A Local Charged Particle Swarm Optimization to track an underwater mobile source

Charles Coquet, Clement Aubry, Andreas Arnold, Pierre-Jean Bouvet

A Local Charged Particle Swarm Optimization to track an underwater mobile source

1st Charles Coquet
WAVES laboratory
Thales Group
Brest, France
charles.coquet@fr.thalesgroup.com

2nd Clement Aubry
L@bisen
ISEN
Brest, France
clement.aubry@isen-ouest.yncrea.fr

3rd Andreas Arnold
WAVES laboratory
Thales Group
Brest, France
andreas.arnold@fr.thalesgroup.com

4th Pierre-Jean Bouvet
L@bisen
ISEN
Brest, France
pierre-jean.bouvet@isen-ouest.yncrea.fr

***Abstract*—In this paper, a possible solution to track a mobile underwater source in a closed environment with N Autonomous Underwater Vehicles (AUV) in a swarm formation is adressed. The source tracking algorithm is defined as successful when the range between the source and the swarm is sufficiently low during a given duration, short enough to perform a specified action (for example a source localization). A source is defined as an entity that releases a scalar information affected by transport and diffusion in the environment. We use a generic time-varying information $f(\mathbf{p}_i(t))$, where $\mathbf{p}_i$ at time t is the m-dimensional position of a tracker i and function $f(.)$ is a function that represents sensor information. In this paper, we propose an innovative tracking method inspired by the Particle Swarm Optimization (PSO) algorithm that we call the Local Charged Particle Swarm Optimization (LCPSO). The proposed algorithm is adapted to range-dependant communication that characterizes the underwater context and includes flocking parameters. Comparison of the LCPSO against state of the art methods demonstrate the interest of our approach in an underwater scenario.**

***Index Terms*—PSO, Swarm, Robotics**

I. Introduction

We consider the problem of where a static or moving target is to be localized by a collection of agents using arbitrary sensors and interactions with the environment. More specifically we consider underwater applications, which is a much more constrained environment than above the surface. For instance, we consider the case where the agents are a swarm of Autonomous Underwater Vehicles (AUV). Sensors could be pressure sensors, including acoustic sensors [1], [3], [16], [18]; magnetic sensors [5]; detectors of various chemicals [12]. The applications include securing a specified area to detect intruders; to prevent illegal oil spilling by tankers; localizing a pipeline leak [2], [22], or various other applications yet to be imagined. The global theory of Odor Source Localization (OSL) works in atmospheric and underwater environment. The source diffuses an odor that is affected by transport and diffusion, but the source does not move. A review of six different OSL algorithms involving laboratory experiments was made in 2010 [12]. The subject is divided into three parts:

- Plume finding: the robot searches a first odor acquisition,
- Plume traversal: the robot is in the plume and has a strategy not to lose it,
- Source declaration: the robot has enough good measurements to localize the source or to determine its nature.

A conclusion of [12] is the efficiency of multi-agent strategies to track a source. Social aspects of search as information exchanges and repartition of robots in space allows to answer more easily to perturbations, like noise and a dynamic environment.

Typical applications are Chemical Plume Tracing (CPT) [19], [22], source pollution tracking , as well in an atmospheric [8], [15] or underwater context [2], [22], and rescue missions [11]. However communications are much more constrained underwater than in the air, thus atmospheric methods can not be applied directly to the underwater world.

Our problem statement consists of a mobile source tracking in a closed underwater area by a swarm of AUVs. The source is represented by the state vector below:

$$\mathbf{x}_s(t) = (\mathbf{p}_s(t), v_s(t), \theta_s(t))^T \tag{1}$$

where $\mathbf{p}_s(t) = (x_s(t), y_s(t))^T$ is the two-dimensional source position, $v_s(t)$ is the source speed norm and $\theta_s(t)$ is the source heading. There is not inter-dimensional terms in the algorithms presented in this paper [23], allowing us an analysis in dimension 2 true for all dimensions. The source releases an information that can be measured at position $\mathbf{p}$ and time t as $f(\mathbf{p}, t)$ where f is a function with scalar output in $\mathbb{R}$ and $\mathbf{p}(x, y)^T$ is any position in two dimensions. We assume that the evolution of the measure obeys to the following mathematical model:

