Cooperative Area Extension of PSO

Transfer Learning vs. Uncertainty in a simulated Swarm Robotics

Adham Atyabi1 and David M.W. Powers1 2

1School of Computer Science Engineering, & Mathematics (CSEM), Flinders University, Australia
2Beijing Municipal Lab for Multimedia & Intelligent Software, Beijing University of Technology, China

{Adham.Atyabi,David.Powers}@flinders.edu.au

Keywords: Swarm Robotics, Particle Swarm Optimization, Cooperative Learning, Transfer Learning, Knowledge Transfer

Abstract: Navigation in dynamic and uncertain environments in the absence of reliable environment map is challenging. In this study, we investigate the effectiveness of two variations of Particle Swarm Optimization (PSO) called Area Extended PSO (AEPSO) and Cooperative AEPSO (CAEPSO) in noisy environments in which the noise does not represent random noise originated from a single type of source but the combination of noises originating from different sources located in nearby or faraway positions. Knowledge Transfer and Transfer Learning that represent the use of the expertise and knowledge gained from previous experiments can improve the robots decision making and reduce the number of wrong decisions in such uncertain environments. This study investigates the impact of transfer learning on robots’ search in such hostile environment. The results highlight the feasibility of CAEPSO to be used as the movement controller and decision maker of a swarm of robots in the simulated uncertain environment when gained expertise from past trainings is transferred to the robots in the testing phase.

1 INTRODUCTION

Navigation is the art of steering a course through a medium. Localization matches an actual position in the real world to a location inside a map. Planning is finding a short, collision-free path from the starting position towards the predefined ending location. As such, navigational techniques are useful and effective when the map on similar information is reliable. Nevertheless, in most real world application domains, the environment is dynamic, time-dependent, and uncertain. Such environmental conditions pose challenges to localization and map reliability. Under such circumstances, behavior-based approaches are suitable to address real world applications when engaged with navigation problems.

The Swarm Intelligence (SI) term, introduced by Beni, Hackwood, and Wang in 1989, used for systems in which, unsophisticated agents with collective behaviors have the capability of emerging to a global pattern by interacting locally with their environment. SI systems have the capability to solve the collective problems without centralized control. Particle Swarm Optimization (PSO), introduced by Kennedy and Eberhart in (Kennedy and Eberhart, 1995), has been inspired from animals’ social behaviors which are illustrated by their social acts resulting in population survival. PSO is a self-adaptive population-based method in which, behaviors of the swarm are iteratively generated from the combination of social and cognitive behaviors of the swarm. A swarm can be imagined as consisting of members called particles (also known as agents, observers or robots as in other studies (Bill et al., 2004; Nomura, 2007; Pugh and Martinoli, 2006; Ribeiro and Schlansker, 2005; Sousa et al., 2004). Particles cooperate with each other to achieve desired behaviors or goals. Particles’ acts are governed based on simple local rules and interactions with the entire swarm. As an example, movement of a bird in a flock is based on adjusting movements with its flock mates (near by neighbors in the flock) (Sousa et al., 2004). Birds in a flock stay close to their neighbors and avoid collisions with each other. They do not take commands from any leader bird (there are no leader birds). This kind of social behavior (Swarm behavior) helps birds to achieve tasks such as protection from predators and searching for food (Grosan et al., 2006; Sousa et al., 2003; Yang and Gu, 2004).

Although PSO has proved to be efficient in solving problems in various domains, it comes with consid-
erable shortcomings. The shortcomings include premature convergence and difficulties with dynamic and real-world optimization. Enhanced versions of PSO called Area Extended PSO (AEPSO) and Cooperative AEPSO (CAEPSO) showed potential in dynamic and uncertain simulated environments (Atyabi et al., 2010). In the study, two variations of uncertainty (i.e. random noise and relational noise) are employed to assess the feasibility of the suggested approaches. The dynamically changing nature of the simulated environment in (Atyabi et al., 2010) make it more difficult for the simulated swarm of simple robots to compensate the noise. CAEPSO tackled this problem through incorporating previously learned knowledge to its decision making process. This study investigates the impact of knowledge transfer on the achieved performances from several experiments.

