Physical Parameter Optimization in Swarms of Ultra-Low Complexity Agents

(Short Paper)

Ryan Connaughton
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556, USA
rconnaug@nd.edu

Paul Schermerhorn
Cognitive Science Program
Indiana University
Bloomington, IN 47406, USA
pscherme@indiana.edu

Matthias Scheutz
Cognitive Science Program
Indiana University
Bloomington, IN 47406, USA
mscheutz@indiana.edu

ABSTRACT

Physical agents (such as wheeled vehicles, UAVs, hovercraft, etc.) with simple control systems are often sensitive to changes in their physical design and control parameters. As such, it is crucial to evaluate the agent's control systems *together* with the agent's physical implementation. This can consequently lead to an explosion in the parameter space to be considered.

In this paper we investigate the use of swarms of ultra-low complexity agents, and address the issue of finding workable physical agent parameters. We describe a technique for reducing the dimensionality of the search space by performing evaluation tasks that can be used to predict near-optimal parameter values for agents in related multi-agent tasks. We validate our approach on an example task, and demonstrate that this technique can greatly reduce the computational resources required to design a multi-agent system.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Algorithms, Design, Experimentation

Keywords

Agent-based simulations, Swarm, Optimization

1. INTRODUCTION

Multi-agent swarms provide attractive solutions for many important search tasks, such as detecting or tracking targets of various types ([4, 5]), especially when the agents that compose the swarms are low-complexity, low-cost platforms. For example, we have previously demonstrated a method of detecting and tracking targets using ultra-low complexity agents that is generalizable to a range of vehicle types [8].

Cite as: Physical Parameter Optimization in Swarms of Ultra-Low Complexity Agents (Short Paper), Ryan Connaughton, Paul Schermerhorn and Matthias Scheutz, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 1631-1634.

However, these demonstrations used a highly idealized agent model that does not take important physical characteristics of many types of embodied agents into account. The aim of the current paper is to investigate the effects of using a more physically realistic simulation model, in particular the interaction between physical and control dimensions, and we present and validate a technique for optimizing agent parameters of this nature.

2. OPTIMIZATION METHOD

We will use the following method, which is aimed at reducing the search space for the problem of optimizing physical and control parameters in a multi-agent swarm:

1. Suppose agent with physical/control parameters $(P_1, ..., P_n)$, and multi-agent swarm task T
2. Determine performance measures for swarm for T: $f(T, P_1, ..., P_n)$
3. Select some parameter P_i, which impacts $f(T, P_1, ..., P_n)$, such that there is some less expensive "predictor task" T_p with performance measure $g(T_p, P_1, ..., P_{i-1}, P_{i+1}, ..., P_n)$ that can be used to predict optimal values for P_i in T
4. Run T_p for all values of the parameter space $(P_1, ..., P_{i-1}, P_{i+1}, ..., P_n)$ and record the values of $g(T_p, P_1, ..., P_{i-1}, P_{i+1}, ..., P_n)$
5. The recorded performance values for each parameter configuration in T_p are taken as values of P_i for the corresponding parameter configurations in T, eliminating P_i as a parameter of the larger, more costly experiment set.

Essentially, an agent parameter P_i for the swarm configuration is selected, and a new task T_p, which is less computationally expensive in simulations than the original swarm task T, is used to predict an optimal value for P_i for each possible configuration of the other agent parameters.

For validation, we apply our technique to an example multi-agent task. The target monitoring task requires a multi-agent group to keep the specified target (e.g., a cloud of hazardous chemicals, or some military unit) under continuous observation by at least one agent. We assume a simple *hovercraft* agent model, described in the following section.

Despite the low complexity of the agent model, the complicated interaction between physical and control parameters and agent behavior creates a difficult optimization problem. We apply our method by selecting and simulating our predictor task, and then verify its success by simulating the multi-agent task as well.