$$f(\mathbf{p}, t + \Delta t) = g(\mathbf{p}_s(t), \mathbf{p}, t, f(\mathbf{p}, t)) \tag{2}$$

where f and g are two functions with a scalar value output in $\mathbb{R}$ and $\mathbf{p} = (x, y)^T$ is any position in two dimensions. Our goal is to optimize the position of the trackers as a function of their information f measured. Equation (2) is in fact a domain-specific differential equation depending on time and space, which allows to model the interaction of the physical parameter the agents have to detect with the environment. For instance, it could be an equation representing diffusion, such as the Laplace equation for heat diffusion ; it could also model transport (convection, advection), or propagation, if the physical parameter is a sound or electromagnetic wave, or it could be the Navier-Stokes equation, or a modification thereof.

We base our problem resolution on the conclusions of Lochmatter [12]. In this paper, we propose a local strategy based on the Particle Swarm Optimization (PSO) algorithm to track a mobile source. We use here a metric constraint r_p ("p" for "$perception$") beyond which a robot is not able to communicate. To the best of our knowledge, there is no existing strategy in underwater context to track a mobile source dynamically with a swarm of robots.

This paper is organized as follows. Section II gives the keys to understand the PSO and gives an analysis of its different components. In section III, we give the necessary modifications of PSO to track a mobile source, we called this strategy LCPSO. In section IV, we simulate our system. Section V concludes this paper with an opening to future work.

II. Particle Swarm Optimization

A. *Original algorithm*

This model was first published by Kennedy and Eberhart [10], and is today a swarm method to solve several problems, as optimization [10], source search [14], [26], [27], Search and Rescue (SaR) [11] and some work was done for OSL application [7], [8].

The original algorithm is based on an optimization problem: we consider N trackers measuring a value from the function below that we try to optimize:

$$f : \begin{cases} \mathbb{R}^m & \to \mathbb{R} \\ \mathbf{p}_i(t) & \mapsto f(\mathbf{p}_i(t)) \end{cases} \tag{3}$$

where $\mathbf{p}_i$ is the m-dimensional position of tracker i described by its speed vector $\mathbf{v}_i(t)$ as follows :

$$\begin{aligned} \mathbf{v}_i(t + \Delta t) = & c_0 \mathbf{v}_i(t) + c_1 \alpha_{i1}(t)(\mathbf{p}_i^b(t) - \mathbf{p}_i(t)) \\ & + c_2 \alpha_{i2}(t)(\mathbf{p}^g(t) - \mathbf{p}_i(t)) \end{aligned} \tag{4}$$

As described in (4), the speed vector at time $t + \Delta t$, with Δt the time step, is a sum of three elements:

- The previous speed vector of tracker $\mathbf{v}_i(t)$, weighted by a constant coefficient c_0.
- The difference between best historical position registered by tracker noted $\mathbf{p}_i^b(t)$ ("b" for "$best$") and its current position. The best historical position $\mathbf{p}_i^b(t)$ is the position $\mathbf{p}_i(t_i)$, with t_i between time 0 and t where measure $f(\mathbf{p}_i(t_i))$ was the greatest. This component is weighted by a constant coefficient c_1.
- The difference between the swarm's best tracker position $\mathbf{p}^g(t)$ ("g" for "$global$") and its current position. The best tracker of swarm $\mathbf{p}^g(t)$ is tracker j measuring the greatest $f(\mathbf{p}_j(t))$ among N trackers. This component is weighted by a constant coefficient c_2.

The second and last components are weighted by a random number, respectively $\alpha_{i1}(t)$ and $\alpha_{i2}(t)$, uniformly distributed in [0;1]. These random numbers provide diversity to the system, and can avoid the swarm to be trapped in a local optimum.

Using the Euler integration scheme, the updated position of tracker i is computed as the sum of its previous position and the updated speed vector as follows :

$$\mathbf{p}_i(t + \Delta t) = \mathbf{p}_i(t) + \mathbf{v}_i(t + \Delta t)\Delta t \tag{5}$$

The system is decentralized because no tracker gives instructions to other ones, but only shares its data with other elements of the swarm. The system is also very simple because there is no hierarchy, and exchanged data are limited (position and scalar values measurements). But, in robotics, some modifications or constraints are added to apply this system to problems like OSL:

- Communication constraints: the global best position cannot be efficient because of communication between individuals: in large multiagent system, there will be too much information to transmit and process at the same time [25]. Limiting the number of robots is a solution, adopted by some publications [7] in an atmospheric environment. Another solution consists of applying a spatial constraint to communication links leading to ad-hoc communciations where data are exchanged only between neighbours and gradually spread over all robots of the swarm. Another solution is to make use of a local best position only with the neighbours seen by tracker [26], [27].

