Reliable Swarm Design

S. Kazadi, A. Wen, M. Volodarsky
Jisan Research Institute
515 S. Palm Ave., Unit 3
Alhambra, CA 91803

***Abstract*—This paper describes how phase space diagrams may be used to predict the final states and system evolution of swarms. Phase space diagrams and system evolutions through phase space are constructed for systems of multiple and single attractors. While the diagrams indicate the number of attractors, the system evolutions confirm them. A method is described for generating *stochastic differential equations*, which can then be used to construct phase space diagrams. This method is applied to the TSP system update equations, generating a stochastic differential equation whose dynamics provide insight into the TSP system. The use of ensemble system updates transforms systems from directed random walk systems to systems governed by stochastic differential equations. We demonstrate the use on the puck clustering and TSP problems, generating improved reliability and quick evolution of the system.**

***Index Terms*—swarm engineering, swarm control**

I. Introduction

Engineering is generally about building systems that behave in predictable ways and carry out well understood functions. Many different considerations go into developing real systems including functionality, robustness, durability, lifetime, and generality. What makes swarm systems interesting is primarily the fact that many swarm systems are emergent, and that the capabilities of the swarm are greater than the capabilities of a single agent.

Several researchers have considered carefully the question of how to generate swarms of predetermined desired characteristics [3], [4], [1]. These methods have the common quality of working to develop methods of not only determining what the swarm will do *a priori*, but also being able to estimate parameters of the swarm, such as reliability, robustness under various perturbations, speed of achieving the final state, and quality (or limiting the possibility that the swarm will do something *undesired*). Kazadi's method [3] involves a two-part approach in which the global goals for the swarm are mathematically developed in terms of the basic behaviors of the agents. A second part then uses a condition derived from these equations to develop group behaviors. This has the advantage of being able to determine whether or not the swarm will generate the global goal *without* the need of building it. Spears' method [4] involves utilizing many different agents whose behaviors are adapted from physical laws (r^{-2} laws and perturbations of them) to control the agents. Sensor data of range and speed of local neighbors alone is used to determine the agents' behaviors. The advantage of this method is that it allows extremely simple sensory apparati and control algorithms and provides a robustness that is similar to that of chemical compound formation in the real world. Winfield [2] has examined the problem of determining characteristics using *temporal logic*, a formal mathematical approach. The approach has the advantage of providing a rich theoretical background from which to generate results about what the swarm is going to do.

In this paper, we examine a method of improving the performance of swarms. We begin by utilizing the method of Kazadi et. al. [6] to examine two swarm systems. The evolution of these systems through phase space follows the expected trajectories indicated by the phase space diagrams. We then examine a method of sharpening the phase space trajectories. The method utilizes ensemble updates and may be used to transform swarm update rules to differential equations. These differential equations may be used to generate phase space diagrams and to generate trajectories through these phase spaces. We apply this to two problems common to swarm work.

II. Phase spaces of clustering systems

Many fields use phase spaces as useful tools for understanding what a system will do. Phase spaces are generally defined as n-dimensional vector spaces in which each point is an n-tuple with each component representing a numerical value for one degree of freedom. Each point thereby represents the system in one configuration. Dynamic changes in the system amount to movement of the system through the phase space.

Of particular interest in swarm systems is the development of an understanding of where the attractors are, what kinds they are (points, cycles, etc.), and how to find them. Swarm systems are considered to be well engineered if they have a class of well known attractors which correspond to desired final states to which the system will evolve from any of the realistic starting configurations of the swarm. Therefore, our approach is to examine the phase space of the problem under the behavior of the various agents.

Systems that utilize the Hamiltonian Method of Swarm Engineering [9], [10] have behaviors that are directly connected to the global variables in question. That is, all global properties P_j are functions of the local properties and behaviors. I.e.,

$$P_j = f_j\left(x_1, , x_{N_l}, \dot{x_1}, , \dot{x_{N_l}}\right). \tag{1}$$

Therefore, the effect of the properties and behaviors of the agents may be determined as

$$\frac{dP_j}{dt} = \frac{df_j\left(x_1, , x_{N_l}, \dot{x_1}, , \dot{x_{N_l}}\right)}{dt}. \tag{2}$$

We use this to determine the behavior of the agents when graphing them as a function of system properties.

One of the simplest swarm systems is the puck clustering system. This system consists of agents and inanimate objects, known as pucks. The inanimate objects may be picked up, carried, placed down, and sensed by the agents in the system. The agents are assumed to be embodied, able to move around at will, able to carry pucks, and fully autonomous. Several flavors of clustering exist in which the agents may climb up on the pucks or not, may communicate or not, and may carry multiple pucks or not.

