

Prioritized Foraging using Robot Swarms

Nicolaas J. Taljaard and Andries P. Engelbrecht

Department of Computer Science, University of Pretoria, South Africa

Abstract

Introduction

Swarm Robotics is an entity constructed of simple behavior robots in collaboration, the inspiration for these robot algorithms draw from social insects such as: Bees, ants and termites [Robots-Dorigo]. These insects are reliant on water, food and organization as a source of survival through searching, retrieval and sorting which is the basis of the algorithms in the form of foraging and clustering [Proposal 6].

Honey bees have to explore away from their hive to find resource objects and return it represented as foraging [Proposal 11]. Also clustering in the form of sorting object as modeled after how ant organize their dead dependent on the density of corpses [Proposal 8].

Background

The following selections introduces further information on the field and also the clustering and foraging algorithms that are experimented with.

Swarm Intelligence / Robotics

Swarm Intelligence (SI) as a method of problem solving through the use of agents interacting with their environment as the source of their knowledge [Beni-Robot]. The agents are represented as robots in a collective body known as Swarm Robotics. As collective intelligence of the swarm it raises the social behavior an individual robot, the test algorithms used are independent of an controlling entity to provide guidance to other robots. The independence of this allows robots to base decision on their immediate environment, their current robotic status and ability.

Swarm-based robotics (SR) is defined by Bonabeau *et al* [Robot-Swarm] that it may be able to perform tasks without needing a explicit representation of the environment. These robots are modeled after insect colonies respectively as desert ants for clustering and honey bees for foraging. Each colony has their own internal communication method to achieving their goals and enable the process that is mimicked on the robots.

SI and SR as a self-organized social insects use a communication method call stigmergy [Robot-Swarm], it can be in a form of sematectonic and sign-based communication. Where sign-based is in the form of physical information exchange and sematectonic by means of environment changes. Stigmergy will be used as follow: Direct where the honey bee broadcasts a foraging site by dancing when it arrives at the hive. Also by in-direct for ant cemetery clustering where each ant exams the environment to affect its decision making.

Desert-ant Cemetery Clustering

The desert ant is used to model the ant clustering process, it is chosen for these ants are independent of any direct communication or information sharing. Desert ants do not require any

pheromones or beacons to provide navigation it only uses vision of its immediate environment [12 Jade]. The cemetery cluster is based on the ant that move there died into clusters dependent on the observed environment and the density of object clusters [77 AI]. As for movement decisions it uses random navigation or if detection movement is towards the lowest density [13 Jade]. Once an object is picked up it is carries towards the highest dens array until it satisfies the dropping probability where it is then dropped. This clustering process will aim at improving the performance of foraging due to the closer group of the same priority objects. The abilities of a desert ant is simplistic and does not require many detectors to be added to the robot making it viable and can be cost effective as only a camera is required.

Honey-bee Foraging

The honey-bee is proven by Jade [Jade] to be success foraging algorithm when working with a wider variety of grid configurations and object placing. These bees have three roles assigned to them [Jade 7]: Employed-, unemployed foragers and scouts. Scouts search the environments, once an object successfully foraged depending on its desirability it can dance for the unemployed bees. These bees wait for a dance and employed bees are actively foraging a location. As suggested by Jasen [Jade 14] waiting bees change to scout after a duration of time. An improvement made is during employed foraging, if a higher dens location is detected while following the determined Path Integration (PI) it would rather forage that location. PI is the process where a bee continuously changes its baring vector towards its starting location to be able to share the direction other bees during the dance [Jade 22].

Prioritized Clustering / Foraging

An environment containing prioritized and non-prioritized object throughout may introduce the following: It may occur that prioritized items are encircled or obstructed by non-prioritized

objects. There are two methods to resolve this, by foraging the non-prioritized objects or by cluster the which will separate the object types away from one another. To achieve success and optimal result for the multi-problem both of these solutions is used.

