to be made. This methodology, which has also been "borrowed" from the natural world, also has the beneficial characteristics described above. Namely, the clustering can be accomplished by a single individual, it benefits from the addition of new individuals, and the addition of new individuals is nearly invisible to the already existing individuals in the system.

Studies have also focused on the movement and correct placement of the clusters formed using the puck clustering methodology. These studies have demonstrated that it is possible to move clusters into predetermined relative positions with precise specificity. This means that a complex cluster grouping can be constructed using agents whose knowledge of the overall structure is extremely limited. However, the final arrangement of clusters is predictable and reasonable. Estimations can be made as to the completion time of such a task. Once again, these algorithms have the general properties described above - the potential for completion even with a single agent, and the seamless addition opportunity for new agents of similar design.

3 Swarm Engineering

Swarm engineering [8] is a method composed of two steps that generates agents and associated algorithms which accomplish predetermined tasks. The first step of swarm engineering consists of the creation of a swarm condition. It is the condition that, when satisfied, leads to the specific completion of the global goal. No specific method exists for this step and the creation of a swarm condition is problem-dependent. The second step is the creation of swarm behaviors that satisfy the given swarm condition.

This two-step process guarantees the achievement of the desired global behavior. However, no general set of techniques has been developed for designing swarms. Because the agents interact independently, they may exhibit unforeseen behaviors. Small deviations in the individual agents' behaviors may cause a large change in the overall system behavior. Therefore, much effort has been expended to devise a rigorous methodology to avoid this potential problem. One of these efforts is [8]; we review the major points from [9].

Our strategy is to begin with an examination of the desired property for the construction of walls using properties that the agent can sense. Once this property has been constructed, an examination of its dynamics will yield a requirement for the behaviors of the agents which causes the system's generation of the desired property.

Our theoretical work is validated using a simulation consisting of a two-dimensional "world" populated by inanimate and animate objects. The inanimate objects are the pucks, which are circular in shape and do not move. The animate objects are the agents. The agents can move around, avoid one another, walk on, pick up, and put down the pucks. Agents are also circular, primarily in order to simplify the simulation. Agents are larger than pucks, with the previous having a diameter of two units while the latter has a diameter of one unit. The simulation is depicted in Figure 1.

The rules of the simulations are meant to mimick conditions that occur in with real laboratory robots. Agents are embodied and so cannot pass over one another. Agents are equipped with collision sensors which allow them to avoid collisions. Moreover, their visual sensors are limited in range, though they've been made to allow a 360 degree visual field. The agents can sense other agents and pucks within four hundred (400) units and avoid collisions with Agents are limited in their ability to pick up or place down pucks, so as to simulate similar limitations for real agents equipped with grippers. Once pucks have been placed, they stay where they've been placed until another agent comes along and moves the puck. We examine the behaviour of our agents within an embodied virtual system.

Initially, the agents do not carry any pucks. The system contains clusters of pucks that have been placed to mark the endpoints of a wall in accordance with

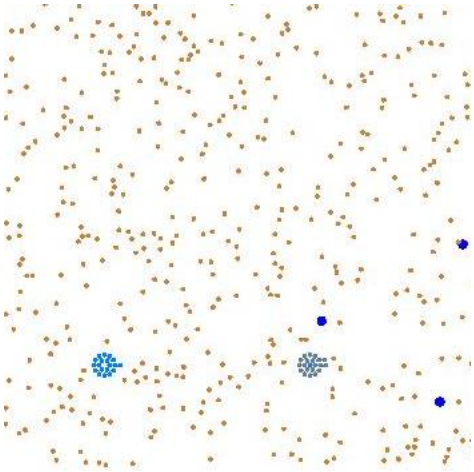


Fig. 1. This depicts the two-dimensional system. The small building materials are scattered about. Two large clusters used as construction markers are evident, as are three agents. The agents' behavior will move the building materials into the space between the two clusters.

the approach of [9,10]. Each of these endpoint clusters is made using pucks of a type that the agents have been programmed to ignore. This way the agents do not move the marking clusters. Pucks of a different type than any of the marking clusters are initially placed somewhere in the arena to be used as building materials. They are either randomly placed or placed in a single large pile, typically much larger than the marking clusters. These pucks may be picked up and put down as the structure being built is constructed.