2.1 Agent Model and Control System

For the purposes of this paper, we assume that the swarm agent platform is already determined: a hovercraft with two stationary thrust fans for directional control and a very simple "bang-bang" control system: either the left or right fan (but not both) is activated at any given time. This model is motivated by its ultra-low cost and complexity. However, due to the control constraints, optimal control of individual hovercraft agents is not possible. The following rulesystem constitutes the agent model for the hovercraft agents:

R1: If distance between self (A_S) and nearest peer (A_N) is less than avoidance range ($D_{A_S,A_N} < AR$) and A_N is to the left ($\theta_{A_N} \leq 180$, relative to current heading) then engage left motor.

R2: If $D_{A_S,A_N} < AR$ and $\theta_{A_N} > 180$ then engage right motor.

R3: If target (T) is to the left relative to current heading ($\theta_T \leq 180$) then engage right motor.

R4: If $\theta_T > 180$ then engage left motor.

Note that the details of acquiring the target location are not addressed here (see [8] for one solution); it is assumed that the target is either (a) always within sensory range or (b) at a known fixed location, and thus its location is always available. Effectively, the agent first checks to see if its closest peer (A_N) is within its avoidance range (AR); if so, it determines whether that agent is to its left or right, and then activates the fan on that side, thus turning away. Similarly, if no peers are nearby, the agent determines whether its target is to its left or right, and then powers on the opposite motor, in order to turn towards the target.

Assuming this simplified hovercraft model, we approach the problem of selecting the best combination of physical values for the hovercraft's parameters to accomplish the swarm's objectives. The control system for the hovercraft is fixed, but there are two parameters available for optimizing agent performance. The first is the agent's *avoidance range*, defined to be the distance between two agents at which they engage in collision avoidance behavior. In practice, a large avoidance range leads to fewer collisions.

The second parameter we are able to vary is the thrust applied by each fan, which is fixed for the duration of each experiment. Intuitively, higher thrust values lead to faster hovercraft movement and increased rotation speed, but excessive thrust can make it difficult to accurately control the agent's movement. We assume all other agent parameters are fixed.

Using this control scheme, it is evident that precise control of the hovercraft's movement is not possible. There is no straightforward way to stop or even slow the vehicle. The agent can only react to the target's angle relative to itself; there is no anticipation of the need to change direction as a target is approached. Instead, the agent continues until it has passed the target, at which point it must overcome any momentum it has built in order to turn back toward the target. Hence, the agent's target-monitoring behavior takes the form of constant "oscillations" around the vicinity of the target.

2.2 Task Description

Given the target-monitoring task and the agent model described above, we are required to optimize the avoidance range and thrust parameter values of the hovercraft. First, performance measures must be determined for the multi-agent task. The target-monitoring task requires the agents to stay as close to a stationary target as possible while minimizing collisions. The control system makes it impossible for the agents to stop, so the agents continually "swarm" around the target. We use the area enclosed by this swarm as a performance measure. Generally speaking, low swarm area is better, as it implies that the swarm is able to more precisely define the region in which the target is located. Naturally, low swarm area increases the likelihood of collisions, so good performance requires an agent configuration that balances these two priorities.

The crucial step in the parameter reduction process is determining a good candidate parameter for elimination. Here, we would like to find a predictor task that allows us to derive avoidance range while varying only the thrust dimension. In this case, note that ideal avoidance range should be directly correllated with the agent's ability to change direction. An agent that is capable of quickly changing direction is likely to need a smaller avoidance range to avoid collisions. Thus, a single-agent task which measures the distance needed to turn around may make a good predictor task. From this, we derive the following task description.

In the single-agent task, a hovercraft is randomly placed in an environment containing a stationary target. According to the control algorithm, the hovercraft is continuously attracted to the target, but it is unable to stop. Thus, the agent typically overshoots the target location by some distance, and continually oscillates around the target for the remainder of the fixed duration run. We take the space required for an agent to return to the target after detecting that it has passed it to be an indication of the agent's ability to react to sensed events. This distance is the agent's *maximum overshoot* (MO) for that "cycle" (from the target and back). Note the agent records a new MO value each time it returns to the target. Intuitively, the avoidance range indicates how soon the agents react to one another, and the MO represents an agent's ability to adjust to a target. Thus, an agent with a high MO should require a large avoidance range.

