Optimizing Swarm Intelligence in Solving Transport Problems

Kristin Stephens

May 3, 2009

Introduction

Swarm intelligence is based on collective behavior of a self-organized group of agents. Each agent is following a relatively simple set of rules and interacting with only its local surroundings. Flocking behavior is when a swarm is acting similarly to a group of birds or herd of animals.

The problem that is being addressed in this project is transportation. The swarm in the artificial world has a goal of moving a set of supplies from one base, the home base, to another, the goal base. This goal has two parts. The swarm will start at the home base with the supplies and will first need to find the goal base's location. After an agent in the swarm has succeeded in finding the goal base, it switches to the second goal of transporting all supplies in the home base to the goal base.

The swarm will be optimized by altering the swarm's agent's behavior using a genetic algorithm. Genetic algorithms are based on the evolutionary theory of natural selection. Its usual steps include: finding the fitness value for each individual in a pool of genomes, creating a new population by probabilistically selecting the fittest individuals, and repeating. The population generation is done by either duplicating a genome or taking two parent genomes and creating offspring by crossover and mutation.

Using a genetic algorithm on the swarm in the artificial world, I hope to find an effective strategy to move the most items from one base to another in a disaster type environment. A disaster type environment in this experiment will be defined as a two dimensional artificial world with stationary obstacles present that an agent can only go around. My hypothesis is that flocking behavior will appear as the optimal solution.

Background

Multi-agent systems and swarm intelligence were first introduced by G. Beni with cellular robotics [1]. Swarm intelligence is one approach to solving problems by imitating nature and, more specifically, by imitating swarms such as bees, ants, and birds. The significance of swarm intelligence is that each agent in the swarm is governed by a simple set of rules, but the entire swarm as a whole exhibits complex behavior. This also usually means that each agent does not have global knowledge of its environment nor does it have access to all knowledge of the swarm. It acts based on local environmental knowledge and nearby neighbors only.

Genetic algorithms are a part of evolutionary computation, which is the process of growing and/or developing a population. Each individual in the population is represented by a genome that is made of the aspects that can be changed. The full genetic algorithm process used by this experiment will be: (1) Generate a pool of genomes of a mix of both random values and hand-made. (2) Run each genome in multiple simulations to obtain the genome's fitness value. (3) Generate a new population based on probabilistically selecting individuals from the population, with the higher the fitness the higher the probability of being chosen. After being chosen a genome can either be duplicated into the next generation or create "offspring" with another genome through crossover and mutation. (4) Recurs from step (2).

Flocking Optimal Solution for Transportation Problem without Obstacles

In [5] there is a study that found flocking behavior to be more effective than independent movement for a similar transportation problem. The transportation problem had homogeneous swarms based on different genomes compete for the same resources in a two dimensional environment. An agent's goal was to find resources, in this experiment labeled minerals, and then transport the minerals from the mineral deposit to their home base.

Independent variables were the swarm's agents ability to either have flocking or independent movement and follow one of three guarding strategies. The guarding strategies were full-guarding, home-guarding, and non-guarding. Full-guarding was where the home base and any mineral deposits found were guarded from enemy agents. Home-guarding required agents to only guard their home base. And non-guarding swarms did not guard anything.

In the simulations where two teams competed for minerals with the guarding type held constant and one swarm set for flocking behavior and the other with independent movement, the flocking team was more effective in solving the search and transport problem. The experiment continued by placing more than two swarms in a single environment. It was found that no matter what combination the six possible teams competed in the flocking agents would be more effective than their independent counterpart with the same guarding type.

The conclusions of [5] were swarms of agents that moved collectively were more effective than their independent counterparts. This led me to hypothesis that flocking behavior will evolve as the most effective strategy for my environment. My experiment will differ in when finding a swarm's fitness during the evolution process only one swarm will be present in the environment and it is working under a time limit rather than competition. However these constraints are similar to [5]'s experiment because each swarm would have to move as many items as possible as quickly as possible to increase their mineral holdings and replenish any minerals that are stolen from their home base.

[5] also shows that flocking was more effective because when an agent discovers a mineral deposit and moves towards it, the agent's neighbors move towards the mineral deposit due to their flocking tendency. By moving towards the mineral deposit the other agents were more likely to discover the mineral deposit as well. This could also benefit the swarms in my simulation during the first goal where agents are searching for the goal base.

Evolution of Inter-Agent Signaling in an Artificial World

The purpose of [3] was to systematically find conditions when grounded signaling does or does not evolve and find how variations in the fitness function would influence the outcome. The experiment had agents with a two bit genome that represented whether an agent sent or received a particular signal. One bit represented communication about food discovery and the other bit represented the communication of predators. So there are four possible agents. An agent could not communicate, communicate only about food, communicate only about predators, or communicate about both.

