Available online at www.pelagiaresearchlibrary.com

Pelagia Research Library

Advances in Applied Science Research, 2011, 2 (3): 461-467

ISSN: 0976-8610
CODEN (USA): AASRFC

Ant Colony Optimization for Co-operation in Robotic Swarms

Lalit Kumar Behera[*] and Aparna Sasidharan

School of Computing Science and Engineering, VIT University, Vellore, Tamil Nadu, India

ABSTRACT

The swarm robotic concept has become a topic of extensive research in hazardous environments where a fault tolerant, robust and energy efficient approach is needed. In this paper we present behavior based algorithm for target navigation and swarm distribution over hazardous targets. This work is inspired by ant colony models. Our goal is to derive a heuristic algorithm for target localization and decentralized swarm distribution. Target localization is based on ant colony behavior combined with random walk.

Keywords: hazardous environment, behavior based, target localization, random walk.

INTRODUCTION

Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems. The essential trait of ACO algorithms is the combination of information about the structure of a promising solution with information about the structure of previously obtained good solutions. The metaheuristic part permits the low-level heuristic to obtain solutions better than those it could have achieved alone, even if iterated. Usually, the controlling mechanism is achieved either by constraining or by randomizing the set of local neighbor solutions to consider in local search, or by combining elements taken by different solutions.

A typical ant colony system can consist of many components and subsystems, such as for ants, creating path, search path according to pheromone value (e.g. neighbor), search for beacons, updating pheromone value.

Motivation

Emerging technologies in micro machining hold the promise of creating extremely small robots. Although limited in size and power, such robots can work together in large numbers to conceivably accomplish a wide range of significant tasks including surveillance, reconnaissance, hazard detection, path finding and payload conveyance.
So we have to build a system where coordination and interaction with very large numbers of robots involves issues not encountered when dealing with one or a few robots. Coordination

schemes that require unique identities for each robot, explicit routing of point-to-point communication between robots, or centralized representations of the state of an entire swarm can be overwhelmed when dealing with extremely large numbers. To assure that our multi-robot system is both scalable to large robot swarms and tolerant to individual robot failures, we focus on techniques that effectively limit each robot's interactions to its own local neighborhood. The previous systems e.g. land mark localization, which needs land marks for mapping in general. Similar like landmark localization, Ant *Colony Optimization System* by which the same level of work can be achieved. We are inspired by techniques used by ants and termites for communication and coordination.

Then we are concentrating to develop the system using multi-agent swarm based concept. Because multi-agent systems have been contributing to the development of the theory and the practice of high-speed, mission-critical, decentralized information systems where mutual interdependencies, dynamic environments, uncertainty, and sophisticated control play a major role.

Background

Swarm Intelligence-based Applications are as complex interactive virtual environments generations in movie industries, cargo arrangement in Airline companies, route scheduling in delivery companies, routing packets in telecommunication networks, power grid optimization control, data clustering, data routing in Sensor Network, Unnamed vehicles controlling in the U.S. military's, planetary mapping and micro-satellite controlling in NASA.

Swarm Robotics Surveillance systems: Surveillance systems are often needed in areas too hostile or dangerous for a direct human presence. One major problem is the control and coordination of multiple cooperating robots. We have looked to the distributed control strategies found in nature in the form of social insects as an inspiration for new control schemes for the coordination and control of large-scale distributed robotic systems. The research goal is to manipulate the interactions of multiple small, low-cost robots, with a limited range of local communication ability, to collaboratively search and engage tasks in an unknown large-scale hostile area.

Related Work

Swarm navigation algorithm using olfactory-based steering for terrain surveying by spatial concentration of odor fields is presented and the ability to estimate the location of odor source in continuous odor fields from sparse data taken with a group of mobile sensing robots is addressed. Using a cooperative approach for positioning system, this method addresses the problem of localization of the robots [1].

Goss and co-workers proposed a model for explaining the foraging behavior of ants was the main source of inspiration for the development of ant colony optimization. In ACO, a number of artificial ants build solutions to an optimization problem at hand and exchange information on the quality of these solutions via a communication scheme that is reminiscent of the one adopted by real ants. In ACO, a number of artificial ants build solutions to an optimization problem and exchange information on their quality via a communication scheme that is reminiscent of the one adopted by real ants [2].