The MO values are used to predict avoidance range in the following way: the MOs for each visit to the target in an experimental run are sorted in ascending order. The value of the MO at the nth percentile in the sorted list is taken as that run's prediction for the optimal avoidance range value in a swarm of these agents. The value of n selected reflects the relative importance of the two competing performance measures. For example, with larger n, the more conservative avoidance range leads to fewer collisions but greater swarm area.

With the single-agent task defined, it can be simulated using various values for thrust and the MO can be recorded as the predicted avoidance range for each thrust value.

To verify our results, we simulate the full multi-agent task for each value for *both* parameters, thrust and avoidance range; note that the avoidance range dimension is included only for the purposes of demonstrating the accuracy of the technique. The multi-agent task again requires a cluster of agents to stay as close to a stationary target as possible. As described, we use swarm area and the number of collisions to evaluate swarm performance. To measure the swarm's area at a particular point, we compute the area of the convex hull enclosing all agents. The average swarm area for each cycle over the course of an experimental run is taken for the performance measure. Further, the number of collisions are recorded for each simulation.

3. EXPERIMENT AND RESULTS

The simulations were carried out in SimWorld, a simulation environment built upon the SimAgent toolkit [1]. Supplementing the SimWorld environment is a library of C functions that makes use of the Open Dynamics Engine (ODE) [7], a rigid body physics engine. Both the single and multi-agent simulation tasks take place in an unbounded continuous 3D environment. Agents are initially randomly placed in a uniform distribution throughout a 400 m $\times$ 400 m environment with the target placed at the center. Simulations run for 10,000 cycles, or 500 seconds of simulated time.

In our representation of the hovercraft, the mass of the hovercraft was set at 1.5 kg, and the coefficient of friction between the ground and the hovercraft was set at 0.03. ODE does not consider air drag in its calculations, so we approximate the force of air drag for each agent using the standard equation. We then apply this force to the agent such that it opposes the agent's direction of motion.

In the single-agent experiments, the force representing fan thrust was varied from 0.7 N to 2.0 N, in intervals of 0.1 N. In the multi-agent experiments, thrust was varied from 0.8 N to 2.0 N, in intervals of 0.4 N, and avoidance range was varied from 5 m to 50 m, in 5 m intervals. A swarm of 20 agents was used for each run. 100 experimental runs were conducted for each configuration, and all results presented in the following sections represent averages over these simulations. In total, 1,400 single-agent simulations and 4,000 multi-agent simulations were conducted. Each run of the single-agent task took approximately 10 seconds to complete. The multi-agent task took approximately 60 seconds. Thus, approximately 4 CPU hours were used for the single-agent simulations, whereas approximately 2.75 CPU days were required for the multi-agent simulations.

The results from the single-agent task are summarized in Figure 1, which shows the 40th, 60th, and 80th percentile MOs for each thrust value tested. It indicates a parabolic trend in the MO values, with the lowest point reached at 1.3 N. Given these results, it is necessary only to determine the degree of risk one is willing to take and select the corresponding level (e.g., 80th percentile if collisions are viewed as catastrophic, or 40th percentile if ensuring a tight net around the target is more important). The avoidance ranges indicated by the selected threshold can be used in the determination of the optimal thrust setting.

For the purposes of the present paper, however, we proceed with the multi-agent experiments as though we had not conducted the predictive study. Figure 2 shows the relationship between avoidance range and the average number of collisions per experimental run. A super-linear decline in the number of collisions emerges as the avoidance range increases. Without the single-agent results, it is necessary to examine the collision data to determine for each thrust value the lowest collision avoidance range that yields an acceptable number of collisions. The small black circles on the curves in Figure 2 indicate the avoidance ranges predicted using the 80th percentile results from Figure 1. The single-agent MO proves to be an accurate predictor of appropriate avoidance ranges for swarms with equivalent thrust; as the MO percentile used to predict the avoidance range is increased, the average number of collisions per experiment for the corresponding swarm approaches 0.

