

Multi-Robot, Multi-Target Particle Swarm Optimization Search in Noisy Wireless Environments

Kurt Derr
 Idaho National Laboratory
 2525 Fremont Avenue
 Idaho Falls, ID 83415, USA
derr5843@uidaho.edu
Kurt.Derr@inl.gov

Milos Manic
 Department of Computer Science
 University of Idaho at Idaho Falls
 1776 Science Center Dr., Ste.306
 Idaho Falls, ID 83402, USA
misko@ieee.org

Abstract

Multiple small robots (swarms) can work together using Particle Swarm Optimization (PSO) to perform tasks that are difficult or impossible for a single robot to accomplish. The problem considered in this paper is exploration of an unknown environment with the goal of finding a target(s) at an unknown location(s) using multiple small mobile robots.

This work demonstrates the use of a distributed PSO algorithm with a novel adaptive RSS weighting factor to guide robots for locating target(s) in high risk environments. The approach was developed and analyzed on multiple robot single and multiple target search. The approach was further enhanced by the multi-robot-multi-target search in noisy environments. The experimental results demonstrated how the availability of radio frequency signal can significantly affect robot search time to reach a target.

Keywords—fitness, Particle Swarm Optimization (PSO), Received Signal Strength (RSS), wireless

I. INTRODUCTION

Many potential applications exist for the deployment of an army of small robots [1]-[3]. Teams of small robots may potentially provide solutions to surveillance, monitoring, and search and rescue operations, removing the need for human intervention in dangerous areas. One such task is exploration and search of an unknown environment such as an earthquake damaged building where each robot has a sensor(s) that can detect heat, light, sound, or other, and communicate wirelessly with other robots. Other examples may include bomb or radiation detection and accidents with hazardous materials.

Small robots have limited mobility range, limited energy availability, and reduced sensing, communication, and computation capability due to their size and power constraints. Because of these constraints resources must be distributed across multiple robots which must work in unison to accomplish a mission. Specialization of robot functions and collaboration amongst the deployed robots is employed to deal with these constraints. Each low-cost robot is equipped with limited sensing and processing capabilities for specific tasks; e.g., mapping the surroundings, providing feedback for human operators, or carrying sensors for the

mission.

PSO techniques are algorithms used to find a solution to an optimization problem in some search space. PSO has been used for hazardous target search applications, such as landmine detection, fire fighting, and military surveillance, and is an effective technique for collective robotic search problems [4]-[5]. When PSO is used for exploration, this algorithm enables robots to travel on trajectories that lead to total swarm convergence on some target. Particles may represent small robots exploring some search space and attempting to find a target, such as the source of a chemical odor, based on the robots sensor readings. The robot sensor, in this example, must be able to detect the residue given off by the chemical odor.

Two basic approaches to controlling multiple robots to achieve collective tasks are centralized control and distributed control [6]. The PSO algorithm can work in both centralized control and distributed control scenarios. In centralized control the robots are organized in a hierarchical fashion similar to the military; e.g. teams of robots are controlled by designated robot leaders which are controlled by the head robot for the entire swarm. The robots send pose information to the head robot which executes the PSO algorithm and returns new directional information to each robot. In decentralized control, each robot operates on local information but works toward accomplishing a global goal. A decentralized PSO algorithm is used in this paper for robots to find targets at unknown locations in an area of interest.

Some issues in design and implementation of an unsupervised distributed PSO algorithm for target location include robot dispersion and deployment [7]-[9], localization [10], obstacle avoidance, overshooting targets, effect of particle neighborhood sizes on performance, dynamic environments where emissions from targets vary in both time and space, and scalability. This paper addresses overshooting targets and varying radio frequency emissions from targets. No global information such as GPS or a map is presumed to be available to the robots. Each robot is equipped with sensors to detect the target. Additionally, the robots have 802.11 wireless communications capabilities to communicate directly with other robots within wireless range to share information.

The focus of this paper is to present a distributed PSO approach [11] with wireless augmentation for collective

robotic search in a hazardous environment. The paper is organized as follows. Section 2 discusses related work, section 3 reviews the classic PSO algorithm, Section 4 describes the PSO algorithm with wireless augmentation that each robot uses to find a target in the search space, and Section 5 presents our conclusions.

