# Energy-Saving Decision Making for Aerial Swarms: PSO-based Navigation in Vector Fields

Palina Bartashevich
Faculty of Computer Science
University of Magdeburg
Germany
Email: palina.bartashevich@ovgu.de

Doreen Koerte
Faculty of Computer Science
University of Magdeburg
Germany
Email: doreen.koerte@st.ovgu.de

Sanaz Mostaghim
Faculty of Computer Science
University of Magdeburg
Germany
Email: sanaz.mostaghim@ovgu.de

***Abstract*—This paper presents the Navigation Wind PSO (NW-PSO) as a search mechanism for aerial micro-robots with limited battery capacity acting in environments with unknown external dynamics (such as wind). The proposed method uses the concepts of multi-criteria decision making and let the individuals to decide on their movements almost following the flow with small course deviations in order to save as much energy as possible. One of the goals is to investigate how the arising premature energy-loss on individual level affects the performance of collective search. The experiments show that NW-PSO can save a good amount of energy as well as perform search behavior with good approximation of the global optimal solution almost without awareness regarding disturbance factors and particle loss.**

## I. Introduction

Recently there has been a surge of interest in aerial swarm systems. Flying robots cooperating in swarms could cover large areas of surface and perform search and rescue operations without human intervention [21], [20]. Acting autonomously and in order to be able to reach the goal, such systems must additionally take their energy consumption into consideration [8]. The crucial point is that the energy of small aerial robots depends on on-board batteries and therefore is finite and unreliable [23]. The limited amount of battery power available for flying is normally only 15-20 minutes [16]. The situation getting worse when external perturbations arise, e.g. wind, as they cause instability to the robots. Flying in windy conditions is forcing the motors to work harder to neutralize the errors, what leads to even faster battery loss. In this paper we focus on the wastes of energy caused by the dynamics of the external environment (such as wind, flow, bad weather conditions, etc.) and propose a new PSO-based search mechanism for aerial micro-robots, which could be further employed in robotic search scenarios to extend the activity of the systems with limited battery capacity.

In a previous work [2], we modeled the unknown dynamics of the environment by using Vector Fields and developed a new PSO-based approach called VFM-PSO, which can cope with arising perturbations, despite incomplete awareness of the corresponding vector fields structures. However, this approach may not be practically beneficial in real aerial robotic search applications, as it does not consider the amount of energy required for moving to the next position. According to the method proposed in [2], if the individuals get into the areas with known (explored) perturbations they show resistance to them by total neutralization of disturbance factor. This is good in terms of optimization (search), as it behaves better then as a baseline search algorithm. However, in terms of energy consumption, such behavior is unprofitable/disadvantageous as it needs a lot of effort especially in areas of inconvenient flows. According to [2], while the area is unknown, the combined movement of velocity vector calculated by standard PSO and vector field at the according positions can also lead to over speed-up or sharp turn of the robots. That might be very dangerous, especially in strong winds with opposite directions, and even cause damage of the robots. Since the robots cooperating in a swarm, the premature loss of individuals might have an essential influence on the collective search process. In this work, we aim to prevent that.

In this paper, different from our earlier work [2], the key aspect is optimizing the energy consumption, and not just the fitness landscape. To our best knowledge, this is the first effort that addresses the issue of energy consumption of aerial autonomous swarms in collective search process within influence of external environment.