In puck clustering systems, the goal of the system is to put all of the pucks together into one cluster. Whether or not the agents are intelligent, a simple requirement based on the probability of an agent to pick up a puck from a cluster f (which is a function of the cluster size), and the probability of putting down a puck h next to a cluster, adding to the cluster (which is also a function of cluster size), must be satisfied. The condition is that the ratio of these two probabilities g is a decreasing function of cluster size, and the condition holds in every kind of swarm. In what follows, we assume that our clustering system employs agents which implement behaviors that obey these statistics. As a result, we can expect that in our clustering systems, the entire set of pucks will converge into a single cluster.

One great way of visualizing the system we've outlined is using a phase space diagram. We can visualize the current state of the system by using a Euclidean system in which each axis corresponds to the size of a single cluster. For instance, consider the system in Figure 2.1. This is a two-dimensional system in which we graph the size of two clusters in a system of N pucks. We assume also that there is a third cluster, which is not graphed. In this case, we have a feasible triangular region in which the system resides. The question is, what path will the system take when in this region?

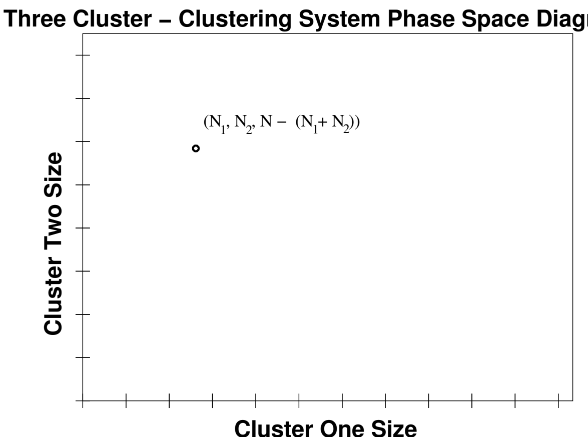


Figure 2.1: This figure illustrates the phase space of a three cluster system (one is in the z axis, which isn't pictured) under the action of clustering agents. Each point represents a system configuration. The configuration pictured has cluster one with size N_1, cluster two with size N_2, and cluster three with size $N - N_1 - N_2$, where N is the total number of pucks.

In order to answer this system, we examine the behaviors of the agents. We have noted that the condition for clustering is that $g(N) = \frac{f(N)}{g(N)}$ is a decreasing function of cluster size. This means that the probability of a puck being moved from a smaller cluster to a larger cluster exceeds that of puck being moved from a larger cluster to a smaller cluster. As a result, the system in Figure 2.1 has the phase-space flow diagram given in Figure 2.2.

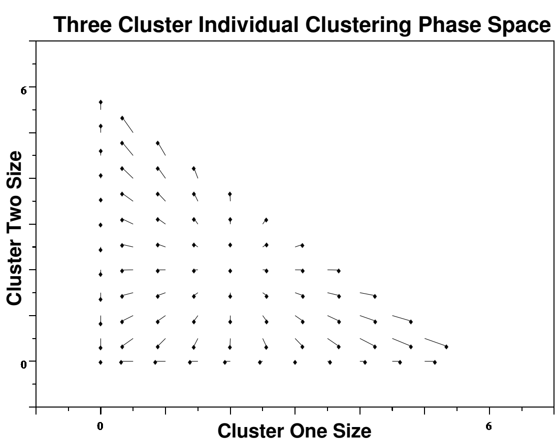
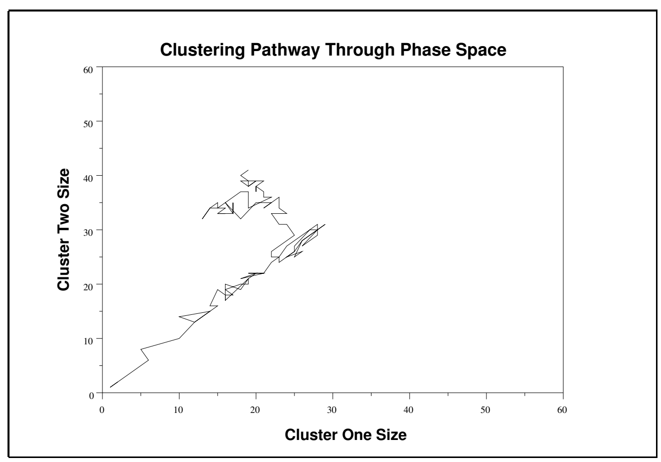
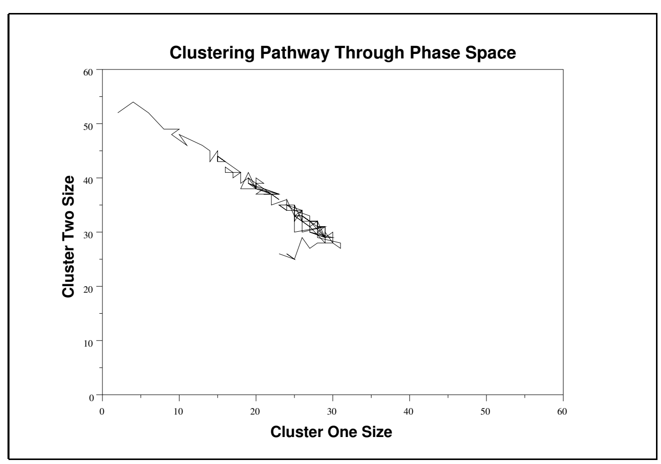