The outline of the study is as follows: Section 2 introduces the PSO and AEPSO algorithms. The details about the simulated uncertainty are discussed in Section 3 and CAEPSO is presented in section 4. The experimental setup and the achieved results are presented in Section 5. Section 6 presents the conclusion.

2 PARTICLE SWARM OPTIMIZATION (PSO)

2.1 Basic PSO

Basic PSO is an evolutionary approach introduced by Kennedy and Eberhart in 1995 and it is inspired from animal social behaviors. PSO generates a population of possible solutions called particles (denoted by X) and evolve them toward optimum by iteratively readjusting particles’ velocity (denoted by V). PSO takes advantage from cooperation among particles resulted from sharing their best findings with each other for achieving the optimum. The cooperation among particles in the swarm is facilitated through the use of social and cognitive components in the basic PSO velocity equation as shown in equation 1.

\[
\begin{align*}
V_{ij}(t) &= wV_{ij}(t-1) + C_{ij} + S_{ij} \\
C_{ij} &= c_1 r_{1ij} \times (p_{ij}(t-1) - x_{ij}(t-1)) \\
S_{ij} &= c_2 r_{2ij} \times (g_{ij}(t-1) - x_{ij}(t-1))
\end{align*}
\]

(1)

In the equation, C and S represent social and cognitive components. In the equation, i and j represent particle’s index and particle’s dimension in the search space respectively. t is the iteration number. \( r_1 \) and \( r_2 \) are random values between 0 and 1, w is the inertia weight. Linearly Decreasing the inertia weight (LDIW) formulated in equation 2 is one of the typical approaches used for adjusting w in basic PSO in which \( w_1 \) and \( w_2 \) the initial and final inertia weight, and \( \text{maxiter} \) is the maximum number of iterations.

\[
w = (w_1 - w_2) \times \frac{(\text{maxiter} - t)}{\text{maxiter}} + w_2
\]

(2)

In equation 1, \( c_1 \) and \( c_2 \) are the acceleration coefficients used to control the impact of social and cognitive components on the readjustment of the particles’ velocity. The cognitive component attracts the particles in the swarm toward their best personal findings (p). The social component attracts the particles in the swarm toward the global best findings (g). In PSO, personal and global best are updated using following equations:

\[
P_i(t) = \begin{cases} 
P_i(t-1) & \text{if } f(x_i(t)) \geq f(P_i(t-1)) \\
x_i(t) & \text{otherwise} \end{cases}
\]

(3)

\[
g(t) = \text{argmin}\{f(P_1(t)), f(P_2(t)), ..., f(P_P(t))\}
\]

(4)

f represent the fitness (evaluator) function. The new position of each particle can be computed by following equation:

\[
x_{i,j}(t) = x_{i,j}(t-1) + V_{i,j}(t)
\]

(5)

A detail discussion about PSO, its advantages and shortcomings is presented in (Atyabi and Samadzadegan, 2011).

2.2 Area Extended PSO

This new enhanced version of PSO is first introduced with the aim of solving basic PSO problems in robotic domains. The idea is based on using advanced versions of neighborhood topology and communication methodology with the aim of improving basic PSO performance in two dimensional multi-robot learning task in static, dynamic and noisy environments. In AEPSO, we solved fundamental problems\(^1\) of basic PSO by adding some heuristics to it. Later on, the feasibility of the proposed modifications are examined in a simulated environment within several survivor rescuing scenarios. These heuristics are as follows:

2.2.1 To handle dynamic velocity adjustment

AEPSO takes advantage from a new velocity adjustment heuristic which tackles the premature convergence using equation 6.

\[^1\]Fundamental problems of basic PSO are known as i) Lack of Dynamic Velocity Adjustment, ii) Premature Convergence, iii) Controlling Parameters and iv) Difficulties in Dynamic and Time Dependent environments (Peter, 1998; Mauris, 2002; Jakob and Jacques, 2002)
2.2.2 To handle direction and fitness criteria:

AEPSO takes advantage from two heuristics known as Credit Assignment and Environment Reduction to addresses the clue de sacs problem (Suranga, 2006; Majid et al., 2001).