The main goal is to propose a model to estimate and minimize additional energy costs arising in a swarm by the affects of unknown external dynamics during the search process. The unknown dynamics are modeled by Vector Fields and involves the concept of *Information Map*, which stores the values of the explored vector fields areas in a matrix as described in [2]. We propose a new PSO-based search mechanism called Navigation Wind PSO (NW-PSO) which let particles to decide on their movements almost following the flow with small course deviations in order to save as much energy as possible to find the global optimum. We suppose that doing by this, the robots will be better served by the battery and could be able to accomplish the task within defined energy limits without loss of individuals. The proposed approach uses the concepts from multi-criteria decision making, where the individual selects energy efficient target points using Pareto Front inside the local neighborhood defined by its current vector fields vector. Since the vector fields structures are unknown and directly affect the choice, identification of the best target point is presented by two models of multi-criteria decision making depending on the *Information Map*. The main research questions are to investigate in which vector fields in this case the optimal solution could be found and how significant is the role of information regarding surrounding vector field on obtaining reasonable results under energy constraints. In the experiments we analyze the emerged behavior under influence of different vector fields and examine how the premature energy-loss on individual level (of certain particles) affects the performance of collective search. The results prove that with NW-PSO we can save a good amount of energy as well as perform search behavior with good approximation of the global optimal solution almost without awareness regarding disturbance factors and particles loss.

The remainder of this paper is structured as follows. We describe the background and analyze some related works in Section II. In Section III we propose our model. Section IV presents the experiments and the results with some discussion. The paper is concluded in Section V.

## II. Related Work and Background

Energy consumption of aerial micro-robots under external influences in search scenarios has not been studied so far in the literature. However, several researches concerning path and task planning in dynamic environments were done in [24], [22], [5], [18], [25], [1], proposing different approaches in order to conserve energy such as: recharging optimization for flight tour missions under the wind uncertainty [18]; the ability of drones to fly at lower altitudes [25]; exploitation of wind energy to extend the flight duration during the route from a starting point to another [1], etc. Some other approaches addressed the works in oceanic flows [6], [10], considering construction of time- and energy-optimal paths to leverage the dynamics of the surrounding flows within limited budgets of energy for autonomous surface and underwater vehicles. However, all of these studies do not consider collective search process, i.e. optimization.

Stirling *et al.* in [17] introduced a novel energy-efficient search strategy to coordinate flying robots in indoor environments, assuming that they can land or attach to ceilings. But in scenarios with influence of external factors, in the case of trapping into the areas of strong perturbations, it might be even impossible for the robot to produce such sorts of behavior, e.g. landing in high speed leads to damage of the robot; the lack of points in a search space to stick to, etc. However, the methodology proposed in this paper could deal with such situations and is inspired by animal search and orientation strategies with a lack of information about potential targets in flows considered in [4], [12], [14], [19], [13] and uses the concepts of PSO [7]. In [15] it was firstly introduced a PSO-based search mechanism for aerial micro-robots, addressing the question of efficient energy consumption for systems with limited capacity, but it does not consider the influence of external environment. This paper presents a first attempt to cover the gap in research on energy saving swarm flights for a collective search in environments with unknown external dynamics, addressing the application of PSO. In [10] it was concluded that velocity restriction plays sufficient role for time, but not for energy. That approves one of the assumptions according velocity restriction of the presented model in this work. Teppo *et al.* in [11] showed that if disturbances are continuous, the benefit from temporary stabilization is only momentary, so according to our model, taking into consideration the perturbations arising in the next iteration could improve the results of search. Since the decision on the individuals movement to the next position is conditioned on conflicting parameters, the selection is made based on multi-criteria decision making technique from the set of so called Pareto-optimal solutions. A solution $\vec{x}^*$ is called Pareto-optimal for minimization problems, if there is no other solution $\vec{x}'$ in the search space $S$ so that:

$\forall i : f_i(\vec{x}') \leq f_i(\vec{x}^*)$ and $\exists j : f_j(\vec{x}') < f_j(\vec{x}^*)$.

Accordingly we can use the same definition to compare the solutions. In this case, a solution $\vec{x}_1$ *dominates* another solution $\vec{x}_2$ (denoted by $\vec{x}_1 \prec \vec{x}_2$), if:

$\forall i : f_i(\vec{x}_1) \leq f_i(\vec{x}_2)$ and $\exists j : f_j(\vec{x}_1) < f_j(\vec{x}_2)$.