The system allows agents to carry out construction task by moving pucks in some way from the starting location to other locations. The task is considered completed when all of the pucks that can be placed are placed in the desired region. In this system, we assume that all agents can walk over the pucks, but not over one-another.

3.1 Top Down

In [8], guidelines were generated for writing a global property. The first of these guidelines suggests that the global property be a function of local properties that are themselves accessible to a single agent. Once we write the global property, P_j, then differentiating gives the equation

$$b_j = \frac{dP_j}{dt} = \sum_{i=1}^{n_b} \frac{\partial P_j}{\partial b_i}\frac{db_i}{dt} + \sum_{i=1}^{n_P} \frac{\partial P_j}{\partial P_i} b_i \tag{1}$$

Where $P_1, P_2, \ldots, P_{n_P}$ are the properties of the system, and $b_1, b_2, \ldots, b_{n_b}$ are the behaviors of the system. This equation expresses the golbal property in terms of other, possibly simpler properties that make it up.

The global property for the current work is that the agents move building materials into a region between two already placed clusters. The material must be placed in such a way that it can be used as a foundation for the construction of a larger structure above it. Because this is meant to be a foundation for a larger structure, conditions must be generated which allow for the generation of a foundation with a minimal number of gaps between the building blocks. We assume that the building blocks are all identical. We also assume that they can be placed on top of one another if needed.

We are interested in a system built using agents of minimal complexity. This avoids the potential complications that might arise from the use of sophisticated agents with complex parts and behaviors. As a result, our system utilizes very simple agents which we expect to be able to complete the task in tandem. As a result, we focus on the local actions of the agents. Suppose that an agent encounters a piece of building material. Should the agent pick it up and move it, or should it leave the material where it is? In general, one might expect it to be left where it is unless it is not in the desired region.

It is realistic to expect that a single agent built with current technology on board can tell whether or not an encountered puck is in the desired region. It is also realistic to expect that the agents can determine the range and direction to the nearest region. Thus, our microscopic property can realistically be the minimal distance a puck has to the desired region. This is, of course, zero if the puck is in the desired region.

As a result of this, we can write the global property as

$$P_G = \sum_{p \in \{\text{pucks}\}} d_p. \tag{2}$$

where d_p is the distance the individual puck is from the desired wall region. Differentiating this equation, we obtain

$$b_g = \sum_{p \in \{\text{pucks}\}} b_p. \tag{3}$$

This equation illustrates the idea that the global behavior is a function of the behaviors of the individual pucks. Moreover, these behaviors are functionally independent (they interact with a '+' sign). They can therefore happen without directly interacting agents. Since the behavior of the individual pucks is controlled through the agent behaviors, we have a natural method of manipulating the global property.

Let us examine what the end points of the global property ought to be. Initially, the property has some value determined by the organization of the set of pucks. This value must be reduced by at least

$$\delta P_g = \sum_{p \in \{\text{M closest pucks}\}} d_p. \tag{4}$$

The reduction of δP_g guarantees that the correct layout of pucks has occurred. Such a layout will be guaranteed if at least M pucks are moved into the region from the exterior region. Thus, we now have a general condition which, when satisfied, will give us the global goal. The next task is to create a behavior that is capable of producing it.

It is perhaps not easy to see how this analysis *guarantees* that the global goal will be met. However, the property so chosen has a unique numerical value at the system configuration desired. The task then, is to find out how to generate this specific numerical value. The specific agents required to achieve this numerical value may be chosen from any number of potential agents whose affect on this property is to move its numerical value from the current system configuration to the desired one.

We also assume, for the moment that the agents need only mark the foundation of the structure being built. More than one property is required for three-dimensional construction, and that is beyond the scope of this paper.