- Speed limitation: this limitation is necessary for an application in physical world, but also for convergence [4]. In our model, the norm of $\mathbf{v}_i(t)$ is upper-bounded by a constant coefficient v_{max}.
- Another limitation is the motion of the trackers. Indeed, the PSO model is holonomic and supposes that, at next step, trackers can turn and move in any direction. Robots movements have to be taken into account. In [14], for example, the trackers have to turn before heading for the next calculated point. In [8], the trackers have forbidden areas where they cannot turn, so they have to recalculate their speed vector.
- With PSO, the trackers cannot avoid collisions with obstacles or each other. In [14], robots move slowly, so they do not avoid a collision but change direction after contact. Some models [6], [8] use flocking to avoid the collisions between trackers. It is also complementary with obstacle avoidance strategies [6]: a school of robots avoids an obstacle dividing the school into several parts and the school reconstitute itself when the obstacle is overcome.

B. Charged Particle Swarm Optimization (CPSO)

In the OSL domain, W. Jitmanko *et al.* [8] made some progress to track one static source diffusing a scalar value in a windy environment. Their algorithm is named CPSO. CPSO includes repulsion vectors in equation (II) to avoid a too concentrated clustering. Flocking between two robots i and j is made by vector $\mathbf{a}_{ij}(t)$ below:

$$\mathbf{a}_{ij}(t) = \begin{cases} Q_{ij}\frac{\mathbf{p}_i(t)-\mathbf{p}_j(t)}{\|\mathbf{p}_i(t)-\mathbf{p}_j(t)\|r_c^2} & \text{if } \|\mathbf{p}_i(t)-\mathbf{p}_j(t)\| < r_c \\ 0 & \text{if } \|\mathbf{p}_i(t)-\mathbf{p}_j(t)\| > r_p \\ Q_{ij}\frac{\mathbf{p}_i(t)-\mathbf{p}_j(t)}{\|\mathbf{p}_i(t)-\mathbf{p}_j(t)\|^3} & \text{otherwise} \end{cases} \quad (6)$$

We can summarize the flocking equation as follows:

- If the distance between trackers i and j $\|\mathbf{p}_i(t)-\mathbf{p}_j(t)\|$ is smaller than r_c ("c" for "*core*"), then robots are too close from each other, and they *repel strongly* each other.
- If the distance between trackers i and j $\|\mathbf{p}_i(t)-\mathbf{p}_j(t)\|$ is greater than r_p ("p" for "*perception*"), then trackers do not see each other.
- Otherwise, trackers i and j *repel smoothly* each other.

Constant Q_{ij} is set before the beginning of simulations.

So the speed vector of CPSO can be formulated as :

$$\begin{aligned} \mathbf{v}_i(t+\Delta t) = & c_0\mathbf{v}_i(t) \\ & + c_1\alpha_{i1}(t)(\mathbf{p}_i^b(t)-\mathbf{p}_i(t)) \\ & + c_2\alpha_{i2}(t)(\mathbf{p}^g(t)-\mathbf{p}_i(t)) \\ & + \sum_{j=1,j\neq i}^{N}\mathbf{a}_{ij}(t) \end{aligned} \quad (7)$$

In the speed vector, CPSO adds all the repulsive element $\sum_{j=1,j\neq i}^{n}\mathbf{a}_{ij}(t)$. The Euler integration scheme is the same as in equation (5).

This algorithm is adapted to OSL because experimental results [8] show good results for the swarm to head for one static odor source. The number of trackers and algorithm iterations determine algorithm performance. Jatmiko *et al.* [8] also added a motion constraint: if the angle to turn the tracker is in a forbidden area, calculated with wind, the robot changes its angle to avoid this area.

III. Local Charged Particle Swarm Optimization

CPSO was designed to track a static odor source in OSL domain. To track a mobile source in a constrained environment like underwater, some characteristics need to be modified.