The solutions which do not mutually dominate each other are called non-dominated solutions. In this paper selection of one of the Pareto-optimal (or non-dominated) solutions is done using the weighted sum method [9], which transforms multiple conflicting objectives into an aggregated objective function by multiplying each of the objective functions by a weighting factor and summing up all weighted objective functions (as shown in Section III-D).

## III. Methodology

In this section, we introduce our new model which is designed as navigation collective search based on concepts of multi-criteria decision making for minimizing additional energy costs arising in aerial swarms under influence of unknown external dynamics. The dynamics are modeled by vector fields as in [2]. In order to verify the validity of proposed method, we consider three strategies of aerial swarms behavior which can be applied to cope with such dynamics during the search process, i.e. complete compensation (denoted as **C-PSO**), partial compensation (denoted as **CW-PSO**) and navigation (denoted as **NW-PSO**, which is a new method proposed in the paper). These approaches are based on the concepts of Algorithm 1, which presents the general framework of [2] with extension by our new model of energy costs described in Section III-A.

The proposed mechanisms of search behavior are influenced equally in their movements by vector fields but use different strategies for treatment defined by *ComputeVelocity()* (explained in Sections III-B and III-D). Energy calculation, i.e. *ComputeEnergy()*, is the same for all swarms under external dynamics influence and computed as follows:

### A. Energy Computations

We assume that going in PSO-direction does not require any effort in conditions of no wind (external influences). Since in such conditions robots always use the same units of energy per certain measurement (e.g. distance, time, etc.), we do not consider these expenses in our proposed model of energy

**Algorithm 1:** General Framework

```
Input : Number of N_p PSO individuals
t = 0
Initialize the PSO individuals:
for i = 1 to N_p do
    x_i(t) = StartPosition(S)
    v_i(t) = 0
    e_i^battery(t) = e_max
end
while Stopping Criterion not fulfilled do
    Wait for the Map from the Explorer Population:
    Map = Receive(Map)
    for i = 1 to N_p do
        x_g(t) = FindGlobalBest()
        v_i(t+1) = ComputeVelocity (x_i(t), x_g(t), Map)
        x_i(t+1) = UpdatePosition(v_i(t), x_i(t))
        e_i^battery(t+1) = ComputeEnergy(v_i(t+1),
          Map)
    end
    t = t + 1
end
```

calculations. The attention is paid to the additional energy wastes arising due non-uniform flows. In this case the values of energy costs differ even within one and the same vector field, depending on the value of magnitude and co-directionality with desired direction of movement at current position. In the following we introduce a simplified model of estimating such energy costs for the individuals, which is modeled by considering the required energy related to the definition of total kinematic energy for rotation objects.

The energy of an individual is defined by the battery, i.e. $e_i^{battery}(t)$, which is full at the beginning and discharges by moving and turning of the robot. The initial amount of battery energy is the same for all individuals, i.e. $e_{max}$. The calculation of energy wastes for one individual is inspired by total kinematic energy for rotation objects and performed as the sum of rotational (i.e. the energy for turn, $E^{rot}$) and translation energy (i.e. the energy for movement, $E^{trans}$):

$$E^{kinetic} = E^{rot} + E^{trans} \tag{1}$$

The rotational energy $e_i^{rot}$ estimates how strong the particle rotates in one iteration. $e_i^{rot}$ is defined by the angle (denoted as $\boldsymbol{\omega}$) between the current velocity vector $\vec{v}_i(t+1)$ and the old one $\vec{v}_i(t)$, i.e. on previous iteration:

$$\begin{aligned} e_i^{rot}(t+1) &:= \omega = \angle(\vec{v}_i(t+1), \vec{v}_i(t)) := \\ &\stackrel{def}{:=} arccos\left(\frac{(\vec{v}_i(t)), \vec{v}_i(t+1))}{||\vec{v}_i(t)|| \cdot ||\vec{v}_i(t+1)||}\right) \end{aligned} \tag{2}$$

Where $(\cdot,\cdot)$ is the dot product (scalar product) of two vectors; $||\cdot||$ denotes the length of a vector.