3.2 Bottom Up

Now that a top down condition has been created, we can turn to the bottom up portion of the swarm engineering methodology. Our task is to create a behavior which provably completes the condition from the top-down portion. Thus, this subsection will focus on the development of a behavior set for the individual agents that accomplishes the task in a provable way. This allows the swarm to be designed robustly whilst sidestepping much of the complexity associated with the interactions between agents.

In this study, we've created a global property P_G, which is the sum of all the distances from each puck to the region where the wall is to be built. To manipulate P_G the robots need to have a mechanism for sensing, picking up, transporting, and dropping off pucks. Thus, a basic outline of local behaviors for P_G to reach the desired state is to pick up pucks that are not in the region, move them to the region, and drop them off at an appropriate location within the region. It is necessary that the robots know if they are in the region, in order to ensure that the pucks are dropped off at the right spot. This requires the robots to know where to go if they are not in the region. Many robots are present and moving at the same time in the system. Collisions are extremely likely, particularly as the number of robots increases. In order to avoid collisions, robots must have the ability to sense one another and use this information to robustly avoid collisions.

This gives an idea of what kind of hardware one might need. Clearly some type of sensor array is required which will provide this information along with general obstacle proximity data. Processing is required to determine what direction to go based on sensor data. How much processing is required is not clear, as it will depend on the precise method used to determine the direction. Actuators include both movement mechanisms (i.e. legs, wheels, propellers, floatation devices, etc.) and grippers for grasping and holding the pucks.

Now that we have an idea of what the various requirements of the agents' hardware are, we can proceed to build the behaviors. Note that we can choose any behaviors *as long as they complete the global task.* Thus, the approach allows us to have the same kind of creativeness while guaranteeing that the outcome will be as desired.

The agents in our system have two states: those that are carrying pucks and those that are not. The way each agent moves depends on which of these states it has. Moreover, we assume that the agents are aware of where they are with respect to the construction region (the region between the demarkating clusters). The agents' control algorithm, which utilizes both of these pieces of information, is summarized in the flowchart in Figure 2.