First, the main missing point is local communication constraint. Indeed, the best tracker position of the swarm $\mathbf{p}^g(t)$ in equation (7) is global, shared with each element of the swarm. We use a local-best vector position $\mathbf{p}_i^l(t)$ (l for local), which is the position of tracker j within communication radius r_p of tracker i with the greatest measure $f(\mathbf{p}_j(t))$. This decentralization was already proposed by [26], [27] but never to track a mobile source, to the best of our knowledge.

Second, keeping in memory the best historical position $\mathbf{p}_i^b(t)$ is worthless. Indeed, the source is moving, and values dispersed in the environment are dependent on time and space. So, this information is removed in the proposed approach.

Finally, the repulsion vector $\mathbf{a}_{ij}(t)$ defined in equation (6) is uselessly cut in pieces as a function of distances r_c and r_p. We believe that the fact that $\mathbf{a}_{ij}(t) \to +\infty$ when distance $\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| \to 0$ is a sufficient condition to avoid collisions. We consider the simplified model of repulsion vectors $\mathbf{a}_{ij}(t)$ in the rest of this paper :

$$\mathbf{a}_{ij}(t) = \begin{cases} 0 & \text{if } \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| > r_p \\ Q_{ij} \frac{\mathbf{p}_i(t) - \mathbf{p}_j(t)}{\|\mathbf{p}_i(t) - \mathbf{p}_j(t)\|^3} & \text{otherwise} \end{cases} \tag{8}$$

Those considerations lead to the following model that we name LCPSO:

$$\begin{aligned} \mathbf{v}_i(t + \Delta t) = & c_0 \mathbf{v}_i(t) \\ & + c_2 \alpha_2 (\mathbf{p}_i^l(t) - \mathbf{p}_i(t)) \\ & + \sum_{j=1, j \neq i}^{N} \mathbf{a}_{ij}(t) \end{aligned} \tag{9}$$

The Euler integration scheme is the same as in equation (5). The interest of this model is to give more liberties to the robots, a decentralised agent coordination, and a capacity to track a mobile source in a complex environment.

IV. Simulations

A. Source information releasement model

The source is assumed to release a scalar information in whole environment that is function of distance and time. This function is named $f(\mathbf{p}, t)$ and is defined as the two-dimensional equation below:

$$\begin{aligned} f(\mathbf{p}, t + \Delta t) = & \frac{I_{max}}{1 + \|\mathbf{p} - \mathbf{p}_s(t))\|^d} \\ & + (1 - e^{-\frac{\Delta t}{\tau}}) f(\mathbf{p}, t) + \beta(\mathbf{p}, t) \end{aligned} \tag{10}$$

The equation (10) is a sum of three elements:

- The first element is a decay parameter which decreases with distance between source position $\mathbf{p}_s$ and any position $\mathbf{p}$ in the workspace. d (for "$distance\ decay$") depends on the nature of distance decay: if $d = 2$, degradation is characteristic of sound, if $d = 3$ it is characteristic of a magnetic field. I_{max} is the maximal intensity released by the source.
- The second element is a decrease of the previous source releasement, inspired by a first-order filter model parametered by τ.
- The last element is an additive white Gaussian noise $\beta(\mathbf{p}, t) \sim \mathcal{N}(0, \sigma)$ in the whole environment.

The I_{max} maximal condition is necessary because the equation (10) cannot release an information greater than its maximum.

Table I: Parameters initialization of diffusion equation (10)

Variable	Definition	Value
d	Distance decay	2
I_{max}	Maximal Intensity	1000 W
σ	Noise variance	250
τ	Time constant	1 s

B. Source and trackers models

In our simulations, the source follows a cart model described by [9]:

$$\begin{pmatrix} \dot{x}_s(t) \\ \dot{y}_s(t) \\ \dot{\theta}_s(t) \\ \dot{v}_s(t) \end{pmatrix} = \begin{pmatrix} v_s(t)\cos(\theta_s(t)) \\ v_s(t)\sin(\theta_s(t)) \\ u_1(t) \\ u_2(t) \end{pmatrix} \tag{11}$$

We assume that the source's trajectory is an ellipse. Controllers $u_1(t)$ and $u_2(t)$ are calculated thanks to a linearizing loop [9]. The source heads for the ellipse and follows the trajectory drawn by the ellipse. An example of a cart model is drawn Figure 1.