Figure 2.2: This figure illustrates the flow of a three cluster system (one is in the z axis, which isn't pictured) under the action of clustering agents. The system has three attractors at the corners of the region. The three attractors correspond to the states where the individual clusters contain all the pucks, and the other two have none.

This flow diagram illustrates how the space is partitioned into three regions, each of which move the system toward a specific point, which is the system attractor. These attractors occur at points where the clusters are all sized zero excepting a single cluster that holds all the pucks. As a result, we expect that the final state for this system will be one of three states in which one of the clusters will have all of the pucks and the others will have disappeared. Indeed, this is what happens.

The arrows in Figure 2.2 illustrate the *flow* of the swarm system. That is, they illustrate where the dynamics of the system tend to move the system in phase space. The path that a system takes in phase space is typically a randomized walk in the general direction of the flow of the system in phase space. We ran the clustering system and graphed several paths through phase space in Figure 2.3. As expected, they tended to follow the flow of the swarm system.





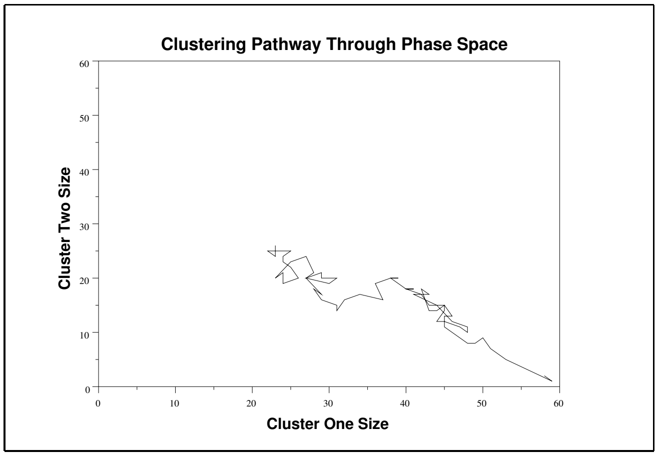


Figure 2.3: The direction of the population from beginning to end follows the expected direction, and ends at the expected attractors when implemented in simulation. Details of the simulations are given in [3].

III. Embedded behaviors

Many systems whose design methodologies do not follow the Hamiltonian Method of Swarm Design [9], [10] are difficult to design. This is because, since there is no direct connection between the behaviors of the agents and the system measurables; the effect of a specific behavior on the system measurable is a complicated function of many actions. In the event that the individual behaviors are stochastic, it is difficult to give any kind of prediction about the effect of a specific behavior on the global behavior of the system.

In order to determine the behavior of a specific action on the system, it is useful to try to mitigate the stochastic nature of the action. This cannot be done, in general, with individual disconnected agents. However, if the agents behave in an ensemble, this tends to reduce the stochastic nature of system changes, and sharpens the dynamics of the system so that repeated implementations of the system from the same or similar initial states tend to generate the same final system.

We have shown elsewhere [11] that the following theorem generally holds for swarm systems.

Theorem 1: Given a stochastic stigmergic swarm whose actions alter a system property known as e. If the agents' behaviors are a function of e then a swarm whose system property updates are done as a large ensemble evolve according to the equation

$$\frac{de}{dt} = \sum_{i=1}^{N} \gamma_i\left(e\right) p_i\left(e\right) \tag{3}$$

where $\{\gamma_i\}_{i=1}^{N}$ is a set of potential outcomes for the system behavior and $\{p_i\}_{i=1}^{N}$ is a corresponding set of probabilities that the event γ_i will occur.