The translation energy $e_i^{trans}$ estimates how strong the particle needs to sheer away the current vector field and therefore to spend energy for necessary movement correction. These corrections are defined by the intensity of target movement vector (denoted as $\mathbf{I}$) and the angle (denoted as $\boldsymbol{\alpha}$) between this vector and corresponding vector of the vector field at the particles position. $e_i^{trans}$ is calculated as the sum of these two parameters:

$$\begin{aligned} e_i^{trans}(t+1) &:= I + \alpha = \\ &= ||\vec{v}_i(t+1)|| + \angle(\vec{VF}(\vec{x}_i(t)), \vec{v}_i(t+1)), \end{aligned} \tag{3}$$

where $\angle(\vec{VF}(\vec{x}_i(t)), \vec{v}_i(t+1))$ is calculated by Equation(7). To sum up, amount of wasted energy at one iteration step is computed as:

$$E^{kinetic} = \omega + I + \alpha, \tag{4}$$

where $\boldsymbol{\omega}$ is the rotation angle; $\boldsymbol{\alpha}$ is the steering control angle; $\mathbf{I}$ is the magnitude of movement vector that indicates the target direction; all the parameters are defined as above.

At the end of each iteration, the energy of individual, which is available for the next movement, is evaluated as follows:

$$e_i^{battery}(t+1) = e_i^{battery}(t) - e_i^{kinetic}(t+1), \tag{5}$$

The individual stops the movement $\vec{v}_i(t+1) = 0$, i.e. dies, when its battery is totally empty, i.e. $e_i^{battery}(t) = 0$.

The total energy of the whole swarm is evaluated by the sum of all batteries values in entire population:

$$E^{total}(t) = \sum_{i=1}^{N_p} e_i^{battery}(t) \tag{6}$$

In order to keep the model as simple as possible, we have met several assumptions. In our model, we suggest that each individual is equipped with velocity safety lock and can not gain its speed in one iteration more than defined velocity limit $v_{max}$. We assume that this action does not require any additional energy costs.

### B. *Compensation*

Full compensation behavior is presented by Compensative PSO, i.e. **C-PSO**. C-PSO performs standard PSO (uses PSO equations [7] for *ComputeVelocity()* and *UpdatePosition()*), where particles move directly towards the current optimum regardless of distractions caused by vector field as shown in Fig.1(a). This can result in high energy wastes since the particle moves in another direction than current predefined vector of the vector field, e.g. high values of the steering control angle $\alpha$. Moreover, acting in unknown flows, such behavior implies strong internal resistance, which is very harmful for robot engines.

To deal with unknown disturbances, Compensative Wind PSO, i.e. **CW-PSO**, uses explorer population gathering the information about vector field in *Information Map*. Thus, in explored areas particles could behave wisely and realize their *movement by a correction vector* (which defines velocity vector $\vec{v}_i(t+1)$) as shown in Fig.1(b). As the result the individual move to originally desired location. Such behavior does not lead to the energy conservation in sense that the resultant movement is the same as in C-PSO. But at the same time, in non-aware areas high energy costs can be occasionally reduced due to the redirections caused by the vector field and particles own velocity as shown in Fig.1(c). In this way, energy consumption of CW-PSO is supposed to a greater or lesser extent depend on the *Information Map*. The strategy of **CW-PSO** was already proposed in [2] as VFM-PSO, addressing only the issue of convergence, and uses the same calculations for *ComputeVelocity()* and *UpdatePosition()*).