**Environment Reduction Heuristic:** The environment reduction heuristic is inspired from the work done by (Fujii et al., 1998; Park et al., 2001; Yang and Gu, 2004) in which it is stated that it is possible to separate a large learning space to several small learning spaces with the aim of easing the exploration. In this heuristic, the large environment is divided to some sub-virtual fixed areas with various credits. As in our simulations the environment is 500×500 pixels; each area is defined as a 20×20 pixels with square shapes. Therefore, the environment is divided to a matrix of 25×25 areas (625 areas overall). Each area contains 400 pixels with each pixel representing a possible location for obstacles, survivors, or robots. A credit associated to each area indicate the proportion of robots, survivors and obstacles positioned in that area. Only the likelihood information about the areas is provided to robots. An overall elimination time (iterations) for each area is also included in the areas credit to help robots to prioritize the observations of areas with lower elimination times. This elimination time indicate the overall amount of time left before all survivors in an area get eliminated. In this environment, the exploration appears when a robot leaves its current area for another one and the exploitation appears when a robot searches for survivors inside an area. In our scenarios, robots only exploit areas that contain survivors (areas with positive credits). This results in a better balance between the exploration and the exploitation behaviors since robots only exploit areas with positive credits. In order to increase robots’ awareness, the likelihood information (areas’ credits) of their neighboring areas are provided to them. These neighboring areas are divided to two layers of near and far neighboring areas as shown in figure 1. In the simulation, robots set their direction according to the direction of the destination area and use maximum velocity whenever they want to leave an area for a new one.

**Credit Assignment Heuristic:** (Jim and Martinioli, 2007) argued that in some cases, as in macroscopic modeling of PSO, in a robotic problem, mathematical functions (benchmark functions) might not be appropriate as fitness evaluators. It is due to the fact that in such modeling, values of particles in the swarm refers to actual locations of agents in the environment. In AEPSO, a Punishment/Reward mechanism (inspired from reinforcement learning) is used instead. In here, robots would be rewarded positive credit whenever they find a survivor or whenever they locate themselves inside an area with positive credit. On other hand, robots would be punished by receiving negative credits if they do not achieve any reward after certain iterations or if they collide with obstacles.

2.2.3 To handle cooperation between robots:

**Communication Heuristic:** AEPSO takes advantage from a heuristic that facilitates the communication across robots based on a predefined ranges. Robots are allowed to communicate and share their knowledge with those that are in their communication range. This heuristic facilitates dynamic neighborhood topology and helps to create sub-swarms (Jakob and Jacques, 2002; Brits et al., 2002; James and Rui, 2002).

**Help Request Heuristic:** This heuristic provides cooperation between different sub-Swarms by allowing robots to request assistant and seek cooperation from other robots that are in their communication range whenever they require it. Robots that receive the request can either acknowledge the request or pass it through to other robots in their communicating range. This provides a chain of communication between robots that are far away from each other.

2.2.4 To handle the search diversity:

**Boundary Condition Heuristic:** AEPSO takes advantage from a heuristic known as Boundary Condition heuristic which solve the lack of diversity in basic PSO (Jakob and Jacques, 2002; Asanga et al., 2004; Bill et al., 2004). The heuristic force robots…

\[
\begin{align*}
\bar{V}(t+1) &= \text{fittest} \\
\end{align*}
\]

\[
\begin{align*}
\text{c}1 &= \text{rand}() \times (p_{i(t)} - x(t)) \\
\text{c}2 &= \text{rand}() \times (g(t) - x(t)) \\
\end{align*}
\]

\[
\begin{array}{ccccccc}
16 & 1 & 2 & 3 & 4 & 5 \\
15 & 8 & \text{current} & 4 & 7 \\
14 & 7 & 6 & 5 & 8 \\
13 & 12 & 11 & 10 & 9 \\
\end{array}
\]

Figure 1: The first and the second neighboring areas of the current area.
that get too close to the boundary of the environment
to relocate themselves to somewhere in the middle of
the environment by selecting an ad hoc direction and
forcing them to move toward that direction for certain
number of iterations.