II. RELATED WORK

Game theory has been applied to multi-robot searching for a target where partial information about the environment is known [12]. The search region is partitioned into small areas where a prior probability for the target is given for each area. In the problem described in this paper, no a priori information such as this is provided.

Frontier based exploration has been used to map unknown office environments using multiple robots [13]. Each robot maintains its own map, cooperates in sharing information with other robots, and makes its own decision as to where to navigate. However, this is strictly a mapping approach and no specific targets are being sought out.

Map merging techniques have been used to combine maps obtained by multiple robots from exploring different parts of an environment [14]. Regions of overlap between different maps are identified for stitching the individual maps together. The goal is to obtain a map of an unknown area but not to locate a specific target within the area.

Other authors have used variants of PSO algorithms with different problem constraints. PSO approaches have been explored in noisy environments [15], optimization of the search PSO parameters [16], and plume detection and traversal [17]. Kennedy [18] has a comprehensive survey of PSO algorithms and applications.

III. CLASSIC PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

Particle Swarm Optimization [19] is an evolutionary optimization technique that differs from other evolutionary approaches, such as genetic algorithms, in that there are no crossover and mutation operators and the entire population of particles is maintained throughout the search or exploration operation. Each particle, or member, flies around some multidimensional search space at a velocity and position that is updated based on both the particle's experience and that of the swarm (the collection of particles). A particle may share information with other particles or the entire swarm during an iteration of the algorithm, effectively influencing other particles' search of an area. The basic pseudo code for the classic PSO algorithm taken from [15]-[16] is:

Step 1: Initialize population with random positions and velocities

Step 2: Evaluate – compute fitness of each particle in swarm

Step 3: Do for each particle in swarm {

Step 3.1: Find particle best (*pbest*) – compute fitness of particle.

If current *pbest* < *pbest*

Pbest = current *pbest*

Pbest location = current location

Endif

Step 3.2: Find global best (*gbest*) – best fitness of all *pbest*

Gbest location = location of min(all *pbest*)

Step 3.3: Update velocity of particle per equation 2

Step 3.4: Update position of particle per equation 3

Step 4: } Repeat steps 3.1 through 3.4 until termination condition is reached.

A population of particles is created and then evaluated to compute their fitness' and find both the particle and global best from the population/swarm. The particle best, *pbest*, represents the best fitness value for a particle up to that moment in time, whereas the global best represents the best particle in the entire swarm. A global best, *gbest*, or a local best, *lbest*, are used for a single neighborhood or multiple local neighborhoods, respectively. When multiple local neighborhoods are used each particle stays within their local neighborhood for all iterations of the program.

A fitness function that may be used by the PSO algorithm is based on the Euclidean distance from a particle to a target. The coordinates of the particle are p_x and p_y , while the coordinates of the target are t_x and t_y as shown in equation 1.

$$fitness = \sqrt{(t_x - p_x)^2 + (t_y - p_y)^2} \quad (1)$$

The velocity of a particle is computed as follows [5]:

$$v_{n+1} = w_i v_n + c1 * r1 * (pbest_n - p_n) + c2 * r2 * (gbest_n - p_n) \quad (2)$$

where parameters used are:

w_i = inertia coefficient

v_n = current particle velocity

$r1$ = uniformly distributed random number [0:1]

$r2$ = uniformly distributed random number [0:1]

$c1, c2$ = acceleration constants

The position of the particle is based on its previous position, p_n , and its velocity over a unit of time:

$$p_{n+1} = p_n + v_{n+1} \quad (3)$$

Each particle is accelerated towards its *pbest* and *gbest* locations using acceleration values $c1$ and $c2$, respectively, at each time step. Typical values for parameters are $w_i=0.9$, and $c1=c2=1$ [20]- [21]. Maximum velocities for some small robots noted in the literature are 20 cm/sec [2], 100 cm/sec [22], and 1 m/sec [4]. A large w_i favors global search while a small w_i favors local search.

The search process is terminated when a predetermined number of iterations are reached, or the minimum error requirement is reached for all particles, such as a particle coming within a specific distance of a target.