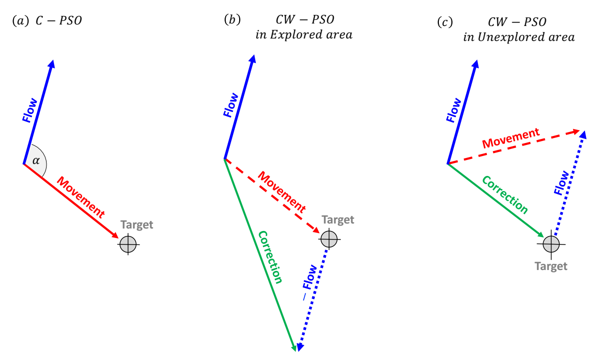


Fig. 1: Triangles of velocities with respect to the target direction under vector field influence for C-PSO (a) and CW-PSO: (b) - in explored area, (c) - in unexplored.

### C. NW-PSO

In the following we introduce Navigation Wind PSO (i.e. **NW-PSO**), based on general framework of Algorithm 1, which contains in *ComputeVelocity()* the multi-criteria decision making process for each individual in population.

The main goal of **NW-PSO** is to spend as less energy as possible in order to find the global optimal solution. Therefore, an individual $i$ has to find such direction to go from reachability set, i.e. $\vec{x}_{target} \in R(\vec{x}_i, \vec{VF}_i)$, following which it can be hit further by the vector field defined at target point, i.e. $\vec{VF}_{tg} = \vec{VF}(\vec{x}_{target})$, to its primary goal $\vec{x}_g$. In such way, the individual does not use its own energy to accelerate, so it can save energy and at the same time get closer or even reach the initial destination (as shown in Fig.2). The main steps for selecting energy efficient target points, i.e. navigation mechanism, are shown in Algorithm 2.

Primary goals $\vec{x}_g(t)$ are calculated as the standard PSO-vectors for each particle as in non-wind conditions. The first step is to calculate the set of possible target points to go for an individual $i$, which is defined by reachability set: $R(\vec{x}_i, \vec{VF}) = \{\vec{x}^l_{target}\}^N_{l=1}$. For this purpose we have to define the search neighborhood: We assume that each individual $i$ is equipped with navigation sensors, so that it can look for possible directions to go within sensors visibility range

**Algorithm 2:** Navigation for individual $i$

```
Input : x_g(t), x_i(t) and Map
Output : x_target(t)
Initialize Neighborhood Ω:
Ω := Set(θ, k, θ*)
Define Reachability Set:
count = 0
for l = 1 to N do
    x^l_target(t) := ComputeTargets(VF_i, Ω)
    if Map(x^l_target(t)) ≠ 0 then
        count = count + 1
    end
end
Par(i) := EnergyProfit (count, x_i(t), x_g(t))
x_target(t) = MCDM (Par(i))
```

$\Omega(\theta, r, k)$ (denoted as grey semicircle in Fig.2), defined by the angle range $\theta$ (in degrees), search radius $r$ and precision of search $k$.

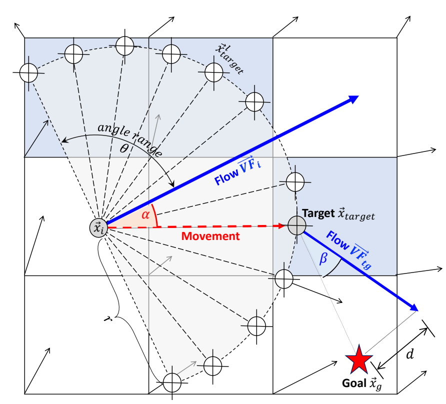


Fig. 2: Scenario 1: Navigation using *Information Map*. Grey semicircle defines the search neighborhood $\Omega(\theta, r, k)$. The points on the border of semicircle, denoted according to the precision $k$ as white targets $\vec{x}^l_{target}$, form the reachability set $R(\vec{x}_i, \vec{VF})$. Painted-over cells define explored areas.