2.3 AEPSO Vs. PSO in a simulated
survivor rescuing scenario

In our simulation, we used variations of PSO as deci-
sion makers and movement controllers of autonomous
robots. Figure 2 (a and b) shows the results of basic-
PSO in terms of trajectory traces in such simulation.
The figures shows two different experiments with dif-
ferent initializations based on a survivor rescuing sce-
nario in which team of 5 homogeneous robots are
meant to find 15 survivors before they get eliminated.
The environment is 500 × 500 meter and the sim-
ulated robots have detect objects (survivors or static
obstacles) within a circle of 5 meter around them. In
the figure, the blue and yellow circles are representing
survivals and obstacles respectively. Furthermore,
lines with different colors represents different robots
working as a team in the environment.

As the figures shows, in basic PSO, the results
are highly dependents to the initial locations of the
instructor elements of the environment (robots, ob-
stacles, survivals). Furthermore, due to the lack of
balance between exploration and exploitation, robots
were not able to cover a high percentage of the envi-
ronment during their search while AEPSO was able
to overcome the problem. The achieved mapping per-
formance by AEPSO in addition to its ability to over-
come random noise as it is demonstrated in (Atyabi
et al., 2010) encouraged us to further evaluate the al-
gorithm in simulated environments that are affected
by highly complicated and more realistic type of
noise.

3 EVOLVING UNCERTAINTY
(ILLUSION)

The illusion noise presented in (Atyabi et al.,
2010) is a combinatorial type of noise in which the
noise value applied to the credit of each area in the
environment is influenced from the neighboring (first
layer of area neighborhood see figure 1) and far away
areas (packs of neighborhood) and based on the ongo-
ing activities in the environment the value of the noise
in each area is changed over time (iteratively chang-
ing noise). In a simulated environment of 500 × 500
pixels divided to 625 different areas of 20 × 20 pixels

Figure 2: The trajectory traces of robots controlled by Basic
PSO (two different executions) and AEPSO. different colors
are used to represent different robots’ trajectories. Blue and
yellow dots represent survivors and obstacles respectively.
resulting to a matrix of $25 \times 25$ cells, the application of the illusion effect results in a matrix representation of 25 equations needs to be predicted in each iteration in order to reveal true credit of each area for that iteration. Equation 7 shows the illusion credit of area $i$ (this equation would be computed for each area in each of the iterations to provide the noisy credit of that area).

$$C_i(t+1) = C_i(t) + N(C_i(t)) + S(C_i(t))$$

$$N(C_i(t)) = \sum_{j=1}^{8}(a \times \text{actual\_credit} (\text{area}_{i,j}))$$

$$S(C_i(t)) = \sum_{j=1}^{8}(b \times \text{neighboring\_pack}_{(i,j)})/8$$  \(7\)

$C_i$ represents the corrupted value of an area affected by illusion noise in iteration $(t+1)$. $a$ and $b$ are constant parameters ($a = b = 0.125$) used to control the impact of $j^{th}$ neighboring area and neighboring pack of area $(i)$ ($C_i$). $N(C_i(t))$ and $S(C_i(t))$ represent the effect of other neighboring areas and neighboring packs on current area $(C_i)$ respectively. actual\_credit(area$_{i,j}$) area is the uncorrupted credit of $j^{th}$ neighboring area of area$_{i}$. neighboring\_pack$_{(i,j)}$ indicates the impact of neighboring packs of the pack which contains area$_{(i)}$. This impact is assumed as the average credit of areas who are the members of that pack.

Figure 3 illustrates an snapshot of the simulated environment which also demonstrate neighboring packs of an area. In the figure, the neighboring packs are depicted by green color rectangles while the neighboring areas of the current area are illustrated by a red rectangle. The filled rectangle with yellow color represent the imaginary current area. Figure 4 demonstrate the impact of the illusion effect on two simple pictures aiming to help with the understanding of the resulting complexity.

4 Cooperative Area Extension of PSO (CAEPso)

Considering the complication that can be arised as the result of application of the illusion noise, the best possible way to solve the resulting huge puzzle is to identify the areas and neighboring packs that have the highest influence on others first. Identifying and mapping such areas results in eliminating their effects on other areas (by reducing their credits to zero). The decisions that agents take about the credit of each area are based on the elements such as Past knowledge, Current knowledge and perception, additional heuristics (Speculation mechanism).