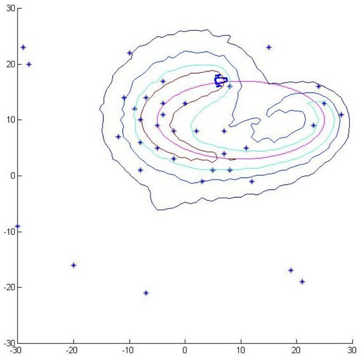


Figure 1: The source (the cart) with the released information, the ellipse and the trackers (the points)

The trackers are holonomic robots, so we apply directly LCPSO as state equation. Tracker i measures $f(\mathbf{p}_i, t)$ and adjusts its local-best position $\mathbf{p}_i^l(t)$ as a function of maximum measurement of the neighbourhood. Since the communication range is small, we make the hypothesis that information exchange is instantaneous between the trackers and is limited to their position in absolute coordinates and their measurements, without noise. Trackers are represented by points in Figure 1.

C. Simulation conditions

The robots move in a 60 m wide square on each side. The simulation lasts $t_{max} = 100$ s with $\Delta t = 0.5$ s. One thousand samples are present for each simulation. In diffusion equation (10), we use arbitrarily the parameters described in Table I.

Table II: Parameters initialization of CPSO and LCPSO

Variable	Definition	Value
N	Number of trackers	10
v_{max}	Maximal speed	5m.s^{-1}
c_0	Speed constant weight	0.8
c_2	Swarm constant weight	2
Q_{ij}	Flocking constant weight	100
r_p	Communication distance	10m
r_c	Core distance	5m

The source trajectory is an ellipse centered in $(10, 10)^T$. The semi-major axis of the ellipse source trajectory is equal to 15 m and its semi-minor axis is equal to 7 m.

At the initial state, N trackers are packed around the point $(0, 0)^T$. They have the same communication distance r_p and are spaced by $r_p/2$ from each other. This repartition allows to avoid having lost robots at the initial state. In this paper, the ”lost robots” expression means robots without neighbours in their communication radius r_p.

To compare our results, we use the mean distance between 25% of those trackers nearest to the source at every iterations. If we use the mean distance between whole swarm and source or with the tracker nearest source, it would not be representative of a swarm following a source. If $N \leq 4$, we evaluate only the distance with the nearest tracker, else we use $\lfloor 0.25N \rfloor$. In this paper, we call this distance D_{25}. When D_{25} is smaller than 15% of the workspace (here 9 m) in average during the simulation time, we consider robots have succeded to track the source and are assumed not to lose it.

We also use the number of collisions during the whole simulation to compare our algorithms. We increment the number of collisions when two trackers have a distance between them lower than $r_c/4$. r_c is defined in Table II.

D. LCPSO parameters tuning

In the following figures, we will tune the LCPSO parameters defined in Table II. We only represent the LCPSO evolution with critical values, as c_2 and N, because these are dimensioning variables of our system. Q_{ij}, c_0 and v_{max} are set as in Table II. v_{max} is set to be greater than the maximal speed of source. Indeed, our solution is not efficient if the source speed norm is greater than tracker speed norm.

Figure 2 shows the evolution of D_{25} distance as a function of c_2. This distance decreases fastly and starts to stabilize from $c_2 = 2$. We can also note that if c_2 is too much important, we have a greater dispersion of samples without improvement of their mean. So $c_2 = 2$ is a good trade-off of samples dispersion and mean.

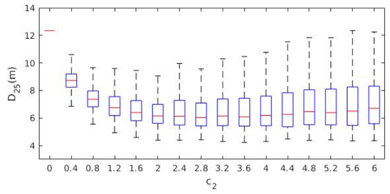


Figure 2: LCPSO evolution as a function of c_2. Box plot with 1000 samples.

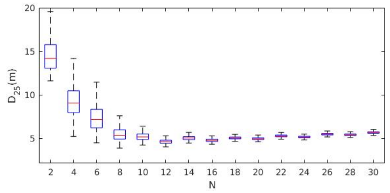


Figure 3: LCPSO evolution as a function of N. Box plot with 1000 samples.