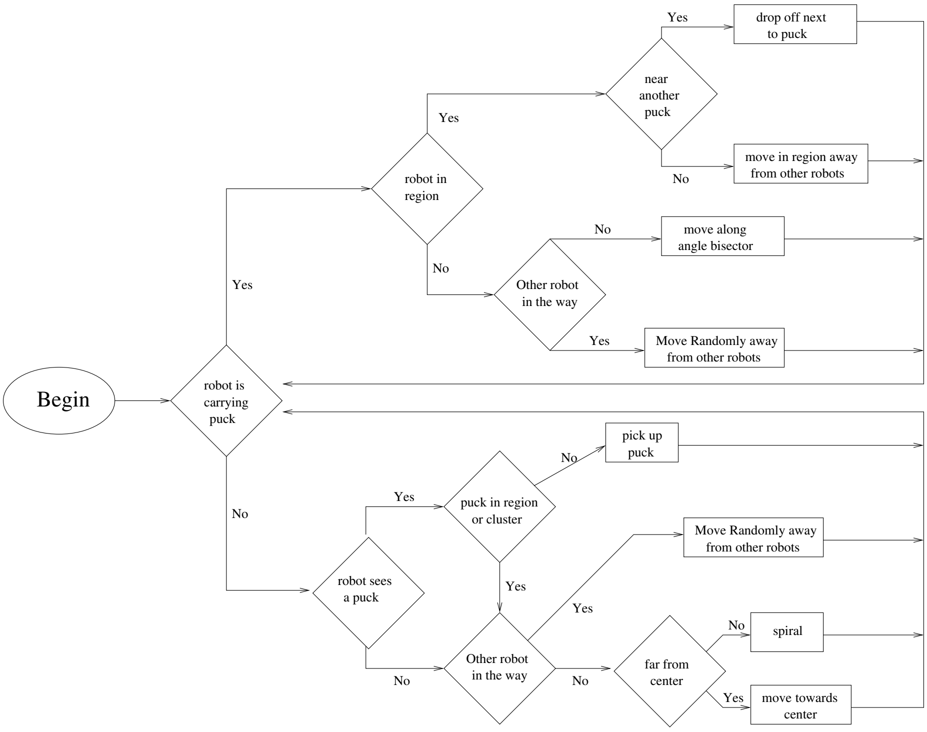


Fig. 2. This flowchart gives an outline of the agent's behavior

If an agent is carrying a puck it will first determine if it is in the region. To do this it first calculates the directions of the tangent segments from itself to each of the clusters. It then calculates the angles between each of the four pairs of

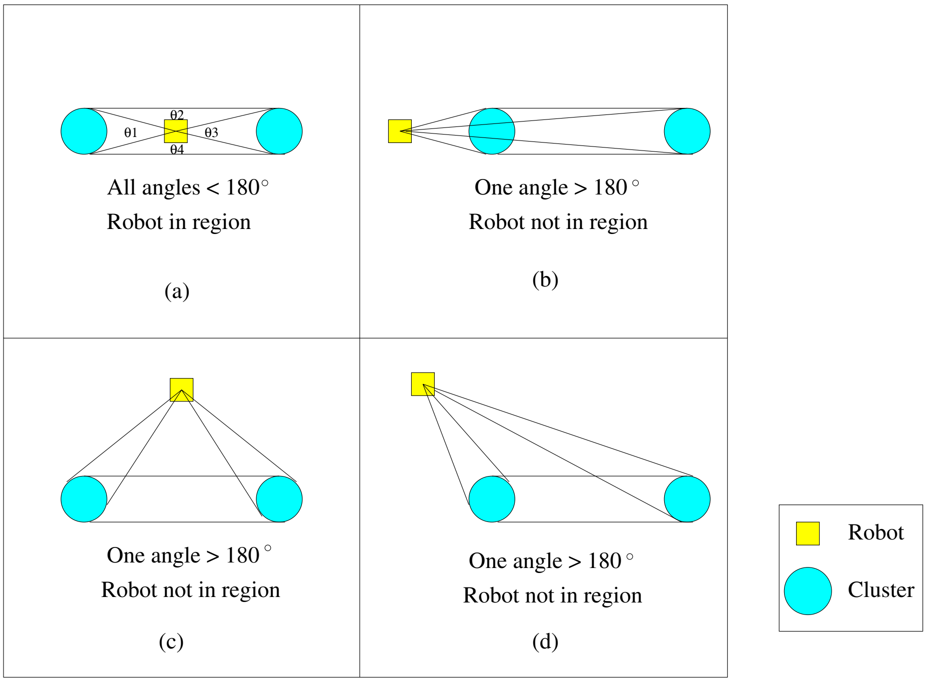


Fig. 3. These figures show different situations in which the agent is either in or out of the region. In (a), the agent is in the region because the all the angles, θ_i, are less than 180^o. In (b), (c), and (d), the agent is not in the region because the at least one θ_i is greater than 180^o so the agent is not in the region.

adjacent tangent lines, as shown in Figure 3. If all four of these angles are less than 180^o, then the agent is in the region; otherwise, it is not.

When a puck-carrying agent determines it is in the region, it will check whether it is near another puck inside the region. If it is, it will drop off its puck next to that puck. If it is not near another puck, it will move a short distance in a random direction.

If a puck-carrying agent is not in the region, it will determine the directions from itself to each of the two wall-demarcating clusters. After this, it will calculate the angle bisector of these two rays. If there are no nearby agents in that direction, it will move a short distance along the angle bisector, as shown in Figure 4. If there is a nearby agent in that direction, it will move in a random direction to avoid a collision.

