A Multiagent Algorithm for finding Multiple Noisy Radiation Sources with Spatial and Communication Constraints

Abstract

In this paper, we will adapt concepts from the well-known Particle Swarm Optimization algorithm to derive a multiagent behavioral algorithm to control mobile sensing robots searching for one or more radiation source(s) in a noisy environment with spatial and communication constraints. In the following sections, we will define the problem, state the assumptions and derive the multiagent algorithm. At last we will refer to the considerations that must be taken when applying this algorithm in real life, and the future work directions in this field.

Key Words: Mobile Wireless Sensor Nodes, Multiagent Behavioral Algorithms, Swarm Intelligence.

1. Introduction

Mobile communicating robots –especially mobile sensor nodes- are utilized in tasks that require operating in remote dangerous environments with extreme conditions, where human interaction may be limited if not impossible.

The advantage of mobile sensor nodes is that they are cheap therefore large numbers of them can be employed in risky tasks without the worry about damaging or losing some of them during the operation [1].

The communication range and power consumption are the usual limitations of mobile sensor nodes, and must be taken into consideration [1].

That raises the need for power-concerned multiagent algorithms and protocols that are not affected with the performance of single agents as much as the performance of the total society of agents.

2. Problem Definition

We will consider here the task of finding one or more source(s) of radiation in an unknown noisy environment. A large number of cheap mobile sensor nodes will cooperate together in order to locate that/these source(s). These nodes can explore most parts of the environment; however, there are some barriers that block the movement of these nodes.

3. Solution Assumptions

We will make these assumptions in this context:

- A mobile sensor node can determine the location of a radiation source if it is near enough from it.
- Each node has a sensor capable of measuring the intensity of radiation –with some noise- at the node location on request.
- Each node can broadcast messages reliably with other nearby nodes.
- Each node has a maximum velocity, and can move in the environment with a specified magnitude and direction instantaneously, i.e., neither acceleration nor torque is assumed.
- Every node can keep track of its relative position with respect to a common virtual point.
- Each node can identify a barrier when it encounters it.
- Nodes can avoid collision with each other.
- Source(s) location and radiation are static not varying.

4. Proposed Algorithm

4.1. Variables

Each node keeps track of these variables:

- Current location X and the radiation intensity at it $f(X)$.
- Current velocity V.
- Best visited location P and the radiation intensity at it $f(P)$.
- Best known location G and the radiation intensity at it $f(G)$.

X, P, G are 2D vectors representing relative positions with in the environment.
V is 2D vector representing relative velocity.

4.2. Constants

There are constants in each node:

- V_m is the minimum velocity of an agent.
- w the current velocity parameter.
- p the best visited parameter

- g the best known parameter.

These parameters can be selected in the same way as selecting parameters of the particle swarm optimization algorithm [2].

4.3. Agent algorithm

1- Initially, the agent is placed at random position with random velocity, and $G=P=X$ and $f(G)=f(P)=f(X)$.
2- Select random numbers $r1,r2 \sim U(0,1)$
3- Calculate $T= w*V + r1*p*(P-X) + r2*g*(G-X)$
4- Normalize T to V_m if it is smaller in magnitude.
5- Apply T and store it in V.
6- If a barrier is preventing motion then try moving in a random direction with random magnitude.
7- The agent calculates current position X and measures current radiation intensity $f(X)$.
8- If the agent reaches a radiation source, then abort the algorithm.
9- Send=false
10- If($f(X)>f(P)$) then
    a. $P=X$
    b. $f(P)=f(X)$
    c. Send=true
11- If($f(P)>f(G)$) then
    a. $G=P$
    b. $f(G)=f(P)$
    c. Send=true
12- If(Send)
    a. Broadcast $message\ (G\ ,\ f(G))$ to all nearby agents.
13- Goto 2.

On receiving a message from other agent:
1- If($message.f(G) >= f(G)$)
    a. $G=message.G$
    b. $f(G)=message.f(G)$

4.4. Convergence

Exactly like particle swarm optimization algorithm, the algorithm does not grantee convergence [3].

4.5 Differences from ordinary PSO Algorithm

Although there are big similarities between the proposed algorithm and the ordinary particle swarm optimization algorithm, the major difference is that: unlike ordinary PSO, there are not shared variables between agents to store best swarm position [4].

We have overcome this problem as follows:
- Once the agent updates either its best visited position or best known position, it broadcasts its best known position to nearby agents.
- That forces the best position of the swarm to propagate through most of the agents in sent messages.
- Also we are forcing a minimum velocity V_m in the calculated direction to prevent the agent from stopping, that happens when the agent cannot communicate with successful agents.

5. Experiments

We have developed an open source project called "*Robots Routing using Swarm Intelligence (RRSI)*" to simulate and test the algorithm with an emulated environment with different parameters and various conditions [5].

"*RRSI*" is written in C# with Microsoft .NET Framework 4.0, and is hosted on Code Plex – an open source community website.

This project allows the creation of barriers, agents and sources (Gaussian radiations), and locating them in an environment.
It also allows selection of either ideal environment or a one with noise (Gaussian white noise for example)

For more detail, please visit http://rrsi.codeplex.com/

6. Practical Considerations

Here we will discuss how to apply this algorithm in practice.

To determine the agent's self-position, either the agent must be able to measure its own movement, or the agent must be equipped with a positioning module (GPS for example)

Reliable communication between nearby agents can be achieved by well- known protocols.

The agent can be equipped with sensors to prevent it from colliding with the environment barriers and with other agents in the environment.

The agent must be able with some means to detect nearby source(s), this can be achieved with short range sensor for example.

The agents may need to memorize the path they went through, in case if we want to collect them back (to use them in another operation).

The agent may decide to return to the base (through memorized path) in case it is on low power.

7. Future Work

More work can be done to determine initial configurations and algorithm parameters that result in faster convergence.

Also we may need to consider the case of moving sources or varying power sources.

We may also need to modify the algorithm to search for other sources after the first source was found.

At last, more simulations and real world experiments may be needed to determine the performance characteristics of this algorithm.

8. References

[1] Wikipedia "*Wireless sensor network*" http://en.wikipedia.org/wiki/Wireless_sensor_network

[2] Magnus Eric Hvass Pedersen, 2010, "*Good Parameters for Particle Swarm Optimization*", Hvass Laboratories, Technical Report no. HL1001, 2010

[3] Wikipedia "*Particle swarm optimization*" http://en.wikipedia.org/wiki/Particle_swarm_optimization

[4] James Kennedy and Russell Eberhart, "*Particle Swarm Optimization*", Purdue School of Engineering and Technology.

[5] "*Robots Routing using Swarm Intelligence (RRSI)"* project on CodePlex http://rrsi.codeplex.com/