The angle range $\theta$ determines the rotation possibility of the individual according to the direction of the current flow. It is symmetric about the vector $\vec{VF}_i$ as shown in Fig.2 and defines the maximum course deviation from it. The direction of $\vec{VF}_i$ defines zero reference. The smaller range means the higher co-directionality with $\vec{VF}_i$ and as the result less energy wastes. The search radius is defined by the norm of the current flow, i.e. $r = ||\vec{VF}_i||$. Given that the particle can only move with a limited velocity, in case if $||\vec{VF}_i|| > v_{max}$, the search radius $r$ is set to $v_{max}$. The number of directions to go, i.e. $N$, within the angle range is defined by precision $k$ ($k \in \mathbb{N}_{>0}$), which means that every $k^{th}$ vector from the angle range could be chosen as a possibility for movement, i.e. $N = \frac{\theta}{k}$. So the set of target points for an individual $i$, i.e. reachability set, is defined as follows:

$$R(\vec{x}_i, \vec{VF}) = \{\vec{x}^l_{target} \in \Omega(r, \theta, k) : ||\vec{x}^l_{target} - \vec{x}_i|| = r, 1 \leq l \leq \frac{\theta}{k}\} \quad (7)$$

In case, if the primary goal occurs inside the visibility range (the particle is already close to the goal), we increase the angle range for this individual, i.e. $\theta^* >> \theta$, and set the search radius $r$ to the Euclidean distance value between position of the individual and the global best. As a result the swarm can really converge, if it is close to the optimum:

$$(r, \theta) = \begin{cases} (v_{max}, \theta), & \text{if } ||\vec{VF}_i|| > v_{max} \\ (dist(\vec{x}_g(t), \vec{x}_i(t)), \theta^*), & \text{if } ||\vec{x}_g(t) - \vec{x}_i(t)|| \leq r \end{cases}$$

### D. Navigation Decision Making

In order to select the best target point to go, the individual computes the *energy profit* for each targets candidate, which depends on the *Information Map*. The position of each target point is checked in the *Information Map*, whether it is in explored area or not, i.e. $Map(\vec{x}^l_{target}) \neq 0$. *EnergyProfit()* determines the set of parameters, which will be used in multi-criteria decision making, i.e. *MCDM()*. According to the values of *Information Map*, there are 2 scenarios for *EnergyProfit()* calculations:

**Scenario 1**. If $card(R(\vec{x}_i, \vec{VF}) \cap Map) \geq N/2$, then the *EnergyProfit()* for candidate solutions is defined by the parameter set: $(\alpha, \beta, d)$ as shown in Fig.2.

Where $\alpha$ is the angle between target vector and the current vector field $\vec{VF}_i$; $\beta$ is the angle between vector flow from the corresponding target point $\vec{VF}_{tg}$ and the direction from $\vec{x}_{target}$ to $\vec{x}_g$; $d$ is the Euclidean distance between the head of $\vec{VF}_{tg}$ and $\vec{x}_g$. The calculations of angles are made using the same definition as in Equation 2 Section III-A.

Since the calculations of parameters $\beta$ and $d$ depend on the *Information Map*, i.e. use the a priori unknown values of the vector field at target points, the amount of candidate solutions $N$ for this case is defined only by the target points which get into explored areas, i.e. $card(R(\vec{x}_i, \vec{VF}) \cap Map \neq \emptyset)$.

**Scenario 2**. If $card(R(\vec{x}_i, \vec{VF}) \cap Map) < N/2$, then then the *EnergyProfit()* is defined only by: $(\alpha, d)$ as shown in Fig.3.

In this case the calculations of the parameters, i.e. $\alpha$ and $d$, do not depend on the *Information Map*. Since $\alpha$ is defined by the current flow $\vec{VF}_i$, which particle can estimate (but not save) despite *Information Map*, and $d$ is the Euclidean distance between target candidate point $\vec{x}^l_{target}$ and $\vec{x}_g$. $\alpha$ is calculated the same as in Scenario 1. The amount of candidate solutions $N$ in this case is defined by all possible target points which can be obtained with given precision $k$ and angle range $\theta$ as was described in Section III-C .

