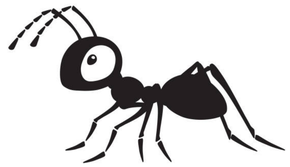
Distributed Ant Colony Optimization on the Travelling Salesman Problem

Andrea Malleo

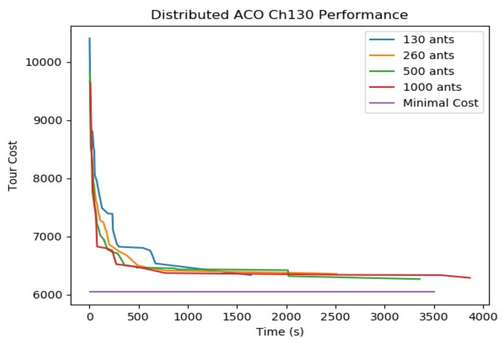
Distributed Systems Fall 2021



Problem

- ACO is a probabilistic technique where a set independent software agents (ants) search the solution space, dropping pheromones in their wake. Pheromones increase the likelihood of ants choosing a pathway, and will evaporate over time.
- The goal is for the positive feedback mechanism, encapsulated in a pheremone matrix shared across all the ants, to lead to convergence to the best path.
- Effort has been made to speed up the execution by parallelizing independent runs of the sequential algorithm
- Less work has concentrated on shifting the problem into a multi-agent framework

Results



Process Architecture

1. Ant
2. Graph manager
3. Pheromone manager
4. Ant manager
5. Colony manager

Main Idea

A Distributed Implementation of Ant Colony Optimization

Using the Actor Model of Concurrency

Provides a clear, correct method of solving the TSP

With Speed Benefits over a Serialized Implementation that increase with problem scale.

Methods

- Naturally, the ACO algorithm is iterative. Each round, every ant individually completes a tour of the graph. The paths are mostly random at first, favoring small steps, and incorporating over time the collective intelligence of the colony of ants by weighting paths more heavily if they produced shorter tours on previous rounds by other ants.
- Elixir uses the actor model, where independent computing processes communicate via asynchronous message passing instead of any shared memory.
- The Actor model for synchronization greatly simplifies the shared read and write access of the pheremone matrix among all of the 'ants' by collecting and sharing path costs and pheremone values via messages and allows ants to execute simultaneously.

Results