Figure 3 shows the evolution of D_{25} distance as a function of the number of robots N. This distance decreases exponentially and starts to stabilize from $N = 8$, and is totally stabilized when $N = 12$.

E. Comparison with PSO and CPSO

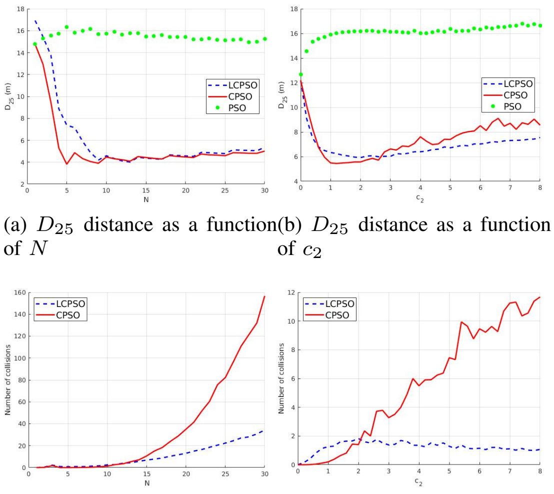
We compare the performances of different algorithms to follow a mobile source. To keep a good comparison between algorithms, the parameters are the same as in Table II. We compare only N and c_2 between different algorithms Figure 4a and 4b. Indeed, we cannot compare communication distance r_p, which is a LCPSO parameter, while c_0 and v_{max} are not key parameters and did not need to be compared.

PSO is not efficient. It is due to the fact that trackers converge at the same point fastly, so there is no possibility to compare data registered by each robot spatially. This solution cannot be applied to track a mobile source.

When we compare LCPSO and CPSO with the D_{25} distance criterion Figure 4a and Figure 4b, there is few differences, although LCPSO has better results. It is logical because algorithms are very close. However, when we observe the number of collisions Figure 4d and Figure 4c, the number of collisions of CPSO increases much more strongly than for LCPSO. It is due to the global communication with only one leader : every robot are attracted to it, and there is a crowd effect, whereas the limitation communication avoids too many robots to be attracted to the same robot, avoiding more easily collisions.

F. Influence of diffusion

Function $f(\mathbf{p}, t)$ parameters were chosen in Table I to be a good compomise between noise $\beta(\mathbf{p}, t) \sim$



(a) D_{25} distance as a function of N

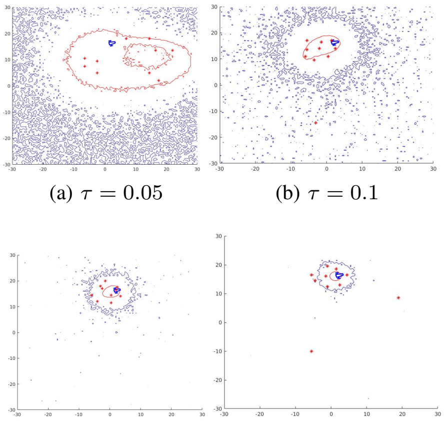
(b) D_{25} distance as a function of c_2

(c) Mean number of collisions as a function of N

(d) Mean number of collisions as a function of c_2

Figure 4: Algorithms evolution as a function of N and c_2.

$\mathcal{N}(0, \sigma)$, spatial diffusion d and temporal degradation τ. In this section, we will compare our algorithms with those parameters. Figure 5 illustrates the influence of temporal degradation τ. We illustrate the $f(\mathbf{p}, t)$ values read in the whole space by two level curves. The blue one is $f(\mathbf{p}, t) = I_{max}/20$, and the red one is $f(\mathbf{p}, t) = I_{max}/2$. We can see a "plateau" of local maxima in Figure 5a, where four trackers using the LCPSO algorithm think the source is where they are.



(a) $\tau = 0.05$

(b) $\tau = 0.1$

(c) $\tau = 0.2$

(d) $\tau = 1$

Figure 5: Scalar values in the space generated by the source when $t = 100s$

Figure 6 shows the response of the algorithms as a function of τ. We can see that LCPSO is able to track a source with few temporal degradation τ, contrary to CPSO which is lost. Indeed, the temporal dissipation is so negligible that the previous releasement has almost disappeared. However, results are more in favor of CPSO when $\tau \geq 0.2$, even if LCPSO keeps good D_{25} distances.

