Engineering Applications of Artificial Intelligence 22 (2009) 431–441

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Collective robotic search using hybrid techniques: Fuzzy logic and swarm intelligence inspired by nature

Ganesh K. Venayagamoorthy *, Lisa L. Grant, Sheetal Doctor

Real-Time Power and Intelligent Systems Laboratory, Missouri University of Science and Technology, Electrical and Computer Engineering, 132 Emerson Electric Co. Hall, Rolla, MO 65409, USA

ARTICLE INFO

Article history:
Received 24 April 2005
Received in revised form
7 August 2007
Accepted 4 October 2008
Available online 26 November 2008

Keywords:
Collective robotic search
Fuzzy logic
Fuzzy swarm
Greedy search
Particle swarm optimization
Swarm fuzzy
Swarm intelligence

ABSTRACT

This paper presents two new strategies for navigation of a swarm of robots for target/mission focused applications including landmine detection and firefighting. The first method presents an embedded fuzzy logic approach in the particle swarm optimization (PSO) algorithm robots and the second method presents a swarm of fuzzy logic controllers, one on each robot. The framework of both strategies has been inspired by natural swarms such as the school of fish or the flock of birds. In addition to the target search using the above methods, a hierarchy for the coordination of a swarm of robots has been proposed. The robustness of both strategies is evaluated for failures or loss in swarm members. Results are presented with both strategies and comparisons of their performance are carried out against a greedy search algorithm.

1. Introduction

In recent years, a lot of research is focused on collective robotic applications (Van Dyke Paranuk, 2005; Zhang and Wunsch, 2003a, b). Robots have been used for numerous applications where human intervention is not feasible. The building of intelligent robots with a number of sensors for various parameters is expensive. In applications such as radioactivity detection, firefighting and landmine detection, the robots need to be dispensable. A large number of robots allow for redundancy and increase the robustness of the swarm. An expensive robot may be able to achieve the task but its failure can prove to be costly and dangerous in mission critical applications. By building a swarm of robots with elementary features, the same task can be achieved for a lower cost and increased reliability. Multiple robots used for search applications also have an advantage of larger coverage of the search space and its simplicity of implementation.

Planning the motion of these robots in a search space differs from application to application. Motion planning of the robots differs in obstacle-laden environments (Foux et al., 1993; Kim and Butler, 1995). The navigation of the robots from the start to the destination in an unknown environment that requires the use of sensors and a method to explore the area with the inputs of these sensors (Kim and Butler, 1995). Fuzzy logic has been a widely used method for these applications (Kim and Butler, 1995; Zhang and Wunsch, 2003a, b; Dadios and Maravilla Jr., 2002). Other methods explored in literature include navigation based on intelligent data carrier systems (Konishi et al., 1999) or geometric algorithms (Sharir, 1989), etc. In the case of multiple robot navigation, basic algorithms like particle swarm optimization (PSO), evolutionary strategies, genetic algorithms, etc, have been explored for this purpose.

Swarm intelligence is a method derived from the study of a flock of birds, school of fish, a colony of ants or a swarm of bees. These methods take into account the natural behavior of social animals that carry out a certain task collectively. Fuzzy logic has been used in a number of ways for robotic navigation. This paper explores the performance of a fuzzy system by introducing the social interaction exhibited in swarm intelligence. This paper presents the integration of fuzzy logic with swarm intelligence specifically the PSO.

The integration of fuzzy logic and swarm intelligence has been carried using two different approaches. The first method is based on the canonical PSO algorithm that uses a fuzzy term to replace a part of the PSO dynamics. This method is referred to as the *fuzzified swarm of robots* in this paper. The second method is based on a swarm of robots that use fuzzy logic controllers, referred to as

* Corresponding author. Tel.: +1 573 3416641; fax: +1 573 3414532.
E-mail addresses: gkumar@ieee.org (G.K. Venayagamoorthy), lisagrant@ieee.org (L.L. Grant), skdkv6@mst.edu (S. Doctor).

0952-1976/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2008.10.002

swarm-fuzzy controllers in this paper. In this method, the robots use fuzzy logic and are guided towards the desired target locations by trying to move towards the robot with the best fitness in the swarm.

The rest of the paper is outlined as follows: Section 2 describes the collective robotic search architecture and the search problem considered in this paper. Sections 3 and 4 describe PSO and fuzzy logic, respectively. Section 5 describes the experimental setup for the implementation of the collective robotic search of Section 2. Sections 6 and 7 describe the implementation of the two different approaches. And finally, Section 8 presents the results for the collective search using the hybrid swarm-fuzzy approaches compared with a greedy search.

2. Collective robotic search

There are various applications as mentioned in the previous section, which require the use of multiple robots or a team of robots. This paper proposes a three-tier architecture for the implementation of a collective robotic search as shown in Fig. 1. The aim of the robotic search application is to identify target locations. The nature of the targets varies in different applications. For the study in this paper, the targets are taken to be stationary light sources. The robots are assumed to have light sensors and they sense the total light intensity at their current locations irrespective of the location and number of light sources.