Instead of each agent being part of a swarm that had a group goal, it was programmed to simply avoid predators, seek food, and consume food to stay alive. Also there was no definite divide between each generation as previously explained about genetic algorithms. Agents would be removed from the population due to starvation, old age, or predation. When the population drops below the initial number the simulator automatically adds to the population at the beginning of the next time step.

Evolution happens at the step where new agents are created. A child agent is created based on the cross over and probabilistic mutation of its parents. The parent selection process was based on tournament selection. However the tournament size could either be two or ten. A tournament size of two meant the agent's fitness value was not relevant to selection; it only had to survive to reproduce. A tournament size of ten meant that the agent had to both survive and win the tournament selection based on its fitness value.

[3]'s results showed that depending on the definition of fitness, and therefore which agents could reproduce, the outcome of communication evolution can vary under some conditions. This shows that there should be special attention paid to the assumptions made on the evolutionary process used. This led to a much more simple fitness function than originally thought for the experiment presented in this paper. One original idea was to base part of the fitness on how well a set of rules kept agents close together during the transport stage of the problem to improve detection of predators and a higher portion of agents fleeing unharmed. It was immediately discarded when it was evident this could encourage flocking behavior without it necessarily improving the performance of the swarm.

Methods

Each experimental run of a simulation will start with all supplies randomly placed within a set distance of the home base. Agents will be placed at random around the supplies. The experiment will run for a predefined period of time that will be determined after a series of trial runs. Each instance of the population will be run several times with different random number seeds and the average fitness value will be used for that instance. Before running the simulation on the disaster environment, a swarm will be evolved in an empty environment. This is to create a base-line for comparison and to find the best strategy without any attributes in the environment that can deter the swarm from fulfilling its goals. Possible future work is to introduce predators into the environment, as well as allowing signaling to evolve.

Environment

The environment will consist of bases and obstacles. The bases will be randomly placed, but required to be far apart. Around the entire environment obstacles will be placed. This was to remove the need for checking if an agent leaves the boundary of the system.

Obstacles will be randomly placed throughout the environment and have varying sizes. They are stationary and impenetrable. Neither an agent nor a supply can occupy the same space as an obstacle. When an agent is being moved from one time step to the next and their path intersects with an obstacle the agent's direction will be reversed. It will then continue the same distance in the opposite direction it would have taken if the obstacle was not present. This is to act as a deterrent on agents from hitting the wall because they incur a penalty by reducing their progress more compared to if they had simply gone around the obstacle.

Agent Abilities

The main focus of this project is the agents. An agent's movement is based on its current state and a set of velocities (Table 1), or forces. The agent will have a limited memory that will remember two locations, the location of the home base and the location of the goal base once it is found.

The agent has three possible states that alter the agent's behavior by changing the goal seeking force's direction (Fig 1). The agent initially starts in the *searching state*, where it is seeking the goal base. The goal seeking force direction is randomly chosen and has a probability of changing at each time step. After the agent finds the goal and records it in its memory it switches to the *go home for supply state*. This is because once the agent finds the goal base it needs to go back to the home base to pick up a supply. This state changes the goal seeking force's direction to point toward the home base and stays constant until this state is exited. The final state is the *carry supply to goal* state. This state is reached when an agent is holding a supply and now needs to drop it off at the goal base. The goal seeking force's direction is changed to lead towards the goal base, which at this point is in the agent's memory. After an agent drops its current supply at the goal base, it will switch back to the go home for supply state and cycle between that state and the carry supply to goal state.

Collision avoidance	Causes agent to move away from other agents and objects
Velocity matching	Causes an agent to match its speed and direction with the agents around it
Cohesion	Moves agent towards the center of the agents around it
Separation	Moves the agent away from the center of the agents around it
Clearance	Steers an agent toward the side when there is an agent in front of it
Goal seeking	Steers an agent towards its current goal
Predator avoidance	If predators are enabled this velocity moves agents away from predators