When we make a similar comparison with d and the noise variance σ Figure 7, we can see that CPSO and LCPSO results are quite similar when those values are weak. However, when $d = 3$ and $\sigma \geq 600$, CPSO has much more difficulties to track the source. Its variance is very large, so it means a lot of samples are trapped by a local minimum.

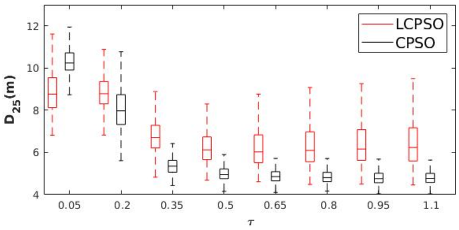


Figure 6: Algorithms evolution as a function of τ. Box plot with 1000 samples.

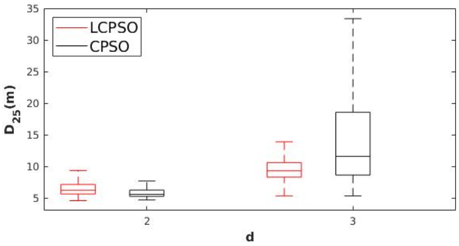


Figure 7: Algorithms evolution as a function of d. Box plot with 1000 samples.

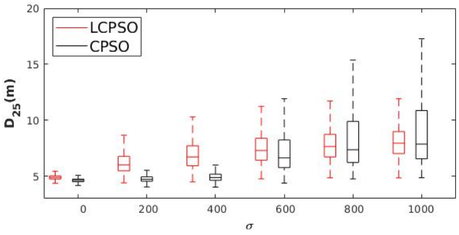


Figure 8: Algorithms evolution as a function of σ. Box plot with 1000 samples.

So LCPSO is a more flexible algorithm than CPSO, because it is able to react in presence of several maxima and to "catch" more easily the source. CPSO is limited by its best particle swarm vector $\mathbf{p}^g(t)$, which allows the tracking of only one maximum.

V. Conclusions

This paper shows the efficiency of LCPSO to track a mobile source with only one type of measured information. The local communication distance r_p adds complexity to our system, allowing our algorithm to track several local maxima, avoid more easily collisions and to be less sensitive to noise than CPSO and PSO. This degree of freedom makes the algorithm usable in ocean.

In a future work, we will try to prove our results theoretically. Few papers trying to demonstrate the PSO convergence exist [4]. There are even less dedicated to derivatives of this algorithm. We will concentrate our convergence efforts on the limitations of our algorithm to be stable, as parameters c_0, c_2 and N.

Moreover, some progress in flocking formation and convergence proofs [13], [17], [20], [21] were made in the past, with a maturation conducting to a success of real experiments [24]. We want to use those tools to control the inter-robot distance in the swarm and the heading of our robots. The control of inter-robot distance is crucial to avoid collisions and to see the surface taken by the swarm and needed to track a source.

We also want to use localization to add an individual component permitting a track not only directed by social characteristics. Moreover, our model only takes into account the diffusion parameters of the source. We want to add a dynamic environment.

References

[1] D. Angley, S. Suvorova, B. Ristic, W. Moran, F. Fletcher, H. Gaetjens, and S. Simakov. Sensor scheduling for target tracking in large multistatic sonobuoy fields. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3146–3150, March 2017.

[2] Lara Briñon Arranz. *Cooperative control design for a fleet of AUVs under communication constraints*. Theses, Université de Grenoble, November 2011.

[3] X. Cao, D. Zhu, and S. X. Yang. Multi-AUV target search based on bioinspired neurodynamics model in 3-D underwater environments. *IEEE Transactions on Neural Networks and Learning Systems*, 27(11):2364–2374, Nov 2016.

[4] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, Feb 2002.

[5] M. I. Hasan, M. Dautta, and M. A. Haque. A study of ferromagnetic object detection in ocean environment using empirical mode decomposition. In *TENCON 2017 - 2017 IEEE Region 10 Conference*, pages 2132–2135, Nov 2017.

[6] Suranga Hettiarachchi and William M. Spears. Distributed adaptive swarm for obstacle avoidance. *International Journal of Intelligent Computing and Cybernetics*, 2(4):644–671, 2009.