This way, at each iteration before the agent reaches the region, the agent will re-calculate the angle bisector as before; however, because the locations of the clusters have changed slightly relative to the agent, the agents' directions vary slightly. Thus the agents move along a curve, whose endpoint is always a point on the wall. Because each individual angle bisector intersects the wall, the agent

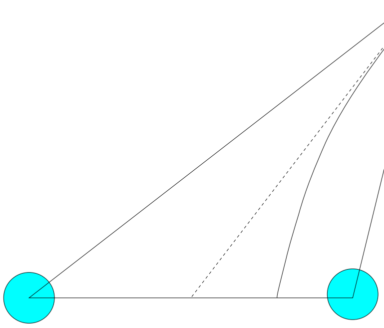


Fig. 4. This shows the pathway that the agent follows. Every iteration, the agent calculates a different angle bisector, as it moves to a new location, resulting in a curved path. The dash line is the angle bisector that agent calculates at the current location.

always move towards the region, as shown in Figure 4. The point along the wall that the agent reaches first depends on the the original location of the puck. The movement along the angle bisectors ensures that the pucks will be transferred to the region.

If the agent is not carrying a puck, it will determine if there is a nearby puck it can pick up. If so and if the puck is not in the region or in one of the clusters it will pick up the puck. If no puck is located nearby the agent which is not in the region or in one of the intial clusters, then the agent will spiral to search for a puck. We say that an agent is lost if it is more than 150 units away from a central location, which in our simulation is defined as the average of the positions of the clusters. Once an agent is lost it will continue to be lost until it is less than 20 units from the central location. If the agent is not lost it will spiral. If the agent is lost it will move toward the central location, this ensures that agents do not just keep spiraling forever and leave the construction area if they do not encounter a puck. In either case if the agent cannot move in the specified manner without hitting another agent, then it will move in a random direction which is not toward any nearby agents.

It is perhaps very easy to see that the effect of the behavior of the agents will be to wander around the arena until they come into contact with the pucks. Once these pucks are picked up, they will be carried by the agents into the building region, and be dropped off at whatever available location the agent carrying them can find. Mathematically, this means that

$$\delta P_g = \sum_{p \in \{\text{M pucks}\}} d_p. \tag{5}$$

which is at least as large a change as that given in (4). Thus, we can be sure that this behavioral set will accomplish the task.

4 Different Geometric Shapes

In the previous section, we discussed how to build single walls. To build rooms with more than one wall, a mechanism must be developed to build different geometric structures. We extend the previously discussed theory to create a multi-walled structure.

To build a multi-walled structure, it is assumed that n clusters, rather than two, have been placed to mark the endpoints of the various walls. Our goal is to build walls between certain specified pairs of clusters, but not other pairs of clusters. This means that there will be multiple construction regions (one for each desired wall), instead of just one. As before, it is realistic to assume that a single agent can determine whether or not an encounted puck is in a region, and that a single agent can calculate the range and direction to any region. So we realisitically make the microscopic property be the distance a puck is from the nearest region. We may write the global property as,

$$P_G = \sum_{p \in \{\text{pucks}\}} d_p \tag{6}$$

where d_p is the distance from p to the desired region. The initial value of P_G is determined by the initial arrangment of the pucks. This must decrease by at least

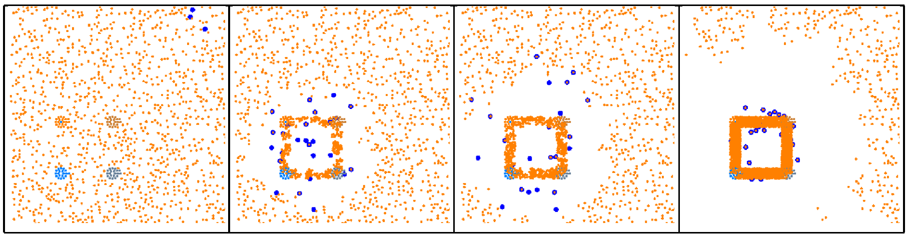
$$\delta P_g = \sum_{p \in \{\text{M pucks}\}} d_p . \tag{7}$$

where the M pucks are the minimal number of pucks, and the closest ones, required to fill the various regions without unabmiguously.