IV. MULTI-ROBOT SEARCH USING PSO WITH WIRELESS AUGMENTATION

The goal of the presented multi-robot PSO search algorithm is to find all targets within an indoor environment. Each target is equipped with a cell phone that radiates a Radio Frequency (RF) signal that can be detected by the robots. The cell phone is not observed directly but is detected via the RF signal transmitted by the phone.

Robots are randomly placed within a building to demonstrate the effectiveness of the algorithm. The robots have 802.11 ad hoc wireless communications to other robots within signal strength range. Additionally, each robot is equipped with RF detectors that provide data on the RSS data from a cell phone as well as direction. All targets within the search area carry cellular phones which are to be located by the robot swarm. The distributed PSO algorithm will be tested for both single target and multiple targets. In the multiple target case, each robot will be programmed before deployment to find a specific target. Robots will track their targets based on RSS data received from each target.

The fitness function, f , for a robot relative to a specific target (cell phone) is based on the signal strength that the robot detects from the cell phone, R_x , and the transmission strength of the cell phone T_x . The fitness function approaches an upper boundary threshold, 1, as a robot approaches a target; i.e., the stronger the signal the higher the fitness.

$$f = T_x/R_x \quad (4)$$

Typical cell phones in the U.S. transmit between 824 to 894 MHz at a maximum power of 23dBm. A 2 watt cell phone, based on equation 4, has a transmit power of 33 dBm. Watts are converted to dBm as follows [23]:

$$\text{dBm} = 10 * (\log(1000 * P)) \quad (5)$$

where P = power in watts

The calculations for RSS based on a free space model of RF propagation are [24]-[26]:

$$R_x = T_x - L_p \quad (6)$$

where R_x is received signal strength value. T_x is the power of the cell phone in dB, and L_p is path loss, and

$$L_p = 33\text{dB} + N * \log_{10}(D) + 20 * \log_{10}(f) \quad (7)$$

where f is a frequency in gigahertz, N is a path loss exponent, and D is a distance in meters.

Based on equations (6) and (7), for a typical office environment, path loss and RF power ($N = 3.3$, $T_x = 23\text{dBm}$), and cell phone frequency $f = 0.824$, received signal strength R_x is [22]:

$$R_x = 23 - 33 - 33\log_{10}(D) - 20 * \log_{10}(0.824) \quad (8)$$

Equation (8) shows how the signal strength decreases exponentially with distance from a cell phone regardless of transmission power and antenna gain.

The multi-robot search using PSO with wireless augmentation algorithm is tested in two cases: 1) single target, and 2) multiple targets. In both cases the robots are initially randomly placed in a search area and the target(s) are at unknown locations. The search space in this problem domain has a hard border, such as searching the interior of a building for people with cell phones. A target is considered discovered when a robot comes within 1 meter of the target.

A. Random Robot Placement for Single Target Search

An example of a multi-robot search for a single target is shown in Figure 1. The robots are displayed as color dots and the target is displayed as an X. The mobility of each robot is limited to 45 degree turns. The 6 robots/particles are randomly distributed and are programmed to find a target based on strongest RSS value. The search stops after all of the robots are within a distance threshold of the target or T units of time have expired. T units of time represent the battery life of the robot.

Figure 2 shows the path taken by each robot up until the point where either all robots have reached the target or the time limit T (15 seconds) has expired. The time, in seconds, taken by each robot to reach the target is noted in the diagram. Note that all robots reached the target within the time limit. Time is relevant due to the limited battery life of a robot and the urgency to reach a target; e.g., human, bomb, biological, radiation, or chemical dispersion device before maximum damage has been done to humans and/or the environment.