Table 1

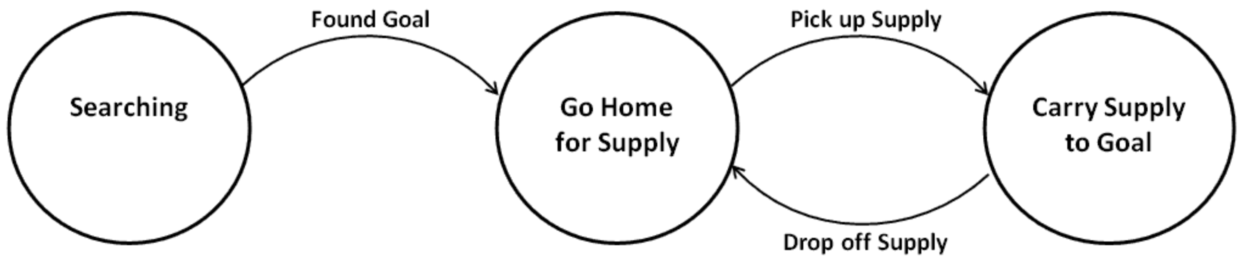

Fig 1

The criteria to pick up a supply will be if an agent is within a predefined distance from a supply and is in the go home for supply state. Dropping off will be similar in it must be within a certain distance of the goal base. However to prevent agents from dropping off supplies at the perimeter of the goal base they will be required to either drop off the supply directly on the goal location or as close as it can without colliding with supplies already present.

Genetic Algorithm

The *genome* for the genetic algorithm will have two values from 0 to 1 for each force. The first value of the pair will determine the order the forces are added to the agent's velocity until the velocity has reached a maximum. The order starts at the force with the value closest to 1 and goes down. The second value for each force is the weight given to the force when it is added into the agent's velocity. Each state will have its own set of force value pairs. And can be thought of as a chromosome in the genome. This is so the order and weights evolve based on the task the agent is performing.

The genetic algorithm's *fitness function* will be based only on the number of supplies present at the goal base at the end of the time limit. Other additions to the function were considered, but discarded because the goal of the problem is to find the most efficient strategy to solve the transport problem. Any additions to the fitness function may skew the evolution.

During each evolution iteration genomes will be probabilistically chosen based on their fitness value. Some will be moved to the next generation unchanged. Others will be paired with another genome to produce offspring. For crossover each chromosome is handled individually. A random position between the pairs will be chosen and used as the splitting point. Mutation occurs by randomly selecting a single value and mutating it to another value between 0 and 1.

Results

The simulator is currently being built. The initialization step has been implemented where obstacles, bases, agents, and supplies are all placed in the environment. The number of obstacles, agents, and supplies can be specified at each experiment run. The visualizer for the simulator is complete, which allows for visual observation of the swarm's movement. Currently being implemented are the forces and state transitions. Afterwards the mechanism for picking up and dropping off supplies will be implemented, which would complete the simulator.

Discussion

Since the simulator is not done future work will initially involve finishing it. After preliminary tests to determine appropriate parameter values, the simulator will be run with and without obstacles. Possible extensions of the simulator are introducing predators into the environment and allowing the evolution of signaling.

Predators would follow a simple set of rules similar to [3]. It can see in all directions and wanders until it sees an agent. Once it sees an agent it will give chase for a set period of time, randomly choosing an agent if there are multiple nearby. If it catches an agent, not necessarily the one it was giving chase to, or runs out of time it will enter a quiescent state, where it is motionless. In the beginning of a simulation run predators will either start in the quiescent state to allow agents to get away or have a minimum distance they must be from the home goal. The argument latter is that if this is emulating a disaster environment the agents and supplies can be thought of as dropped off in a safe zone at the beginning of the simulation run. At the time of starting this simulation a decision will be made as to which will happen or if it will be an independent variable to find if it would have a significant effect.

Signaling could also be allowed as a possible evolved state for the swarm's agents. The agents will be able to signal on the discovery of the goal base and detection of a predator. Signals will only be able to go a predefined distance and also incur the cost of attracting predators. A possible addition to this is allowing agents to relay a signal. As in if a signal is heard an agent can also send a signal with the same message, but altering it so that other agents know it is a relay of the original message. However to prevent this relay signal from spreading to the entire swarm a cost should be incurred or agents will only be able to relay the original signal. Upon hearing a signal agents in the searching state will change the direction of their goal seeking force towards the signal.

References

[1] G. Beni. The concept of cellular robotic system. In Proceedings of the IEEE International Symposium on Intelligent Control, pages 57–62, 1988.

[2] T. M. Mitchell *Machine Learning*, pages 249-270. McGraw-Hill, 1997.

[3] J. Reggia, R. Schulz, G. Wilkinson, J. Uriagereka. Conditions Enabling the Evolution of Inter-Agent Signaling in an Artificial World. *Artificial Life* Vol. 7, Number 1.

[4] C. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

[5] Rodriguez, A. 2007 *Guided Self-Organizing Particle Systems for Basic Problem Solving*. Doctoral Thesis. University of Maryland at College Park.

[6] S. Russell, P. Norvig *Artificial Intelligence A Modern Approach*, pages 116-119. Prentice Hall Pearson Education Inc., Upper Saddle River, New Jersey 07458, 2003