Swarm and Steady Wins the Race: Visualizing Constraint Satisfaction Problems Using Swarm Intelligence

Kyle Rossi | Computer Science | Advised by Daniel Palmer

THE COLLEGE OF WOOSTER

Project Overview

This project visualizes a complex constraint satisfaction problem (CSP), with basis in automated manufacturing, using the methods and techniques available in swarm intelligence (SI). The final program can use a swarm of agents to create multiple pathways between fixed locations on a screen.

Swarm Intelligence

Swarm intelligence (SI) encompasses any algorithm inspired by the social behaviors found in insect or animal colonies. SI has two important components. The first is that swarms are self-organizing, meaning that simple individual behaviors lead to complex group behaviors. This is accomplished via the use of positive/negative feedback, frequent interaction between individuals, and randomness. The second is that any "agent" in a swarm acts on their own and only make decisions by using local information (Figure 1). [2]

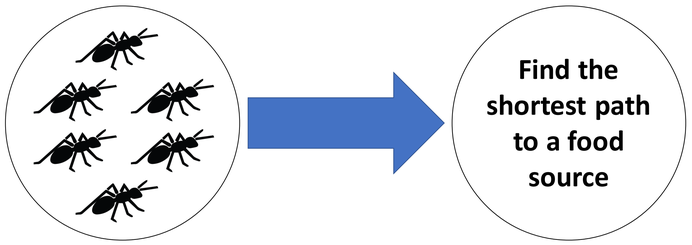


Figure 1. A swarm of simple ants can perform a complex task

Constraint Satisfaction Problems

For a constraint satisfaction problem (CSP), given a set of variables, all variables must satisfy every constraint. In practice, an efficient method to find a solution might not exist, so it is adequate that only most of the constraints are satisfied. Importantly, CSPs generally have multiple solutions. Common examples include the graph coloring problem and cryptarithmetic puzzles. A solution to the former shown is shown below, where none of the adjacent states have the same color (Figure 2). [1]

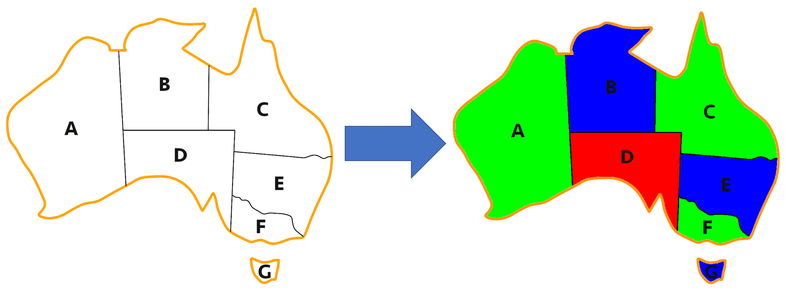


Figure 2. A possible solution to the map coloring problem with the map of Australia

Agent and Pickup Behaviors

Agents (circles) can swap between holding or not holding a pickup (square). Pickups are split into three types: movables, unmovables, and permanents. Agents can pick up movables but are unable to pick up unmovables. Permanents are mostly "invisible" to agents, except that movables cannot be placed on top of a permanent. Each have their own color, as displayed in Figure 3.

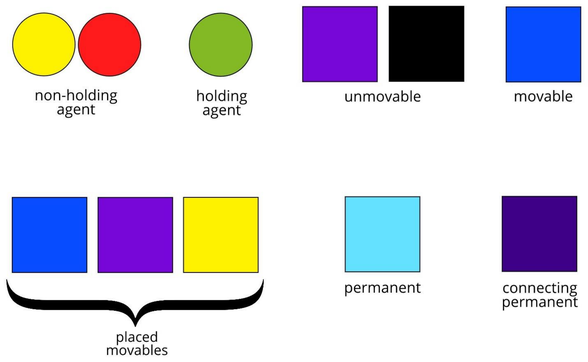


Figure 3. All possible colorations for agents and pickups

Results: Simple CSP

For the Simple CSP, the agents needed to space out all pickups at least 100 pixels away from one another. Two algorithms were implemented to accomplish this. For Algorithm #1, whenever an agent places a movable, it becomes an unmovable. Movables stay as movables when placed by agents in Algorithm #2. Although Algorithm #2 has the potential to get out of tricky situations, its performance was worse than Algorithm #1 (Figure 4).