A novel face recognition system was built to detect faces in images and video tracks faces, recognizes faces from galleries of known people using Genetic and Ant colony Optimization algorithm is proposed. This system is caped with three steps. Initially pre-processing methods are applied on the input image. Consequently face features are extracted from the processed image by ANT Colony Optimization (ACO) and finally recognition is done by Genetic Algorithm (GA. The proposed method is tested on a number of test images [3].

An algorithm for discrete optimization which took inspiration from the observation of ant colonies foraging behavior, and introduces the ant colony optimization (ACO) meta-heuristic. The basic biological findings on real ants are overviewed, and their artificial counterparts as well as the ACO meta-heuristic are defined, and a number of applications to combinatorial optimization and routing in communications networks are described [4].

Hybrid algorithm is proposed to solve combinatorial optimization problem by using Ant Colony and Genetic programming algorithms. Evolutionary process of Ant Colony Optimization algorithm adapts genetic operations to enhance ant movement towards solution state[5].

A new general-purpose heuristic algorithm which can be used to solve different combinatorial optimization problems. The new heuristic has the following desirable characteristics: versatile- in that it can be applied to similar versions of the same problem, robust, as it can be applied with only minimal changes to other combinatorial optimization problems such as the quadratic assignment problem (QAP) and the job-shop scheduling problem (JSP), population based approach as it allows the exploitation of positive feedback as a search mechanism [6].

MATERIALS AND METHODS

To construct a swarm based multi-agent robot which is fault tolerant, robust and energy efficient which is able to localize the target in a hazardous environment, and can decentralize the swarm distribution. The swarm based localization involves ant colony and honey bee behavior models. This technique uses Heuristic algorithm for target localization and decentralized swarm distribution. Target localization is based on ant colony behavior combined with random walk. Collective decision making and distribution of a swarm are based on honey bee behavior. The goal of the Robot is to reliably address all of the fundamental challenges: Accurate positioning and map-building, long-range, adaptive communication, reactive obstacle avoidance and path planning, high-speed waypoint navigation without GPS, self-status awareness and health monitoring, cognitive problem solving to accomplish high-level tasking, human presence detection and tracking, plug-and-play adaptability to different sensor suites

Architecture

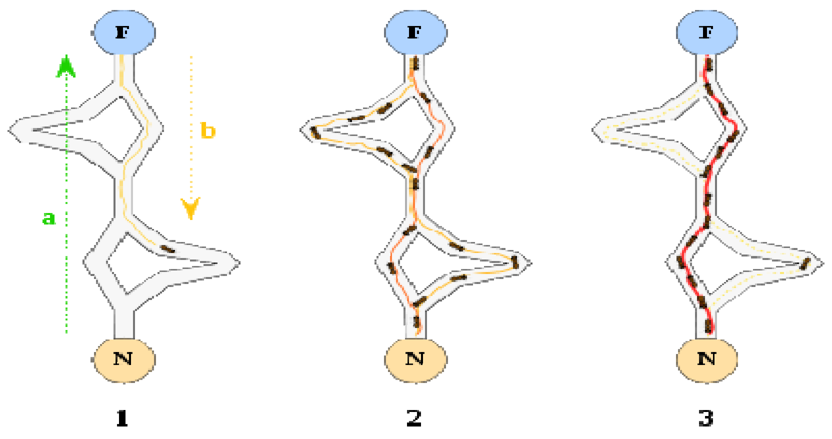

Figure 1: Architecture diagram for ant colony algorithm

Functional Specification

A group of simple robots that can only communicate locally and operate in a biologically inspired manner

- **Metaheuristic:** A metaheuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality.

- **Artificial Ants:** A set of software agents are stochastic, based on the pheromone model. Pheromones are used by real ants to mark paths. Ants follow these paths (i.e., trail-following behaviors). They incrementally build solutions by moving on a graph. Constraints of the problem are built into the heuristics of the ants.

- **Update Pheromones:** It is a process for modifying the pheromone trails. It may be modified by increase (Ants deposit pheromones on the nodes (or the edges)), decrease (Ants don't replace the pheromones and they evaporate).