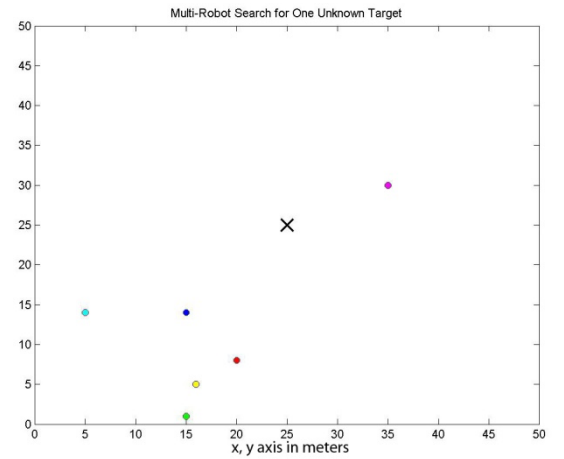


Figure 1. Multi-Robot Search for One Target at Unknown Location

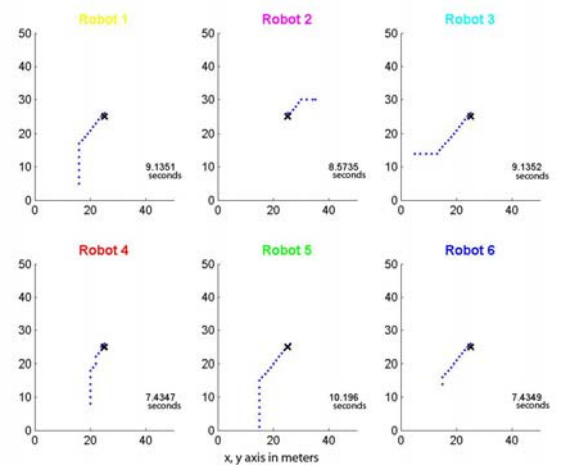


Figure 2. Paths Taken by Each Robot for One Unknown Target

The multi-target search problem assumes 4 targets with cell phones are in the search area at locations unknown to the robots. The theoretic radio map for cell phone targets at specific locations, shown in Figure 3, is based on equation 8. Signal strength decreases exponentially with distance from each cell phone target. Red and dark red indicate the

maximum RSS from each of the targets that can be received by a mobile robot.

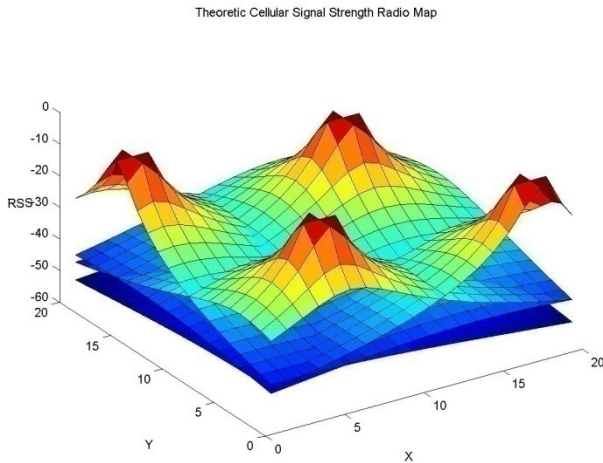


Figure 3. Overlay of Cellular Radio Signals

B. Random Robot Placement for Multiple Targets Search

Figure 4 shows the initial location of randomly distributed multiple robots searching for a known number of multiple targets at unknown locations in the search space. Each robot seeks out a cell phone target based on a cell phone code that is communicated to the robot before deployment. Cell phone codes identify the service provider, the phone, and the owner of the phone. Cell phones transmit their codes to base stations. Multiple robots that seek the same target form mini swarms. Each robot within a mini swarm can only send $lbest$ data to other robots within the same mini swarm. Each robot stays within their local neighborhood (mini swarm) for all iterations of the program.

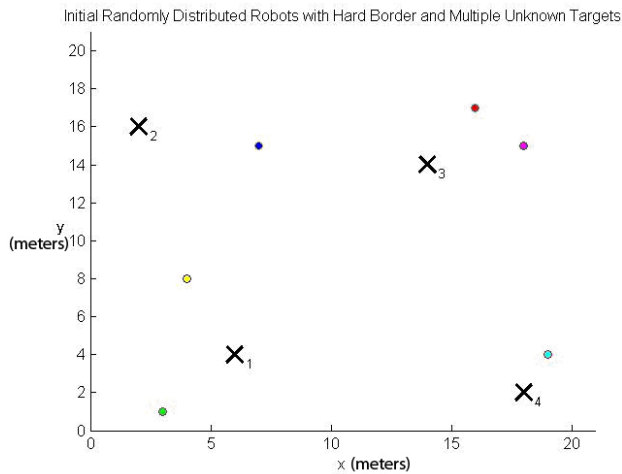


Figure 4. Randomly Distributed Robots with Multiple Unknown Targets

The pseudo code for the PSO multi-robot search using local neighborhoods is as follows:

Step 1: Transfer cell phones codes to robots.