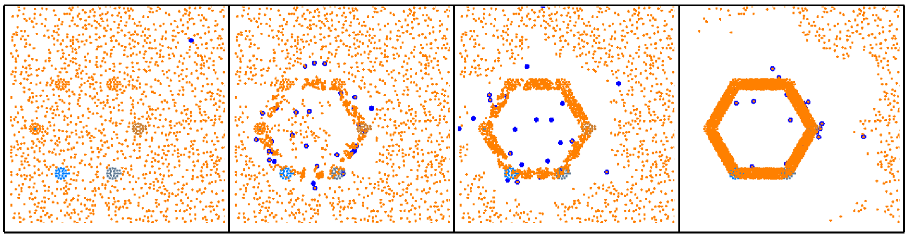
It should be clear that the agents in this new system must have all the behaviors described in section 3. In addition the agents must be able to determine if a given pair of clusters are required to have a wall between them to ensure that walls are only built in the specified locations. Also, when carrying pucks, agents must be able to choose a specific region, determine its direction, and move towards it.

In our simulation, the pucks in each of these clusters are different. In other words, cluster 1 consists of puck type 1, cluster 2 consists of puck type 2, cluster 3 consists of puck type 3, and so on. The use of differing puck types ensures that the agents do not destroy the clusters and remove the markers by using the pucks in the clusters as building material. A different set of pucks than those in the clusters is used to build the walls.

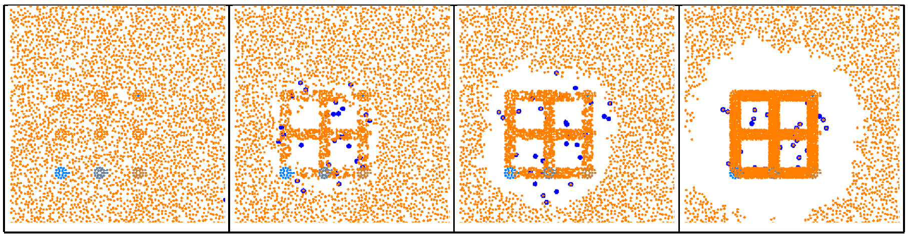
As an example, the square structure from Figure 5a has four clusters. This means that there are six possible walls - the sides and the diagonals. If all the marking clusters were made up of pucks of the same type, the agents would not have been able to differentiate between them and there would be no way to determine which pairs of clusters to build walls between. In this case, all six walls would be built, resulting in a square with a cross inside. Since this final shape is not what we desire in this case, the new method of using multiple cluster puck types is the obvious choice.



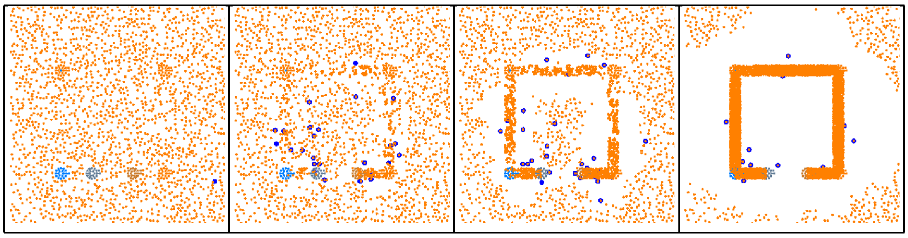
(a) square



(b) hexagon



(c) square divided in 4 squares



(d) This structure can be used as a foundation for buildings with doors.

Fig. 5. The figures above are screen-shots of different steps in our simulation. As the first column illustrates, our simulation is initialized with arbitrarily scattered pucks and specifically placed clusters that mark off the edges of the wall foundation. Each consecutive column is a snap-shot of what the system looked like after a certain amount of time passed. The last column is the finalized structure.

In these simulations, we utilize agents which have identical physical properties to those of the agents used int he previous simulations. However, each of the new agents is provided with a list indicating between which pairs of clusters to drop off the pucks, or in other words where to build the walls.