4.1 The additional Heuristics of CAEPSO

In CAEPSO, two additional heuristics are suggested to tackle the complexities raised by the application of the illusion noise.
the illusion noise. These heuristics are as follows:

- **Leave Force**: The heuristic decreases 10% of an area’s credit in a robot’s mask whenever the robot enters the area or spent certain number of iterations exploiting that area (i.e., the area’s flag would be changed to self speculation and the credit would be reduced by 10% off). The heuristic guarantee that robots do not spend a long time in an area and do not get stuck in areas.

- **Speculation mechanism**: The heuristic helps to provide a high level of noise resistance. Speculation mechanism is based on using a small memory as a mask of the entire environment (i.e., a matrix of $25 \times 25$ cells, each cell representing an area in the simulated environment). In the start of a simulation, the value of each cell in the matrix represent the corrupted credit of the associated area in the environment (corrupted with the illusion noise) and later, robots update their mask of the environment based on their own observations and knowledge gained from other robots through knowledge sharing and communication. The value of robots’ masks would be changed by the following factors:
  - robots’ and their neighbors’ self-observations which reduce the referring cells’ value to zero. This happens whenever a robot fully map an area and locate and rescue all survivors within that area. The robot reduces the value of the cell to zero.
  - robots’ and neighbors’ speculation which decreases the referring cells’ value of their mask to the measured value.

As robots share their masks with each whenever they are in each other communication range, they assess each others expertness to clarify the reliability of the knowledge they are receiving. That is, less expert robots only share their own or the others self-observations (the information that they are sure of); more expert robots also share their speculations about the areas’ true credit. The robots degree of expertness are assessed based on factors such as the number of cells in their masks marked as self-observations and proportion of their rewards and punishments.

Following steps are taken when robots are within each other communication range (Knowledge sharing mechanism):

- **Start of knowledge sharing**
- **Evaluate the expertness level of both robots (a and b)**
- **The non-expert robot (a) only shares cells of its environment mask which are flagged as self and neighbor observation.**
- **The Expert robot (b) shares cells of its environment mask that are flagged as self-observation, self-speculation, neighbor-observation, and neighbor-speculation.**

- **End of knowledge sharing**

As mentioned earlier, in an environment affected by illusion, each agent should choose an area for observation (exploitation) and their decisions can have significant effects on their own and group’s performance (e.g., if they choose the best area (the area with the highest effect on others), they can help to reduce a high percentage of noise in the other areas and therefore, they can help to increase the group performance).

Pseudo-code of taken steps when robots are controlled by CAEPSO is presented in algorithm 1.

**Algorithm 1** Pseudo-code for controlling a robot with CAEPSO.

| Step 1: Begin |
| Step 2: Initialization the robot is randomly located in the environment. |
| Step 3: use CAEPSO algorithm to update the robot’s location. |
| Step 4: Checking the surrounding areas’ credits, if it is changed due to robot’s action then Speculation Mechanism is used to update the credits of surrounding areas in robot’s memory. |
| Step 5: If all survivors are not located yet then go back to step 3. |
| Step 6: If the maximum number of iterations is not reached yet then go back to step 3. |
| Step 7: End |

### 4.2 Past-Knowledge

The major differences between AEPSO and CAEPSO are in the use of past knowledge provided by AEPSO during the training phase and two additional heuristics (Speculation and Leave-Force heuristics) in CAEPSO. That is, AEPSO can only use new velocity adjustment, Environment reduction, credit assignment, communication policy, help request signal, and boundary condition while CAEPSO use them and also additional heuristics known as leave force and speculation mechanism. A detailed discussion and description of AEPSO and CAEPSO can be found in (Atyabi, 2009; Atyabi et al., 2010).