Step 2: Initialize population with random positions and velocities

Step 3: Robots associate with a target based on the maximum signal strength received from each target and cell phone code, thus forming mini swarms

Step 4: Evaluate – compute fitness of each particle in swarm

Step 5: Do for each particle in each mini swarm {

Step 5.1: Find particle best ($pbest$) – compute fitness of particle.

If current $pbest < pbest$

$Pbest = \text{current } pbest$

$Pbest \text{ location} = \text{current location}$

Endif

Step 5.2: Find local best ($lbest$) for the mini swarm

$Lbest \text{ location} = \text{location of min}(\text{all } pbest \text{ in this mini swarm})$

Step 5.3: Update velocity of particle per equation 9

Step 5.4: Update position of particle per equation 3

Step 6: } Repeat steps 5.1 through 5.4 until termination condition is reached.

The velocity of a particle is computed as follows:

$$v_{n+1} = w_i v_n + c1 * r1 * (pbest_n - p_n) + c2 * r2 * (lbest_n - p_n) \quad (9)$$

where $lbest_n$ represents the location of the best fitness within the local neighborhood or mini swarm n .

Figure 5 shows the robot search paths for the robots from Figure 4. Each robot terminates a search after a specific period of time has elapsed or the robot is within one meter of the target. Note the overshooting of the target by robot 4. Overshooting may occur when the velocity of the robot does not slow down sufficiently as the target is approached.

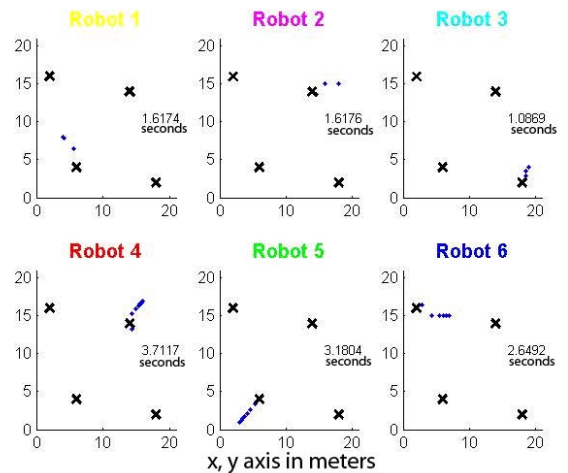


Figure 5. Robot Search Paths

One way to prevent overshooting, where fitness is based on signal strength, is to apply a dynamically weighted value, w_{RSS} , to the velocity as the signal strength detected by the robot increases. w_{RSS} decreases as the distance to the target approaches zero meters. An increasing received signal strength indicates that the robot is approaching the target. The received signal strength of the target decreases

exponentially with distance from the target as shown in Figure 3.

$$w_{RSS} = (RSS_{x,y}/RSS_{min}) \quad (10)$$

where $RSS_{x,y}$ = RSS value at the current location

RSS_{min} = minimum/weakest RSS signal detectable by a robot; the receive threshold on our receivers

The velocity of the robot is now computed as:

$$v_{n+1} = (w_i v_n + c1 * r1 * (pbest_n - p_n) + c2 * r2 * (gbest_n - p_n)) * w_{RSS} \quad (11)$$

Multiplying the velocity of equation (9) by w_{RSS} yields equation (11). When a robot is far from a target and at the RSS_{min} , w_{RSS} will be 1. As the robot approaches the target, w_{RSS} will be a small fraction. Equation 11 has the effect of slowing down the robot as the target is approached. This effectively lengthens the amount of time the robot takes to reach the target but prevents overshooting. Figure 6 shows the robot search paths when equation 11 is used to calculate robot velocity.

Collision avoidance is essential in a multi-robot environment because each robot in the swarm becomes a moving obstacle to the other robots in the swarm [5]. When robot r_i at position p_n calculates the next position to move to, p_{n+1} , r_i must determine if any other robots are on the path from p_n to p_{n+1} . If a potential collision is detected with another robot r_j , the movement algorithm will calculate a new random direction for r_i and check for collisions again until a path with no obstacles is found or r_i decides to stay in its current position until some of the other robots move out of the way. Our robots cannot jump over other robots.