A. Sharpening direct action phase space diagrams

We illustrate the effect of utilizing ensembles using a modified version of the puck clustering system described in Section 2. The new puck clustering system is identical to that of Section 2 with the exception of the use of *active pucks*. These pucks, as described above, interact with the agents as the agents act to create clusters.

We add two memory elements to the pucks, which may be queried, changed, and reset by the agents. One of these memories elements is a pick-up memory element, capable of storing nonnegative integers. The second is the drop-off memory element, also capable of storing nonnegative integers. As each agent approaches a puck and makes a decision to pick it up, it must first query the pick-up memory element of that puck. If the memory element is below some user-defined threshold, it is incremented and no other action is taken. On the other hand, if it is at or above the threshold, the puck will be picked up and the counter will be reset. Likewise, if an agent decides that it will drop off a puck next to a given puck that is already on the ground, it must first query the drop off counter of the puck on the ground. If the counter is at or above the threshold, the carried puck will be placed next to the puck on the ground. The drop-off counter of the puck on the ground will be reset to zero.

The effect of the behavior is that there is an ensemble decision about whether or not to pick up pucks. Many agents must agree that the puck in question should be picked up in order for it to be picked up. Likewise, many agents must agree that placing a puck down next to a specific other puck is acceptable before any puck can be placed down. As a result, the probability of making a decision that will lead the system away from the region's attractor is significantly reduced. The dynamic phase space diagram is also changed, with the apparent direction of evolution of the system significantly more directed toward the attractor. The application of this rule set in the three runs given in Figure 3.1 demonstrates the much more directed system evolution, as does the reduced number of pick-ups and drop-offs. Figure 3.2 illustrates the much more directed system trajectories through phase space.

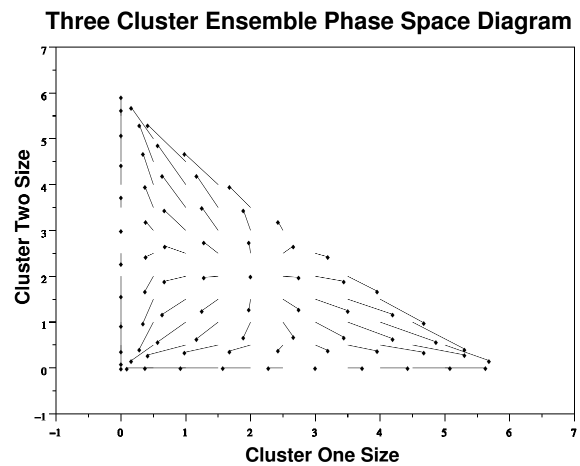
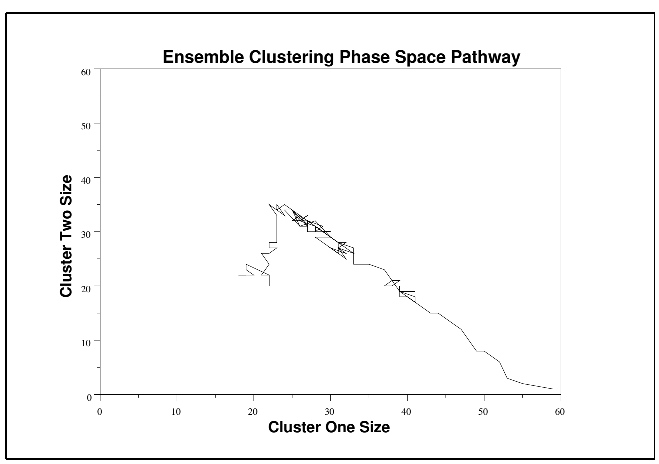


Figure 3.1: This figure illustrates the greater flow of a three cluster system (one is in the z axis, which isn't pictured) under the action of clustering agents. While the system still has the same three attractors at the corners of the region, the decision areas are much sharper due to the nature of the ensemble updates.