Several studies in cooperative learning domain discussed the incorporation of knowledge gained...
from multiple training sessions in order to increase the overall performance of swarm of robots (Majid et al., 2001; Tangamchit et al., 2003; Yamaguchi et al., 1997). The training phase in robotic swarm can be designed to be either with individual or team of robots. Using a single robot for training and knowledge gathering is problematic given that the resulting information do not compensate the changes in the environment caused by other members of the swarm in the testing phase. On other hand, when a swarm of robots are used in the training phase, assessing the impact of the made decisions by individuals on the overall achieved performance is challenging if not impossible. In this study, a swarm of robots controlled by PSO are used in the training phase with an environment that is under the influence of illusion noise. Twenty runs of the experiment with random initial locations for robots, survivors and obstacles are executed and the gained knowledge is passed to the swarm of robots controlled by PSO in the testing phase (different initializations is used in the testing phase). The difference between the initial locations used in the training and testing phases helps to better reflect real world situations in which the world is continuously changing. It worths highlighting that the most important decision that robots make in such noisy environment is which area to exploit first. If areas with highest noise impact on others (see illusion effect description) are chosen first, high percentages of the noise would be reduced from the environment. Considering the afore mentioned factor the past knowledge gained from the training phase is designed to reflect the potential of the made decisions in terms of the chosen directions (neighboring areas) with the individuals and the team of robots. Table 1 represent a template used for passing the gained overall knowledge from past trainings.

Table 1: The mask used to represent robots overall training phase knowledge (past knowledge). The figure is adapted from (Atyabi et al., 2010).

<table>
<thead>
<tr>
<th>a / b / c</th>
<th>a / b / c</th>
<th>a / b / c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>a / b / c</td>
<td>Current</td>
<td>a / b / c</td>
</tr>
<tr>
<td>8</td>
<td>Area</td>
<td>4</td>
</tr>
<tr>
<td>a / b / c</td>
<td>a / b / c</td>
<td>a / b / c</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

In the table, areas are denoted by numbers one to eight. In each of the areas, a represents the amount of times that the area was the best area to be chosen (area with the most effect on others in its pack) (i.e., actual best). b represents the number of times that the area was chosen as a positive credit area (potential direction), and c represents the number of times that the chosen area was in fact the best area to be chosen. Here, past knowledge refers to the overall knowledge gathered during the training phases from various trials/executions. In the testing phase, agents may be experiencing the same or new random initializations. Such a knowledge help agents to have overall information about their previous training and the quality of their previous decisions.

The pseudo-code of CAEPSO is presented in algorithm 2.

**Algorithm 2 CAEPSO Pseudocode**

**Initialization:** Randomly initialize the robots’ locations in the simulated environment.

Robots’ masks’ values= areas’ credits affected by illusion noise while (maximum number of iterations is reached or all survivors are found) do

for (each robot in the swarm) do

if current area’s credit = 0 chose a new area for observation using environment reduction heuristic, robot’s mask, and past knowledge else if (robot’s behavior = exploitation & robot’s performance is low) then robot’s behavior = exploration using leave force heuristic end else if (suspend factor or binary conditions are true) then use boundary condition, and credit assignment heuristics end else

Update velocity using equation 6 and 2
Update robot’s location using equation 5
Evaluate robot’s new location using credit assignment heuristic
Update robot’s mask using speculation mechanism end

then

end

end Updating the personal_best using equation 3
Communicate with robots located in the communication range
Updating the global_best using equation 4
Update robot’s mask using speculation mechanism

end
5 EXPERIMENTS AND RESULTS

A simulated environment with 500×500 pixels dimension in which each pixel represents 1 meter is used. The environment is polluted with illusion effect. Each simulated robot can only observe objects within 5 pixels of its surrounding environment. A team of 5 robots are used for mapping the environment and locating the survivors. 15 survivors are randomly located in the environment in addition to 50 obstacles. Survivors and obstacles are static. The maximum number of iterations is set to 20,000 while the elimination time for each survivor is set as a random value in the range of 5,000 to 20,000. The robots task is to locate as many survivors possible before they get eliminated and the maximum iteration number arrives. The experiments are designed in two phases. In the first phase, swarm of 5 robots controlled with AEPSO are randomly located in the environment with the task of finding the survivors.