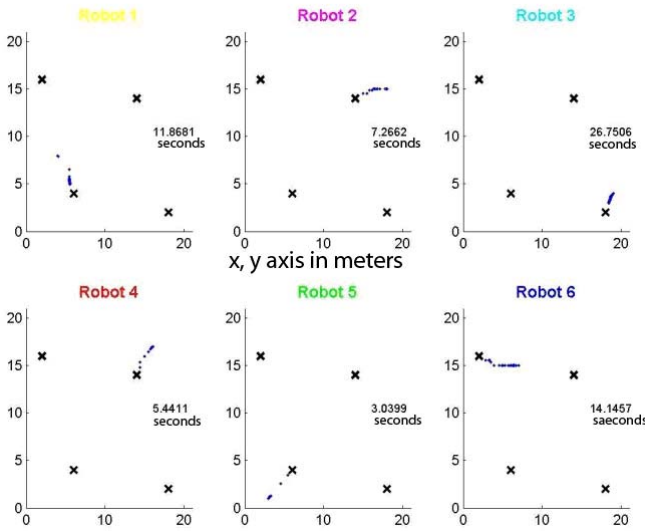


Figure 6. Robot Search Paths using w_{RSS}

The collision avoidance algorithm assumes each robot is of the same size. If a collision is projected to occur for robot r_i after applying equation 11, a new velocity will again be calculated. If collisions are still projected after several tries, robot r_i will remain in the same position until the next time slot or iteration for calculating r_i velocity.

V. MULTI-ROBOT, MULTI-TARGET, SEARCH IN NOISY WIRELESS ENVIRONMENTS

The RSS values detected by each robot sensor in the

previous experiments have been based on a theoretical propagation model that assumes no attenuation or scattering of the radio signal; i.e., no signal degradation or noise. However indoor radio frequency propagation in the real world is affected by objects such as partitions, walls, and floors that scatter the radio signal.

Table 1 shows the results of a series of experiments of introducing signal degradation or noise into the multi-robot multi-target search problem through a single entry way. The first experiment assumes no signal degradation and results in the optimal search path and time to reach each target. Experiments 2 through 4 vary the probability of signal degradation from 30% to 90% with RSS detected by each robot varying by 50%. Experiments 5 through 7 have the same probability variations as experiments 2 through 4 with RSS detected by each robot varying by 25%.

As the probability of signal degradation increases, the path taken by each robot to reach their target becomes more erratic and increases the time to reach target. As the RSS variation from the free space signal propagation value detected by each robot increases the robot path length to reach each target becomes increases as well as the time to reach each target. The time to reach a target was significantly affected by how frequently signal degradation occurred. The RSS variation from the free space propagation value, 25% versus 50%, had little effect on the time to reach the target.

Experiment Number	Probability of Signal Degradation (%)	RSS Variation from Free Space Propagation Value (%)	Average Time to Reach Target (seconds)
1	0	0	43.2
2	30	50	60
3	60	50	77.1
4	90	50	86.3
5	30	25	58.8
6	60	25	76.7
7	90	25	83.9

Table 1. Results of Signal Degradation on Time to Reach Target

VI. CONCLUSIONS

A multi-robot, multi-target search algorithm based on particle swarm optimization is effective in finding targets of interest based on RSS values using a free space model of RF propagation. A dynamically weighted wireless coefficient applied to the standard PSO formula was shown to be effective in preventing overshooting of a target.

Received signal strength (RSS) measurements may vary significantly in an indoor environment where metal or metal containing objects like elevator shafts, support columns, metal framed windows and doors, air ducts, cabinets, computers and computer racks are present. The experiments presented in this paper demonstrate that these variations in RSS measurements may significantly increase the robot search time in finding a target.