From this, we conclude that the single-agent task is sufficient for choosing the avoidance range for a swarm with equivalent thrust. By varying the percentile used to determine the maximum overshoot in the predictor task, it is possible to qualitatively specify the maximum acceptable collision rate. Further, this percentile value can be varied after the simulations have completed, eliminating the necessity to re-execute the predictor task.

Once acceptable avoidance ranges have been decided on, we can proceed to evaluate the relative performance of each

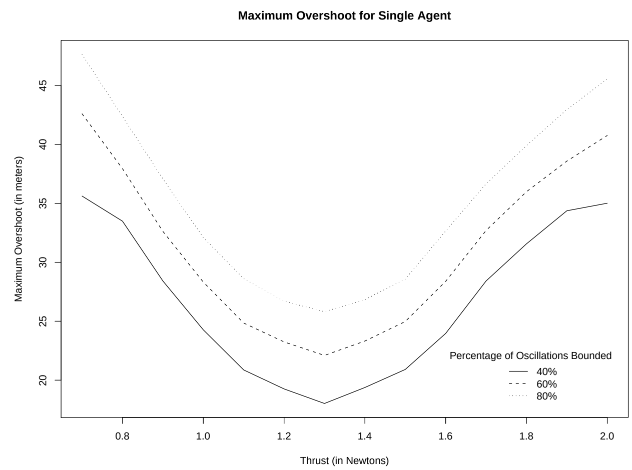

Figure 1: Predicted avoidance ranges from single-agent results.

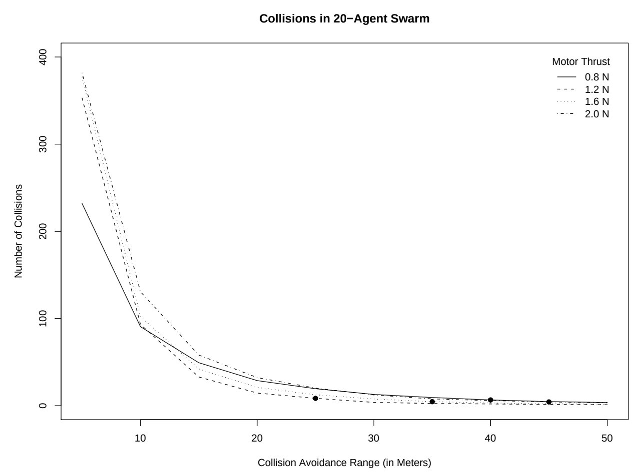

Figure 2: Collision counts for the multi-agent task.

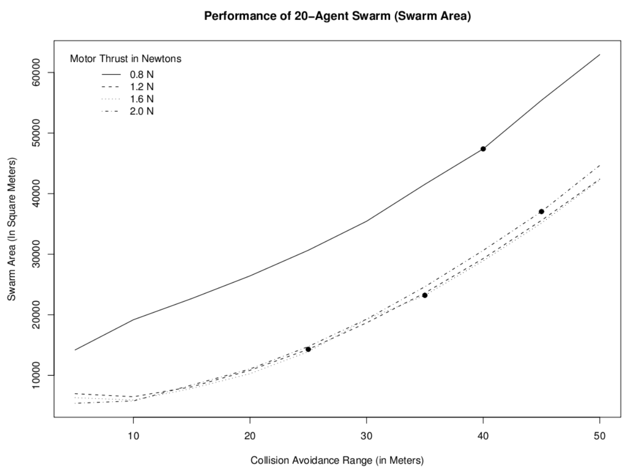

Figure 3: Average swarm area for the multi-agent task.

thrust configuration in the multi-agent task. As we have reduced the problem to a single dimension, this amounts to comparing the performance of just four hovercraft configurations (one for each thrust value). Figure 3 depicts the average area enclosed by the swarm as a function of avoidance range for each of the thrust values in question; predictably, swarm area increases quadratically with the avoidance range. However, we need only look at the values corresponding to the avoidance ranges settled on above (indicated, once again, by a small black circle on the line for each thrust,) and determine which produces behaviors leading to the most densely-packed swarms. Again, the 80th percentile MO is indicated by the small black circles on the line for each thrust. We conclude from this data, that the optimal configuration is a thrust of 1.2 N and an avoidance range of 25 m.