➢ Increasing the pheromones increases the probability of paths being used (i.e., building the solution).
➢ Decreasing the pheromones decreases the probability of the paths being used (i.e., forgetting)

- **Daemon Actions:**

o It is used to implement larger actions that require more than one ant
o E.g.: Perform a local search, Collection of global information

Client Server Diagram

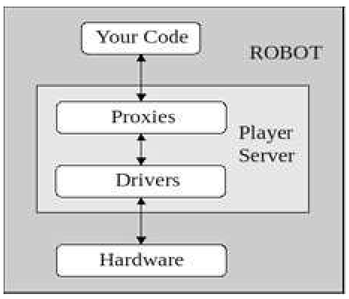

Figure 2: The server/client control structure of Player when used on a robot (Player/Stage manual)

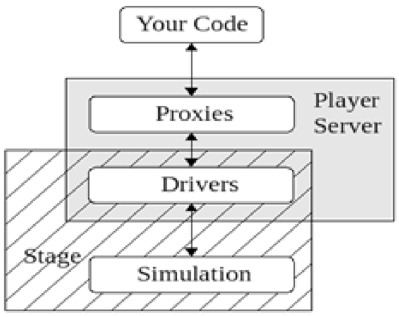

Figure 3: The server/client control structure when used as a simulator (Player/Stage manual)

As illustrated in the Figure 2 and Figure 3, following steps:

Interfaces, Drivers and Devices
Drivers are pieces of code that talk directly to hardware. The drivers are specific to a piece of hardware so, say, a laser driver will be different to a camera driver, and also different to a driver for a different brand of laser. This is the same as the way that drivers for graphics cards differ for each make and model of card. Drivers produce and read information which conforms to an “interface".

Interfaces are a set way for a driver to send and receive information from Player. Like drivers, interfaces are also built in to Player. They specify the syntax and semantics of how drivers and Player interact.

A device is a driver that is bound to an interface so that Player can talk to it directly. This means that if you are working on a real robot that you can interact with a real device (laser, gripper, camera etc) on the real robot, in a simulated robot you can interact with their simulations.

Implementation
The important parts of the ant colony are *neighbor checking and pheromone updating.*

Algorithm
Input: No of robot and no of iteration to find the path.
Output: Find the path for the robot using highest pheromone value.

Methods
Step 1: Read the number of iteration, for check how many path have to calculate at the time of set path
Step 2: Read the number of robot to find the path accordingly the highest pheromone value
Step 3: Initialization of the Matrix

```
a. Fix the start position (1,1) value as 0
b. Fix all the boundary value as B
c. Fix the value of the position of robots as X,Y and Z
d. Fix rest of the position value as zero (0)
```

Step 4: Choose the Path for the First Robot (X)

```
a. Start from initial position (1,1)
b. Check all the value of the neighbors of that position is X or not,
       if X then stop
c. Check all value of the neighbor is 0 or not
       if 0 then select randomly anyone and set the position for next calculation
       else fail
d. Continue step b and c up to destination value X
e. Count the number of move to reach the destination and store it
f. Select the path which has shortest move as the final path for the first robot (X)
g. Increment the value of all the selected position in the path for the first robot (X)
```

Step 5: Choose the Path for the Second Robot (Y)

```
a) Start from initial position (1,1)
b) Check all the value of the neighbors of that position is Y or not,
   if X then stop
c) Check all value of the neighbor is 0 or not
```

if 0 then select randomly anyone and set the position for next calculation else fail

d) Continue step b and c up to destination value Y
e) Count the number of move to reach the destination and store it
f) Select the path which has shortest move as the final path for the first robot (Y)
g) Increment the value of all the selected position in the path for the first robot (Y)

Step 6: Choose the Path for the Third Robot (Z)

a) Start from initial position (1,1)
b) Check all the value of the neighbors of that position is Z or not,
   if Z then stop
c) Check all value of the neighbor is 0 or not
   if 0 then select randomly anyone and set the position for next calculation else fail
d) Continue step b and c up to destination value Z
e) Count the number of move to reach the destination and store it
f) Select the path which has shortest move as the final path for the first robot (Z)
g) Increment the value of all the selected position in the path for the first robot (Z)

Step 7: Finally the path has been set for X, Y and Z

Step 8: Check the Path of the Next Robot with the Highest Pheromone Value

a. Check all the neighbor of the initial position (1,1),
   if the neighbor is not X, Y, Z , 0 & B then store the particular location & value of the position
b. Select the highest value neighbor for next move
c. Increment the value of that position by one (1) which is in the selected path for the robot
d. Repeat steps a, b and c up to reach the destination in-between X, Y and Z

RESULTS AND DISCUSSION

The above algorithm will be built in the simulator. The simulation results can be seen in Figure 4 After the code build, it will be compiled and then one .world file will be created which is comprised of all proxies. That world file will be run to show the following result.