To implement this new global property, the same local behaviors used to build a one-walled structure are maintained. The agent's behavior after picking up a puck, however, is modified so that it does not move along the angle bisector of the angle between the clusters and itself. There are too many clusters, which complicate this behavior. Instead, the agent randomly chooses one wall to move toward and calculates its direction from that wall as if it were in a single-walled system.

The structures that agents may build in this way can be quite complex. Not only can the structures have many walls, but they can be designed to be practical. As an example, consider the agents laying out the floor plan of a house (Figure 6). The mechanisms described here can be used to carry out practical construction, a natural extension of the work in this paper.

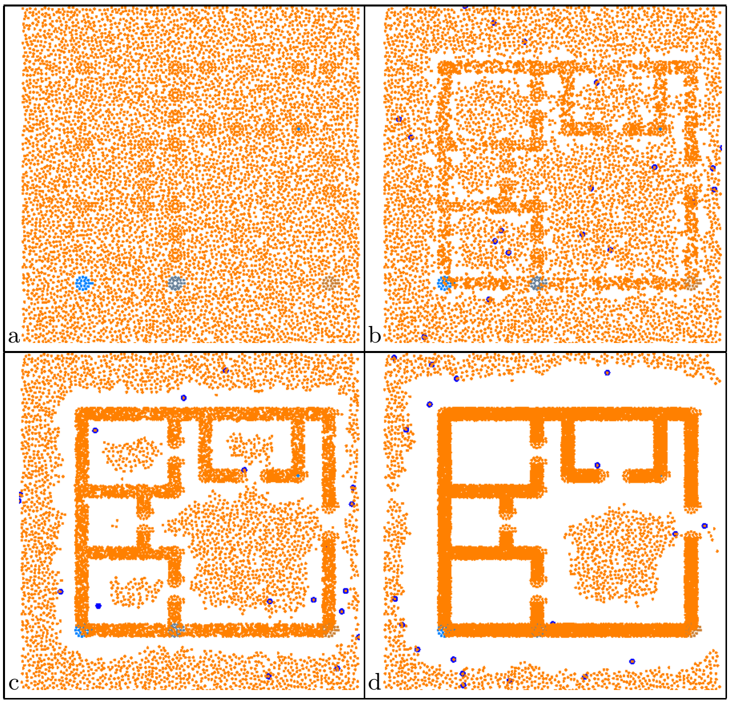


Fig. 6. This algorithm can be used to make a foundation for the outline of a typical house with two bedrooms, a kitchen, and a bathroom

5 Summary and Concluding Remarks

One of the great hopes of swarm systems in the early years was the development of swarms capable of completing construction of large complex structures completely without the need for external intervention [3][11][12][13][14]. Much of this work is based on observations and adaptations of existing systems in the natural world [2] [3][4][5][6][7]. Work continues on this to this day, but has yet to produce a satisfactory solution.

Perhaps one of the reasons this has been so is that most construction swarms have been able to do one or two steps of the whole task, but have not been able to handle the complexity of a true construction task. This may, in turn, be due to the paucity of a theoretical or principalled mechanisms for swarm design. It is extremely difficult to design a swarm of agents that completes the task it is supposed to be completing because the unintended interactions between the agents in the swarm are extremely likely to cause trouble in generating the correct global behavior. In fact, there is no prior theoretical work done that indicates a mechanism for predicting the global effect of particular local behaviors.

In this paper, we've created a method for generating swarms based on an examination of the meaning of the equations that describe the evolution of properties according to their associated behaviors. This is a very powerful exploration, as it affords us the ability to turn the tables on swarm generation. Rather than generating a swarm based on the "guess and check" method during which the individual agent is designed and then examined in simulations or real embodied fabrications, we've designed a method that allows us to say ahead of time what the actual agents will do, individually and in a group. Once these agents are built, we know that the global goal will be able to be achieved, as the interactions are implicitly taken into account.