Algorithm Description

Desert-ant cemetery clustering

The ant cemetery clustering algorithm is dependent on the densities of an ants immediate area. A laden ant will drop its payload if the density surpasses the probability barrier. During the scouting process ants will move towards the lowest dens area and the highest during clustering. The process and states are described by Fig.1 as follows:

Fig1

1. A scout searches randomly until a populated area is detected. Upon object detection scouts move towards the lowest dens area populated with objects.
2. An ant will attempt pick-up of an object once arrived at the source area. The pick-up probability is calculated with the density detected from the surrounding objects. If the probability surpasses [PHI] it will be picked up.
3. Otherwise it will continue to follow the density and attempts to find an object that has a low enough density around it.
4. Once laden an ant changes its detection for the most dens area within its perceived vicinity.
5. The same probability calculation is used by scouting, the object will be dropped when it is above [ROU] for successful clustering.
6. After the drop ants avoid the area for the max of its detection range, to avoid the possibility

of being caught optimizing a single cluster instead of clustering outliers.

7. Finally ants resorts to the default scout method.

Honey-bee Foraging

The honey bee algorithm consists of two type of robots and initialized by random as unemployed foragers or scouts. The position of each robot is initialized along the starting column of the grid for an increased spread of robots. A sink where objects are drop is not limited to a single location for all robots but the initialization position of the specific robot is its own drop point. All robots states are depicted in Fig.2 and described as follows:

Fig2

1. A bee in the scouting state does a random walk through the environment looking for the area with it highest density of objects.
2. Once a dens area is detected and the probability of pick-up is greater than [PHI] an object will be loaded and the bee will start foraging.
3. As laden the bee forages the object along its PI back to the drop sink where it will drop the object. If not laden the bee follows the direction vector back towards the previous found location for the measured distance. During the return process a higher dens area may be detected where upon the bee will changes its bearing vector, this is done if the density is greater than 1.4 of the previous determined density.
4. An object is unloaded from a laden bee once it reaches its sink and deposits it. If the bee was in the waiting state before foraging it will return to wait else as an unemployed forager it will continue to forage the site.
5. After a successful deposit the bee calculates the probability of if the found site is success-

ful. If so it will perform a dance for bee in a close vicinity to promote the sites PI as well as its distance.

6. An employed forager may abort after no object where able to be found where upon it will return to the sink and go into a waiting state.
7. A bee in the waiting state will be employed until it detects a dance within its vicinity that is deemed profitable. The bee may also resort to unemployed scouting if it does not detect a dance for a prolonged time t_{max} .

Simulation Configuration

The simulated environment is representation with a 2D grid which has values assigned for every state for robots and the type of object and their own state of carried. Each object and robot fills one index of the grid. Objects can be picked up by robots once on top of an object or move around it as an obstacle to get to higher dens areas.

An object is successfully foraged once it reaches the end of the environment at column zero where object will be dropped and removed from the environment. The adoption to the setup for a mining environment where the sinks will be place at specific locations would be simple. Different environment layouts are used for testing, with this four object distributions: Uniform (Fig), pre-clustered with Lumer-Faieta (Fig), Vein (Fig), Gaussian (Fig).

Fig3

The configuration are setup as follows where each is configuration is simulated 30 times. Scatter layout, the placement of object as names above to test different dispersion resulting from an explosions. Grid sizes representing the number of position units increasing complex-

ity and required simulation time, $S = 50, 100, 200$. The amount of robots used to cluster and forage objects, $c = 10, 30, 50, 70, 100$. Grid coverage as a percentage of objects placed against open locations for movement, $p = 5\%, 20\%, 50\%, 70\%, 90\%$. A ratio representation, $r = 0, 0.2, 0.25, 0.333, 0.5, 0.667, 0.75, 0.8, 1$ as the number of prioritized objects against non-prioritized object. A priority value is used to determine the priority of a populated area calculated per item detected for $t = 0, 0.2, 0.25, 0.333, 0.5, 0.667, 0.75, 0.8, 1$. Honey-bee specific values are used for the allowed waiting time of $t_{max} = 100$, the max time spent foraging is determine by 1.3 times of the distance acquired from the dance. A stagnation condition has been created to prevent endless random walks that is determined from when the last pick up occurred.

Results

Results

Conclusion

Conclusion

References and Notes