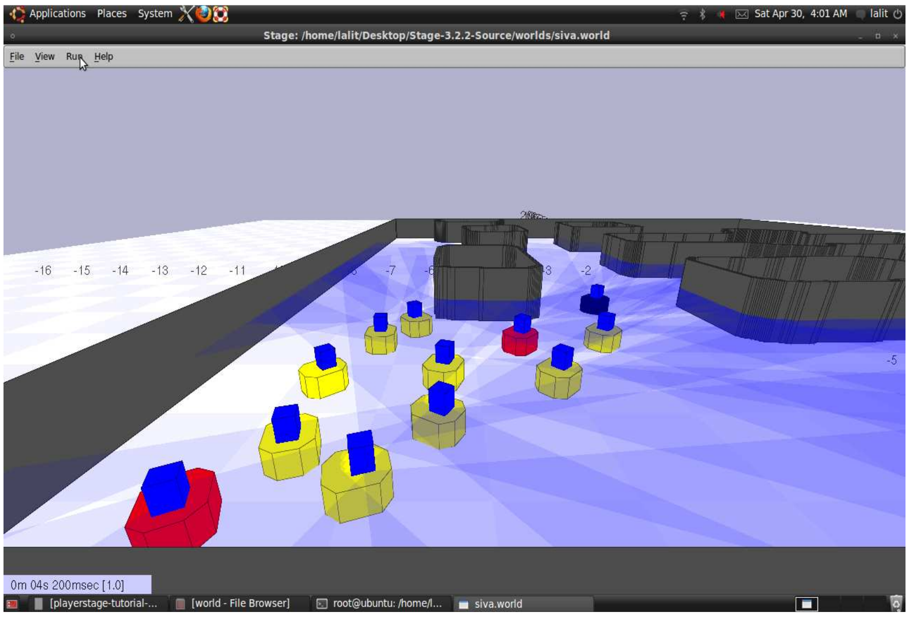

Figure 4: Movements of robots (perspective view)

CONCLUSION

In this paper we introduced the (ACO) Ant Colony Optimization which has been and continues to be a fruitful paradigm for designing effective combinatorial optimization solution algorithms. ACO one of the most successful paradigm in the metaheuristic area. This overview tries to propose both introductory elements and pointers to recent results, obtained in the different directions pursued by current research on ACO. New results will both improve those outlined here and widen the area of applicability of the ACO paradigm.

Acknowledgements
I am grateful to Arijit Das, is a Final year student belongs to M.Sc Computer Science of V.I.T University for helping me in implementation portion of this article.

REFERENCES

[1] Ali Marjovi, Joao Nunes, Pedro Sousa, Ricardo Faria, Lino Marques, *IEEE International Conference on Robotics and Automation*, pp. 4958-4963, **2010**
[2] Marco Dorigo, Mauro Birattari, and Thomas Stutzle, "Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique", IRIDIA – Technical Report Series, 1-12, **2006**
[3] S.Venkatesan and Dr.S.Srinivasa Rao Madane, *International Journal of Innovation, Management and Technology*, Vol. 1, 5, **2010**
http://www.ijimt.org/papers/82-M477.pdf
[4] Marco Dorigo and Gianni Di Caro and Luca M. Gambardella, "Ant Algorithms for Discrete Optimization" *in Artificial Life*, Vol.5, No.3, 137-172, **1999**
[5] Nada M. A. Al Salami, "Ant Colony Optimization Algorithm", *UbiCC Journal*, Vol. 4, 3, **2009**
http://www.ubicc.org/files/pdf/10_336.pdf
[6] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni, "The Ant System: Optimization by a colony of Cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, Vol.26, 1, pp.1-13, **1996**