The decisions made by robots during 20 random runs of the experiment (each lasting up to 20,000 iterations) are evaluated and aggregated in mask and past to robots in the second phase (see an example of the mask in table 1). The experimental design and configuration in the second phase is similar to the first phase with the exception of using the past knowledge and CAEPSO as the controller of the robots. In all experiments LDIW is used with $w_1 = 0.2$ and $w_2 = 1$ and Fix Acceleration Coefficients of $c_1 = 0.5$ and $c_2 = 2.5$ are employed. Other parameter settings and swarm configurations are considered and discussed in (Atyabi, 2009) among which the chosen setting showed consistently better overall performance. Four sets of scenarios are designed in two phases of training and testing to address homogeneity and heterogeneity. The scenarios also investigate the potential of the transferred knowledge from the training phase when similar and new initializations are used in the testing phase. A detailed description of the experiments and the achieved results can be find in (Atyabi et al., 2010). In here, we are only interested in the impact of knowledge transfer on overall achievements across all scenarios discussed in (Atyabi, 2009; Atyabi et al., 2010).

The results in (Atyabi et al., 2010) showed transcendent improvement in terms of learned knowledge and agents’ movement between training and testing phases. The results indicated that CAEPSO rescued 99% and 95% of the survivors during the testing phase with homogeneous and heterogeneous scenarios while in similar scenarios, robots controlled by AEPSO in the testing phase were only able to rescue 50% and 45% of the survivors. The type of past-knowledge provided to robots during the testing phase helped them to locate survivors in the testing phase regardless the differences between training and testing initialization. That suggest that CAEPSO is reliable in environments that have no direct past knowledge about it. The differences between the number of the eliminated survivors during the training and testing phases indicate CAEPSO’s capability on overcoming the illusion effect and locating survivors in expected times.

The achieved trajectory traces from a randomly selected experiment representing the use of AEPSO as the controller of 5 robots during the first phase is illustrated in figure 5. The comparison between the results demonstrated in figures 5 and 2 indicate inability of AEPSO to overcome the illusion effect evidenced by low portion of mapped environment. The aggregated results of the made choices by robots controlled by AEPSO during the first phase depicted in table 1 further indicate lost of overall performance due to inaccurate and inefficient choices made by robots in terms of which neighboring area to observe and map first. The wrong choices made by robots result in their inability to remove or reduce the effect of illusion noise from the environment.

The achieved trajectory traces from a randomly selected experiment representing the use of CAEPSO as the controller of robots during the second phase is illustrated in figure 6. The figure also illustrate the
impact of past knowledge on the made decisions by robots. Figure 6 (a) represents trajectory traces of 5 robots (presented by different colors) when no past knowledge is passed to them. The figure indicate occasional stagnation of robots in some areas and their inability to map a high percentage of the environment. In contrast, as evidenced in sub figures b and c, when past knowledge is presented to robots controlled by CAEPSO considerable percentage of the environment is mapped by each robot. The impact of the past knowledge on the made decision by robots is also evident in the different in terms of quality of the made decision presented in tables 2 and 3. The difference between the overall results presented as a mask of the made decisions about which neighboring area to be mapped and observed first in the first and second phase indicate considerable improvement in the made decisions in the second phase.

The combination of the presented results in tables 2 and 3 and figure 6 suggest the importance of knowledge transfer in environments polluted by combination of noises originating from different sources as in illusion noise.

6 CONCLUSIONS

This study discussed the impact of past knowledge on decisions made by a group of robots controlled by two variations of PSO called Area Extended PSO (AEPSO) and Cooperative AEPSO (CAEPSO). In order to evaluate such an impact a type of noise called Illusion effect is simulated. The illusion effect represent an iteratively changing noise that is the outcome of some combinations of noises originating from different sources as in illusion noise.

Figure 6: The testing phase trajectory traces of robots when controlled by CAEPSO. a) represent the trajectory trace of 5 homogeneous robots when no past knowledge is presented. Trajectories with different colors represent different robots. b) and c) represent trajectory traces of two members of a homogeneous robotic swarm when past knowledge is also used. The survivor rescuing tasks are time dependent and the environment is corrupted by illusion. Sub-figures b and c are adapted from (Atyabi et al., 2010).
ferent sources located somewhere near or far away. The results of simulated experiments indicates the important role of past knowledge in compensating the illusion noise and making correct decisions by the simulated robots.

REFERENCES