The robots are randomly deployed in the problem search area. Since collective robotic search is targeted to applications where human intervention is not possible, the random deployment can be carried out for example by dropping the robots to the search area of interest by an aircraft.

As seen in Fig. 1, the search area is divided into a number of swarms and the operation of each swarm is independent of each other. In environments with extreme topology/terrain, this division of the search area into multiple swarms makes sure that the entire search space is covered and independently the targets are searched out, thus performing a fast, efficient and parallel search. Not all robots may have access to the entire search space depending upon their initial position of deployment and the severity of terrain, which they will have to traverse, such as being restricted to a certain area by a natural boundary such as a body of water or mountain range. In addition, with divided search space, the communication range of the robots does not have to be too long. Thus, transmission power can be minimized. The concepts of the neighborhood level and global level are explained in the next section with respect to PSO.

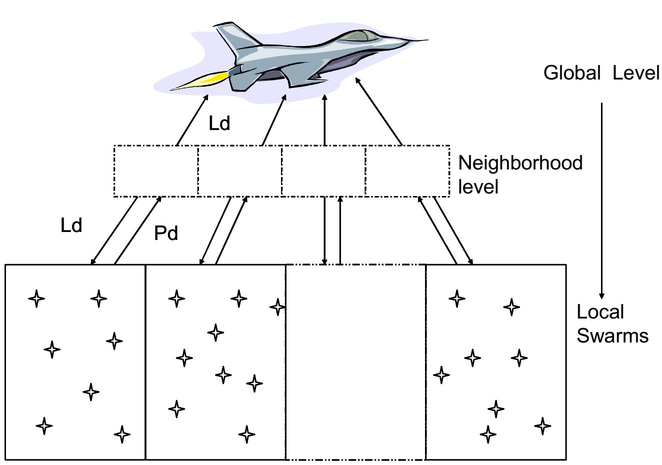

Fig. 1. A three-tier architecture for the collective robotic search. Robots are randomly deployed from an airplane.

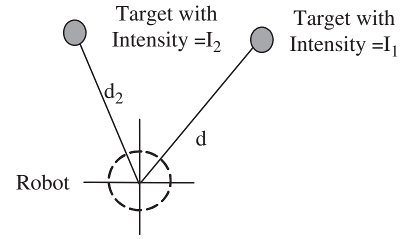

Fig. 2. Graphical representation of sensor reading at the robot.

At the ground/local level, all the robots can pick up sensor readings from all the targets within a certain range irrespective of the location of targets. This means that a robot can pick up a sensor reading from a target lying in another neighborhood swarm. Targets are assumed to be stationary for this application. Fig. 2 shows a graphical representation of the sensor readings. Eq. (1) shows the total sensor reading at the robot as considered for this paper. The sensor readings of each robot are used to locate the target. In the first few iterations of the search, exploration of the swarm area is carried out and then the search is focused gradually towards locating a primary target and readings from secondary targets fade away or shadowed:

$$\text{Intensity reading at the sensor node} = I_1/d_1^2 + I_2/d_2^2 \tag{1}$$

3. Particle swarm optimization

PSO is relatively a new concept reported by Kennedy and Eberhart (2001), in 1995. PSO has been applied for target tracing by autonomous communicating bodies (Gesu et al., 2000). A problem space is initialized with a population of random solutions in which it searches for the optimum over a number of generations/iterations and reproduction is based on prior generations. The concept of PSO is that each particle randomly searches through the problem space by updating itself with its own memory and the social information gathered from other particles. In this paper, the PSO particles are referred to as robots and the local version of the PSO algorithm is considered in the context of this application (Kennedy, 1999).

Within a defined sensing area or a swarm as in Fig. 1, there is a population of robots. Each robot is randomized with a velocity and 'flown' in the problem space. They have memory and they are able to keep track of the position that resulted in the highest sensor readings. This position is referred to as 'P_{best}'. Thus, each robot has a 'P_{best}'. The best of all these P_{best} values is defined as the local best position 'L_{best}' with respect to the target. The velocities and positions of these robots are constantly updated until they have all converged at the target location. Thus, in terms of memory requirements, PSO requires only 2 values (other than the velocity and position from the previous iteration), P_{best} and L_{best}.

Fig. 3 gives the vector representation of the PSO in a two-dimensional search space. The stars represent the particles/robots and the circle represents the target. In Fig. 3, vectors V_{pd} and V_{gd} represent the effect of P_{best} and L_{best} on the robots, respectively. The basic PSO velocity and position update equations are given by (2) and (3) respectively. To correlate Fig. 3 and velocity equation, (2), the term V_{pd} in the figure represents '$c_1 \times \text{rand} \times (P_{\text{best}} - P_{\text{cur}})$' in (2) and the term V_{gd} in the figure represents '$c_2 \times \text{rand} \times (L_{\text{best}} - P_{\text{cur}})$' in (2). The constants w, c_1 and c_2 in (2) are called the quality factors:

$$V_{\text{new}} = w \times V_{\text{cur}} + c_1 \times \text{rand} \times (P_{\text{best}} - P_{\text{cur}}) \\ + c_2 \times \text{rand} \times (L_{\text{best}} - P_{\text{cur}}) \tag{2}$$