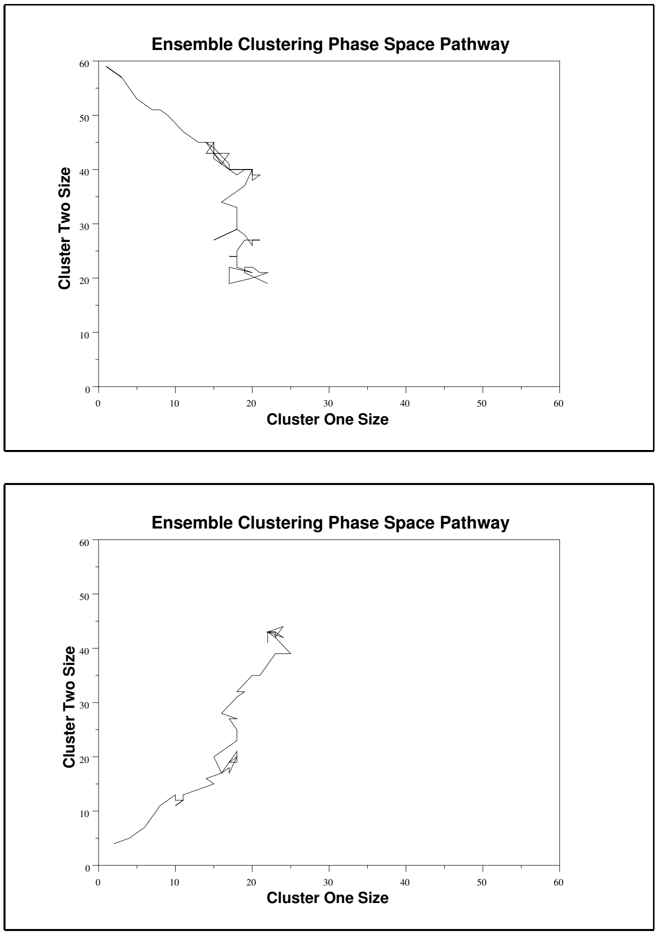


Figure 3.2: This figure illustrates the pathway of the systems through phase space. These pathways are much more directed toward the attractors as a result of the ensemble updates which limited steps in the "wrong" direction.

B. Increasing predictibility of embedded randomness in stochastic stigmergic swarms

Stochastic stigmergic swarms are very difficult to predict in general. Indeed, there are many cases where the agents carry out many steps probabilistically, depending on each step to determine the next, utilizing several global measurables when generating the set of actions, and updating many global measurables simultaneously. These systems are very difficult to analyze, and typically general comments only can be made about their properties. It is interesting and desirable to determine a way to improve the behavior of these systems and obtain coupled differential descriptions of such systems that assist in determining their behavior. In this subsection, we utilize the travelling salesman problem as a representative problem of this kind.

The traveling salesman problem (TSP) is probably one of the better researched and relatively intractible problems to which swarms have been applied [12], [13], [5]. The problem concerns finding a complete closed circuit through a set of locations that has a smaller size than any other circuit. This problem is np-complete, which means that the computation required to solve it exactly grows faster than any polynomial would grow as the number of cities grows. Many different approaches to this problem have been taken.

In the realm of swarm approaches, one of the most popular ways of solving this problem involves utilizing ant algorithm methods. These are methods derived from extrapolations of the behaviors of real ants. Ants are social animals which communicate with one another utilizing a variety of chemical messages. One of these chemical messages signals the presence of food, and is used by ants to provide recruitment signals to other ants to the source of the food. The chemical, called a pheromone, is a volatile chemical which begins evaporating as soon as it is applied to the ground by an ant. Ants encountering the pheromone follow pheromone trails left by ants which have found food to the source, and themselves lay trails from the food to the nest once they've acquired a portion of food from the source. Since the pheromone evaporates, the trail remains only when ants are continually replenishing it, which occurs when the food is not exhausted.

It turns out that food sources closer to the ant hill than others which are further away typically have thicker pheromone trails leading to the ant hill than the more distant source. This derives from the dynamics of the system, but also leads to a greater recruitment of ants for sources closer to the ant hill. A similar dynamic also leads ants to, through deviations from the first established path, dynamically improve the trail so that it becomes increasingly short. As a result, ants typically can find a short (if not the shortest path) from their ant hill to the food source despite having a crude approximation of it at the outset of the scavenging process.

As a result of this last set of considerations, many researchers have applied an ant-algorithm to TSP. In the ant algorithm, each of the cities is considered a point, and a well defined positive distance exists between them. Ants are agents which traverse the map of cities probabilistically. Ants move from city to city, making stochastic decisions about which next city to move to, and summing the total distance travelled as they move. The linkages between cities have a numerical value attached to them which is a measure of "pheromone" on that linkage. As the ant decides which city to move to next, it does so by using the pheromones on the linkages from the city the ant is currently visiting. The ant makes a choice between the cities probabilistically, with the larger pheromone values leading to higher probabilities of being chosen. Once the ant has travelled the entire map and gone back to the original city, it retraces its steps, depositing pheromone on its way back. Shorter paths lead to larger additions of pheromone, and this dynamic slowly reduces the pheromone on the linkages that tend to contribute to longer paths. The tacit assumption is that the linkages that are left are the ones that contribute to the shortest path, though the authors are aware of no rigorous proof of this assertion.