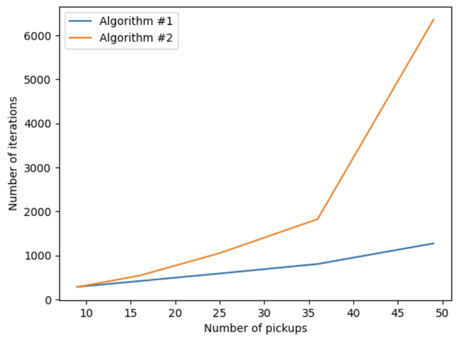
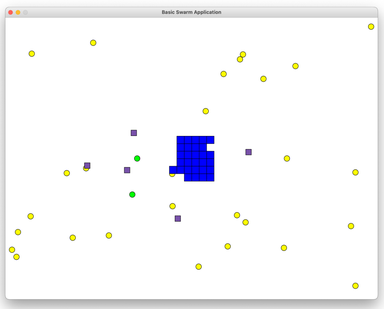
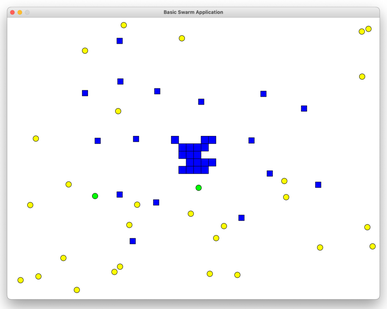


Figure 4. The average number of iterations both algorithms needed to space out a set number of pickups; 100 runs for each algorithm



Algorithm #1



Algorithm #2

Results: Complex CSP

For the Complex CSP, the agents needed to create paths from one unmovable to another. Two algorithms were implemented to accomplish this. Model A focused on loose placement conditions while requiring more pickups to make a path. Model B has stricter placement rules in place to make pathways more efficient. Model A created 2.27 paths on average, while Model B made 2.87 paths on average. When making only 1 or 2 paths, Model B has the edge, but Model A performs much better for 3, 4, or 5 paths (Figure 5).

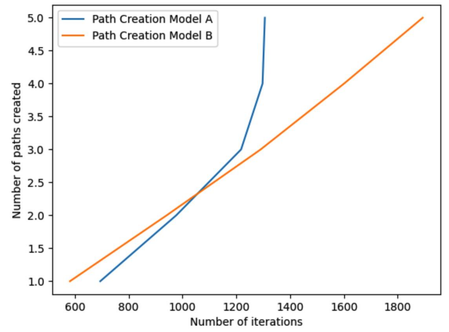
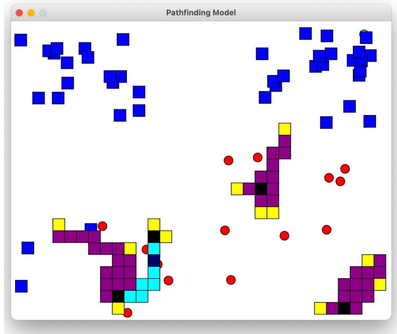
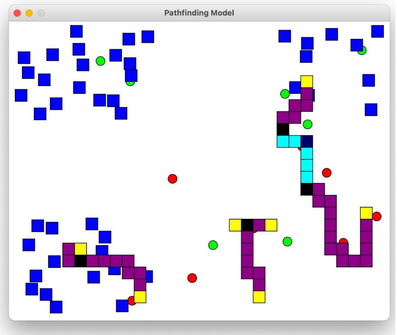


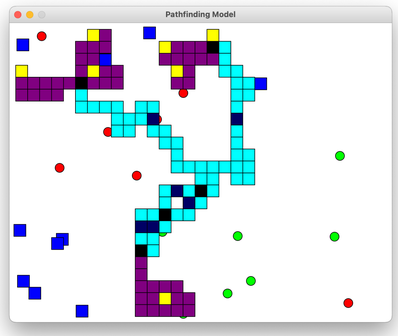
Figure 5. The average number of iterations both algorithms needed to create 1, 2, 3, 4, or 5 paths between unmovables; 100 runs for each model



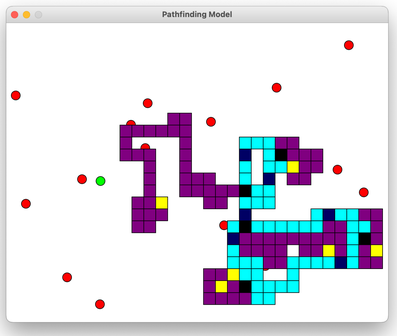
Path Creation Model A *(in-progress)*



Path Creation Model B *(in-progress)*



Path Creation Model A *(best case)*



Path Creation Model B *(best case)*

References:
1. Sally C. Brailsford, Chris N. Potts, and Barbara M. Smith. “Constraint satisfaction problems: Algorithms and applications”. In: *European Journal of Operational Research* 119.3 (1999).
2. Leen-Kiat Soh. “Swarm Intelligence.” *Swarm Intelligence - UNL Computer Science & Engineering*, University of Nebraska, 2016.