REFERENCES

- [1] R. Grabowski, L. Navarro-Serment, C. Paredis, P. Khosla, "Heterogeneous Teams of Modular Robots for Mapping and Exploration", *Autonomous Robots*, Vol. 8, No. 3, pp. 293-308, June 2000.
- [2] R. Grabowski, L. Navarro-Serment, P. Khosla, "An Army of Small Robots", *Scientific American Reports*, Vol. 18 No. 1, pp. 34-39, May 2008.
- [3] G. Bekey, "Autonomous Robots", MIT Press, pp. 391-439, 2005.
- [4] J. Pugh, A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization",
- [5] L. Smith, G. Venayagamoorthy, P. Holloway, "Obstacle Avoidance in Collective Robotic Search Using Particle Swarm Optimization",
- [6] J. Hereford, M. Siebold, "Multi-Robot Search Using a Physically-Embedded Particle Swarm Optimization",
- [7] R. Morlok, M. Gini, "Dispersing Robots in an Unknown Environment",
- [8] T. Hsiang, E. Arkin, M. Bender, S. Fekete, J. Mitchell, "Online Dispersion Algorithms for Swarms of Robots", *ACM SoCG'03*, June 8-10, 2003, San Diego, California, USA
- [9] T. Hsiang, E. Arkin, M. Bender, S. Fekete, J. Mitchell, "Algorithms for Rapidly Dispersing Robot Swarms in Unknown Environments", 19th Annual ACM Symposium on Computational Geometry (SoCG), June 8-10, 2003, San Diego, USA.
- [10] A. Birk, S. Carpin, "Merging occupancy grid maps from multiple robots", *Proceedings of the IEEE*, 94 (7), Special Issue on Multirobot Systems, pp.1384-1397, IEEE Press, 2006
- [11] J. Pugh, A. Martinoli, "Distributed Adaptation in Multi-Robot Search using Particle Swarm Optimization",
- [12] Y. Meng, K. Cao, "Multi-Robot Searching for a Target using Game Theory", *Robotics and Applications*, 8/14/2006 - 8/16/2006, Honolulu, Hawaii
- [13] B. Yamauchi, "Frontier-Based Exploration Using Multiple Robots", *Proceedings of the Second International Conference on Autonomous Agents (Agents '98)*, Minneapolis, MN, May 1998.
- [14] M. Rooker, A. Birk, "Multi Robot Exploration Under the Constraints of Wireless Networking", *Control Engineering Practise*, 15 (4), pp. 435-445, Elsevier Science, 2007
- [15] M. Taşgetiren, Y. Liang, "A Binary Particle Swarm Optimization Algorithm for Lot Sizing Problem", *Journal of Economic and Social Research* 5 (2), 1-20
- [16] W. Jatmiko, K. Sekiyama, T. Fukuda, "A PSO Based Mobile Sensor Network for Odor Source Localization in Dynamic Environment: Theory, Simulation and Measurement", 2006 Congress on Evolutionary Computation, Vancouver, BC, pp. 3781 – 3788, July 2006.
- [17] S. Bergbreiter, K. Pister, "CotsBots: An Off-the-Shelf Platform for Distributed Robotics", *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems Las Vegas, Nevada, October 2003*.
- [18] J. Kennedy, R. Eberhard, "Particle Swarm Optimization," *Proc. of IEEE Int'l Conf. on Neural Networks*, Piscataway, NJ, USA, pp. 1942-1948, 1995.
- [19] J. Pugh, L. Segapelli, A. Martinoli, "Applying Aspects of Multi-Robot Search to Particle Swarm Optimization",
- [20] S. Doctor, G. Venayagamoorthy, V. Gudise, "Optimal PSO for Collective Robotic Search Applications", *IEEE Congress on Evolutionary Computation*, Portland, OR, pp. 1390 – 1395, June 2004.
- [21] S. Doctor, G. Venayagamoorthy, "Unmanned Vehicle Navigation Using Swarm Intelligence", *Intelligent Sensing and Information Processing, ICISIP 2005*.
- [22] *Wireless Mesh Network for Building Automation, Industrial Wireless Book Issue 10:2*, <http://wireless.industrial-networking.com/articles/articledisplay.asp?id=1264>
- [23] "Broadband DSL Reports", <http://www.dslreports.com/faq/2743>
- [24] M. Gast, "802.11 Wireless Networks", O'Reilly Media Inc., 2005, pp. 225-240.
- [25] "Indoor Path Loss", MaxStream, September 2003, <http://digilander.libero.it/windflow/eng/Usage/RFIDSensor/Indoor%20Path%20Loss.pdf>
- [26] W. Debus, "RF Path Loss and Transmission Distance Calculations", Axonn, Technical Memorandum, September 2006, <http://www.axonn.com/pdf/path-loss-calculations.pdf>