4. RELATED WORK

The effectiveness of autonomous swarms has been studied for a variety of applications. Swarm techniques have been applied to toxin detection, as in [3], where chemical gradient information is used to guide the swarm, and in [9], which considers fluid dynamics to follow the chemical trail. In data mining applications, swarms have also been used to perform data clustering. For example, [2] modifies particle swarm optimization (PSO) concepts to allow agent swarms to search an n-dimensional data set to find appropriate data clusters. Given the wide range of uses for swarm technology, we focus on the applications in which the swarm agents represent physical bodies. Specifically, we assume a hovercraft agent model, though aspects of the work presented here should apply to a range of physical autonomous agent models.

The use of swarms for target tracking has been the focus of research for some time. In [8], each swarm agent is equipped with an "attractive beacon" that is activated whenever it detects a target, consequently attracting other agents to that area. The task is then accomplished in two phases: (1) a detection phase, in which agents search for the target cluster until one agent detects it, and (2) an enclosure and tracking phase, during which the agents cover the cluster area as tightly as possible. In this paper, we extend this to account for the physical realities of implementing the swarm.

The authors in [6] implement a simulated swarm for real-time pattern tracking. An approach motivated by genetic algorithms is applied to PSO to achieve facial recognition. While a similar, genetically-inspired approach could be taken to the task of parameter selection which we present, we instead focus on reducing the size of the search space by one or more entire dimensions. This is made possible by examining the intuitive role of physics in the behavior of the swarm, a concern which was not applicable in [6].

5. CONCLUSION

The goal of this paper was to present and validate a technique for physical agent parameter optimization for a multi-agent swarm. We demonstrated our presented technique by determining values for the thrust applied by the hovercraft fans and the *avoidance range*, defined to be the distance between agents at which they engage in avoidance behaviors. We simulated a swarm of hovercraft for a range of thrust and avoidance range values, and showed that our avoidance range prediction maintained a collision rate near zero. Finally, we used a small subset of the swarm simulations and area of the swarms to determine the optimal thrust value. In future work, we plan to explore whether this technique can be extended to a wider variety of agent parameters.

6. REFERENCES

[1] Aaron Slalom. http://www.cs.bham.ac.uk/research/projects/poplog/packages/simagent.html.

[2] S. C. M. Cohen and L. N. de Castro. Data clustering with particle swarms. In *IEEE Congress on Evolutionary Computations*, 2006.

[3] X. Cui, C. T. Harden, R. K. Ragade, and A. S. Elmaghraby. A swarm approach for emission sources localization. In *Proceedings of the 16th International Conference on Tools with Artificial Intelligence (ICTAI 2004)*, 2004.

[4] A. T. Hayes, A. Martinoli, and R. M. Goodman. Distributed odor source localization. *IEEE Sensors, Special Issue on Artificial Olfaction*, 2(3):260–271, 2002.

[5] S. Kazadi, R. Goodman, D. Tsikata, and H. Lin. An autonomous water vapor plume tracking robot using passive resistive polymer sensors. *Autonomous Robots*, 9(2):175–188, 2000.

[6] C. Middendorff and M. Scheutz. Real-time evolving swarms for rapid pattern detection and tracking. In *Proceedings of the 10th International Conference on the Simulation and Synthesis of Living Systems*, Bloomington, IN, 2006.

[7] ODE. http://www.ode.org.

[8] M. Scheutz, P. Schermerhorn, and P. Bauer. The utility of heterogeneous swarms of simple uavs with limited sensory capacity in detection and tracking tasks. In *IEEE Swarm Intelligence Symposium 2005*, 2005.

[9] D. Zarzhitsky, D. F. Spears, and W. M. Spears. Swarms for chemical plume tracing. In *IEEE Swarm Intelligence Symposium 2005*, 2005.

This work was supported in part by the Defense Threat Reduction Agency (DTRA) and Naval Surface Warfare Center-Crane division under grant N00164-07-C-8510.