The remainder of the paper was devoted to demonstrating how the technique could be applied to the swarm-based construction problem. The sensor requirements and mobility requirements as well as behaviors could be shown, then, to satisfy the numerical requirements of the theoretical approach. As a result, the property P reached the desired value, and the fact that it would reach this value meant that the system would converge to the desired state eventually. In other words, it was possible to predict the behavior of the system before the agents themselves had been constructed.

The problem we've examined is part of a larger problem in swarm-based construction. This larger problem can be explored by creating similar properties whose values are unique, given the specific stages that the system is in. As we've seen, as long as this numerical value is specific to the desired state, and the behaviors are designed to generate that desired numerical value, the system will take on the desired states. This is the next stage of this line of research.

In artificial life research, much of the discussion has historically centered around the question of what constitutes a lifelike system. In this particular discussion, we might also ask, what properties of a lifelike system can we expect and how might we generate them? Moreover, we can approach this by utilizing the sensor, processor, and actuator capabilities of the agents to generate the global property. Then, the requirements for the agent behaviors can be constructed. Such an approach might reliably generate agents whose behaviors make the system act more like a living system as a result of generating behaviors of the agents that create the desired system behaviors.

Anecdotally, we found that our method, when properly applied, trimmed the design time from several weeks of trial and error to several minutes of careful planning. If this is the improvement from a single stage of design, one might expect several to have a far greater improvement, possibly making a task that is currently impossible possible. More work is needed to determine whether or not this is so.

References

1. Bowyer, A.: Automated construction using co-operating biomimetic robots. University of Bath Department of Mechanical Engineering Technical Report (November 2000)
2. Bonabeau, E., Theraulaz, G., Fourcassie, V., Deneubourg, J.: Phase ordering kinetics of cemetary organization in ants. Physical Review E 57(4) (1998)
3. Camazine, S., Sneyd, J., Jenkins, M., Murray, D.: mathematical model of self-organized pattern formation on the combs of honeybee colonies. Journal of Theoretical Biology 147, 553–571 (1990)
4. Deneubourg, J., Sendova-Franks, G.S.F.N.A., Chretien, D.C.L.: The dynamics of collective sorting. Robot-like ants and ant-like robots. In: Meyer, J.A., Wilson, S.W. (eds.) Animals to Animats, MIT Press / Bradford Books, Cambridge, MA (1991)
5. Franks, N.R., Wilby, A., Silverman, B.W., Tofts, C.: Self-organizing nest construction in ants: sophisticated building by blind bulldozing. Anim. Behav. 44, 357–375 (1992)
6. Holland, O.: The use of social insect based control methods with multiple robot systems. In: UKACC International Conference on Control 1996 (1996)
7. Holldobler, B., Wilson, E.: The Ants. The Belknap Press of Harvard University Press, Cambridge Massachusetts (1990)
8. Kazadi, S.: On the Development of a Swarm Engineering Methodology. In: Proceedings of IEEE Conference on Systems, Man, and Cybernetics, Waikoloa, Hawaii, USA, pp. 1423–1428 (October 2005)
9. Kazadi, S., Grosz, A., Lim, A., Wigglesworth, J., Vitullo, D.: Swarm-Mediated Cluster-Based Constuction. Complex Systems 15(2), 157–181 (2004)
10. Kazadi, S., Koroleva, O.: Removing Degeneracy From Swarm Mediated Cluster-Based Construction. In: Proceedings of IASTED International Conference Robotics and Applications, Honolulu, Hawaii, USA, pp. 60–66 (August 2004)
11. Leung, H., Kothari, R., Minani, A.: Phase-Transition in a Swarm Algorithm for Self-Organized Construction. Physical Review E 68, 046111-9
12. Parker, C.A., Zhang, H., Kube, C.R.: Blind Bulldozing: Multiple Robot Nest Construction. In: Proc. IROS 2003, Las Vegas, USA (2003)
13. Stewart, R.L., Russell, R.A.: Building a loose wall structure with a robotic swarm using a spatio-temporal varying template. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2004)
14. Wawerla, J., Sukhatme, G.S., Mataric, M.J.: Collective Construction with Multiple Robots. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2002)