[7] W. Jatmiko, W. Pambuko, P. Mursanto, A. Muis, B. Kusumoputro, K. Sekiyama, and T. Fukuda. Localizing multiple odor sources in dynamic environment using ranged subgroup PSO with flow of wind based on open dynamic engine library. In *2009 International Symposium on Micro-NanoMechatronics and Human Science*, pages 602–607, Nov 2009.

[8] W. Jatmiko, K. Sekiyama, and T. Fukuda. A PSO-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement. *IEEE Computational Intelligence Magazine*, 2(2):37–51, May 2007.

[9] Luc Jaulin. *Mobile Robotics*. The MIT Press, 2015.

[10] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.

[11] A. S. Kumar, G. Manikutty, R. R. Bhavani, and M. S. Couceiro. Search and rescue operations using robotic darwinian particle swarm optimization. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1839–1843, Sept 2017.

[12] Thomas Lochmatter. Bio-inspired and probabilistic algorithms for distributed odor source localization using mobile robots. page 135, 2010.

[13] Ugo Lopez, Jacques Gautrais, Iain D. Couzin, and Guy Theraulaz. From behavioural analyses to models of collective motion in fish schools. *Interface Focus*, 2(6):693–707, 2012.

[14] J. Pugh and A. Martinoli. Inspiring and modeling multi-robot search with particle swarm optimization. In *2007 IEEE Swarm Intelligence Symposium*, pages 332–339, April 2007.

[15] F. Rahbar, A. Marjovi, P. Kibleur, and A. Martinoli. A 3-D bio-inspired odor source localization and its validation in realistic environmental conditions. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3983–3989, Sept 2017.

[16] B. Ristic, D. Angley, S. Suvorova, B. Moran, F. Fletcher, H. Gaetjens, and S. Simakov. Gaussian mixture multitarget-multisensor bernoulli tracker for multistatic sonobuoy fields. *IET Radar, Sonar Navigations*, 11(12):1790–1797, 2017.

[17] Yilun Shang and Roland Bouffanais. Consensus reaching in swarms ruled by a hybrid metric-topological distance. *The European Physical Journal B*, 87(12):294, Dec 2014.

[18] M. Shaukat, M. Chitre, and Sim Heng Ong. A bio-inspired distributed approach for searching underwater acoustic source using a team of AUVs. In *2013 MTS/IEEE OCEANS - Bergen*, pages 1–10, June 2013.

[19] Diana Spears, Dimitri Zarzhitsky, and David Thayer. Multi-robot chemical plume tracing. In Lynne E. Parker, Frank E. Schneider, and Alan C. Schultz, editors, *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, pages 211–221, Dordrecht, 2005. Springer Netherlands.

[20] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, part I: fixed topology. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 2, pages 2010–2015 Vol.2, Dec 2003.

[21] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents part II: dynamic topology. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*, volume 2, pages 2016–2021 Vol.2, Dec 2003.

[22] Y. Tian, W. Li, and Fumin Zhang. Moth-inspired plume tracing via autonomous underwater vehicle with only a pair of separated chemical sensors. In *OCEANS 2015 - MTS/IEEE Washington*, pages 1–8, Oct 2015.

[23] Frans Van Den Bergh. *An Analysis of Particle Swarm Optimizers*. PhD thesis, Pretoria, South Africa, South Africa, 2002. AAI0804353.

[24] Gábor Vásárhelyi, Csaba Virágh, Gergő Somorjai, Tamás Nepusz, Agoston E. Eiben, and Tamás Vicsek. Optimized flocking of autonomous drones in confined environments. *Science Robotics*, 3(20), 2018.

[25] Gerhard Weiss. *Multiagent Systems*. The MIT Press, 2013.

[26] S. Xue, J. Zhang, and J. Zeng. Parallel asynchronous control strategy for target search with swarm robots. *International Journal of Bio-Inspired Computation*, 1(3):151–163, 2009.

[27] L. Zhiqi, X. Songdong, Z. Jianchao, Z. Jing, and Z. Guoyou. An evaluation of PSO-type swarm robotic search: Modeling method and controlling properties. In *2010 International Conference on Networking, Sensing and Control (ICNSC)*, pages 360–365, April 2010.