The main problem with this approach to the TSP problem is that the system really has $\frac{N!}{2}$ attractors, and there is no direct way to understand from what regions and with what initial conditions the minimum path will emerge. There is no simple way to see what subspaces will lead to the real solution or to know from which portions the final solution definitely won't emerge, though we can make a few trivial assertions. One assertion, which establishes the system's large number of attractors, is that if the system starts with a converged state, it has no way of reversing without potentially affecting the finality of any one solution. Using ant algorithms for TSP typically generates a large number of potential solutions, one each for each run, with some amount of overlap. However, there is no way to know that the converged solution is actually the optimum, or to provide a confidence interval of the solution. Despite this, many practitioners and some companies are using this and derivative approaches to solve complex problems.

In order to use the ant algorithms in a way that limits the variance of the final system state (and therefore improves the reliability of the system), a very simple ensemble approach can reduce the variance of the system. This involves making updates to pheromones on the various linkages in tandem, with updates utilizing the averaged contributions of more than one ant at a time. Though we cannot connect the global property to the individual behaviors in each step, we can look at the global effect of the ensemble on the linkages.

Initially, the update rule for the pheromone τ on the linkage between city r and city s was given by

$$\tau_{n+1}(r,s) = (1-\alpha)\,\tau_n(r,s) + \alpha\tau_o(l)\,\delta_{l,(r,s)} \tag{4}$$

where α and τ_o are parameters and $\delta_{l,(r,s)}$ is a function whose value is 1 if the link (r,s) is on the path l and 0 otherwise. Creating an ensemble changes this to

$$\tau_{n+1}(r,s) = (1-\alpha)\,\tau_n(r,s) + \alpha\varsigma_{(r,s)}p_{(r,s)} \tag{5}$$

where $\varsigma_{(r,s)}p_{(r,s)}$ refers to the product of the probability that link (r,s) is on a path in the ensemble and the average value of $\tau_o(l)$ over all paths containing the link (r,s). Since this last probability is well defined, the ensemble will generate reliable and reproducible results. It is interesting to note that this is a discrete approximation of the differential equation

$$\frac{d\tau_{(r,s)}}{d\alpha} = -\tau_{(r,s)} + \varsigma_{(r,s)}p_{(r,s)} \tag{6}$$

which is generally a set of coupled differential equations whose solution is the outcome of the problem with a given initial pheromone state. Note that, since most ant systems initially have the pheromones set to a specific state, one can expect the ensemble ant system to evolve identically each time.

Note that the solution to equation (6) in the case that ς is constant is

$$\tau = c + C_1 e^{-t} \tag{7}$$

or that all the pheromones will eventually be equal. This is the obvious solution in the case that distance-related information is ommitted and that all τ values begin uniformly. It is also easy to see that, under this formalism, starting the system in any single-path configuration yields only that path in future iterations, verifying the fact that these are attractors of the system. In fact, doing so in a neighborhood of these areas also yields the same result. Therefore, the question for ant algorithm systems is what is the region of the pheromone space that yields the optimal path?

We implement an ensemble system using a variant of the ant algorithm on the bays29 problem from TSPLIB. We utilize the pheromone update rule

$$\tau_{N+1} = \tau_N * 0.98 + \frac{1}{N_e}\sum_{i=1}^{N_e} e^{c*(d_m-d)} \tag{8}$$

where N_e represents the ensemble size, d_m is the minimum distance thus far discovered, and d is the agent's path distance. As we can see in Figure 3.3, the ensembles yielded increasingly reliable swarms, in the sense that their performance was consistent from run to run (a necessary quality of a reliable engineered system. Generating dynamics that yields the optimal state each time would seem to be dependent on choosing the right form of ς.

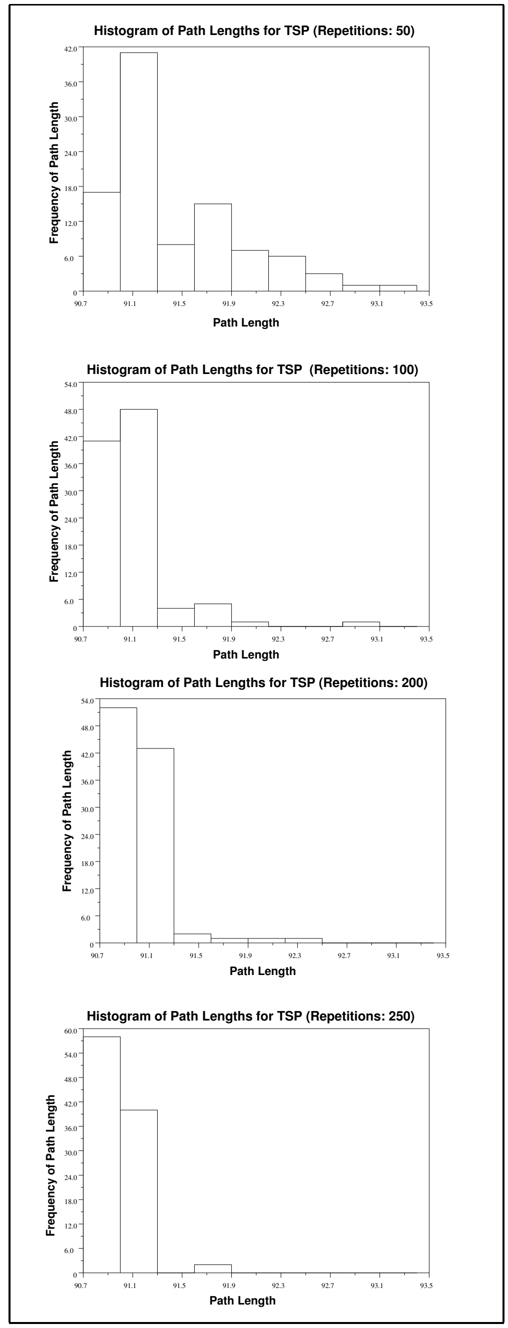


Figure 3.3: These figures illustrate the effect of increasing the agent ensemble size on this path. The minimum distance is 90.74, and the greater the ensemble size, the more reliably the system will be in achieving it. However, the system still has randomness, and the randomness can continue to bump, although with decreasing success, the system into suboptimal pathways through its phase space.

The differential equation from equation (6) is that given in equation (3), as indicated by Theorem 1. A complete investigation of the phase space dynamics is now possible using this formalism, but it is beyond the scope of this paper to do so. However, doing so will undoubtedly yield valuable insight into the characteristics (and thereby usefulness) of the ant algorithms.

IV. Discussion and Conclusions

Developing reliable swarms, whose properties generate the same results time after time, is a critical part of swarm engineering, and necessary for the eventual acceptance of this field of engineering to the general field of engineering. To date, much of the work in swarms has been largely anecdotal, generating methods of limited scope and applicability as well as many different problem specific solutions to specific problems. While this work has been fascinating and tantelizing, capturing that part of swarms that fascinate scientists and give us an idea of the grand scope of the field, more work is needed to generate flexible and general methods that capture the dynamics of a given swarm system.

Although the current work approaches the problem from a very different point of view than the work of Lehrman, Martinoli, Hayes, Agassounon, and Iispert, it is very much similar to the probabilistic modelling used by this group of researchers. Their work sought to develop numerical models, which could be computationally implemented, which captured the essence of the systems they worked on.

In our work, we examine the phase space structure of the swarm as a way of determining what the swarm will do, and then demonstrate that the swarm does indeed seen to behave according to the expected dynamics. This seems to be the case for swarms that are simple, as in the puck clustering case, and for more complex swarms, such as the construction swarms. We are able to connect the behaviors directly to the global variables, and use this information to begin to make predictions about a system's attractors. The attractors represent the final stable stage of the system.

We have also examined the application of ensemble updating rules in swarm based systems. We have found that the application of such ensembles tends to sharpen up the behavior of the phase spaces, making the variation at the borders of the various regions between attractors much smaller, thereby making the determination of the overall swarm behavior more reliable. Moreover, we have demonstrated elsewhere that equation (3) [11] holds true in general for ensemble updating behavior of global variables in swarm based systems. This aids in the development of models that can predict the behavior of the swarm. Also, using equation (3), we can create a set of coupled differential equations that indicate the behavior of the system and determine the behavior from a specific set of initial conditions for a fraction of the computation required in a full model.

The most important part of the work here derives from the idea that complex actions that take multiple stochastic steps may still be used to determine predictors of the swarm-based system. When the behaviors are tightly coupled to environmental variables, deriving the phase space is simple. Things become more complex when stochastic steps are dependent on previous stochastic steps. These tend to defy conventional analysis. However, in an ensemble-based system, a set of coupled differential equations can be generated which accurately describe the system and can be used to determine some if not most of the characteristics of the system.

References

[1] S. Kazadi. **Swarm Engineering**. Caltech PhD Thesis, 2000.

[2] A. Winfield, J. Sa, M.C. Gago, C. Dixon, M. Fisher. *On Formal Specification of Emergent Behaviours in Swarm Robotics Systems.* **Int. J. Advanced Robotic Systems**, 2(4),363-370, 2005.

[3] S. Kazadi. *On the Development of a Swarm Engineering Methodology*. **Proceedings of IEEE Conference on Systems, Man, and Cybernetics**, Waikoloa, Hawaii, USA, pp.1423-1428, October 2005.

[4] P. Wiegand, M. Potter, D. Sofge, and W. Spears. *A Generalized Graph-Based Method for Engineering Swarm Solutions to Multiagent Problems*. Thomas Philip Runarsson, Hans-Georg Beyer, Edmund K. Burke, Juan J. Merelo GuervaL. Darrell Whitley, Xin Yao (Eds.): **Parallel Problem Solving from Nature - PPSN IX, 9th International Conference**, Reykjavik, Iceland, September 9-13, 2006, Procedings. **Lecture Notes in Computer Science 4193** Springer 2006.

[5] M. Dorigo, V. Maniezzo, A. Colorni. *Ant system: optimization by a colony of cooperating agents*. Bruxelles: **Proceedings, IEEE Systems, Man, and Cybernetics Society**, 1996.

[6] S. Kazadi and J. Lee. *Swarm Economics*. **Advances in Computational Algorithms and Data Analysis Series: Lecture Notes in Electrical Engineering**, Vol. 14 Ao, Sio-Iong; Rieger, Burghard; Chen, Su-Shing (Eds.) 2008.

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

[8] C. Lee, M. Kim, S. Kazadi. *Robot Clustering.* **Proceedings of IEEE Conference on Systems, Man, and Cybernetics**, Waikoloa, Hawaii, USA, pp.1449-1454, October 2005.

[9] S. Kazadi, P. Kim, J. Lee, and J. Lee. *Swarm Economics*. **Proceedings of the World Congress on Engineering and Computer Science 2007**, San Francisco, California, USA, October 24-26, p. 602-610,2007.

[10] S. Kazadi, J. Lee, and J. Lee. *Artificial Physics, Swarm Engineering, and the Hamiltonian Method*. **Proceedings of the World Congress on Engineering and Computer Science 2007**, San Francisco, California, USA, October 24-26, p. 623-632, 2007.

[11] S. Kazadi and A. Lee. *Swarminess*. **Proceedings, 11th IASTED International Conference on Intelligent Systems and Control 2008**, Orlando, Florida, USA, November 16-18, p. 220-227, 2008.

[12] M. Dorigo and L. Gambardella. *Ant colony systems: a cooperative learning approach to the traveling salesman problem*. **IEEE Transactions on Evoluionary Computation**, 1(1): 53-66, 1997.

[13] M. Dorigo, G. Di Caro, and L. Gambardella. *Ant colonies for the traveling salesman problem*. **Biosystems**, 43: 73-81, 1997.

[14] K. Lerman, A. Galstyan, A. Martinoli, and A. J. Ijspeert. *A Macroscopic Analytical Model of Collaboration in Distributed Robotic Systems*. **Artificial Life**, 7(4):375-393, 2001.

[15] A. Hayes. **Self-organized robotic system design and autonomous odor localization**. Caltech PhD Thesis, 2002.

[16] A. Hayes, A. Martinoli, and R. Goodman. *Self-organized robotic system design and autonomous odor localization*. **Proceedings, Fifth International Symposium on Distributed Autonomus Robotic Systems**, 2000.

[17] W. Agassounon, A. Martinoli, and K. Easton. *Macroscopic Modeling of Aggregation Experiments using Embodied Agents in Teams of Constant and Time-Varying Sizes*. **Autonomous Robots**, 17(2-3), pp. 163-192, 2004.

[18] W. Agassounon. **Macroscopic Modeling of Aggregation Experiments using Embodied Agents in Teams of Constant and Time-Varying Sizes**. Caltech PhD Thesis, 2003.