We consider these two scenarios, because in case if there are too few explored areas in the visibility range (as in Scenario 2), the results with parameters defined by Scenario 1 might be misleading, e.g. explored area is only in non

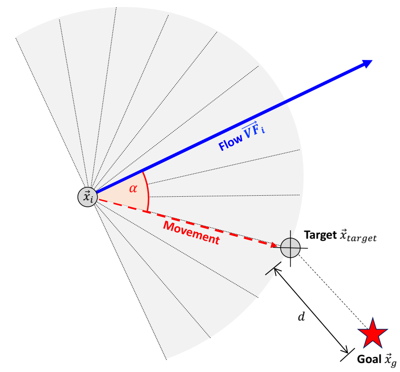


Fig. 3: Scenario 2: Navigation without *Information Map*.

energy efficient direction (high $\alpha$ values). However, regardless of the scenario all the parameters are in conflict with each other; the candidates with low energy costs (small $\alpha$ values) can be far away from goal $\vec{x}_g$ and prevent convergence. In this case, the individuals must select one of the $N$ candidate solutions $\vec{x}^l_{target}, 1 \leq l \leq N$, using concepts from multi-criteria decision making (denoted as MCDM () in Algorithm 2). All the parameters $(\alpha, \beta, d)$ are normalized and the best target point $\vec{x}_{target}$ is evaluated by the weighted sum approach as follows:

Each target point $l$ of the individual $i$ is assigned with a weight vector for the parameters according to the current scenario for this individual $i$: $w^1 = (w^1_\alpha, w^1_\beta, w^1_d), w^2 = (w^2_\alpha, w^2_d)$, where $||w^1||, ||w^2|| = 1$. The weight of each parameter shows its priority. In this paper the values for weights $w^1, w^2$ are selected constant by the experiments for all individuals and differs only within vector fields. The table with weights values is provided in Section IV (i.e. Table II) according to the vector fields considered in this paper. After setting the preferences (i.e. parameters and weights according to the arose scenario) for each possible target point $l$, the profit of each candidate ranks by:

$$Profit(l) = \begin{cases} w^1_\alpha \cdot \alpha + w^1_\beta \cdot \beta + w^1_d \cdot d, & \text{if } Scenario1 \\ w^2_\alpha \cdot \alpha + w^2_d \cdot d, & \text{if } Scenario2 \end{cases}$$

Where $l = 1, \cdots, N$. The target $l$ with the lowest *profit* will be selected as the most profitable, i.e. the best target $\vec{x}_{target}$, by the individual $i$. Then the individual movement vector in *ComputeVelocity()* of Algorithm 1 is calculated as follows:

$$\vec{v}_i(t+1) = \begin{cases} \vec{x}_{target}(t) - \vec{x}_i(t) + \vec{VF}_{tg}, & \text{if } Scenario1 \\ \vec{x}_{target}(t) - \vec{x}_i(t), & \text{if } Scenario2 \end{cases}$$

However, only the velocity vector for *target position* (as in Scenario 2) is used for calculating energy costs *ComputeEnergy()* in Algorithm 1 regardless of the scenario type.

## IV. Experiments

### A. Parameter Settings

In the experiments we compare three different algorithms described in Section III, **C-PSO**, **CW-PSO** and **NW-PSO**, all of which illustrate the proposed methods in this paper. All the algorithms use the same model of energy computations described in Section III-A and the same initial battery capacity $e_{max}$ for all individuals. The algorithms parameters are selected as in Table I. Inertia weight $w = 0.6$ and acceleration coefficients $C_1 = C_2 = 1$ are set according to our extensive preliminary tests. The optimization stops after 100 iteration.

TABLE I: Parameter values

| Parameter | Value |
|---|---|
| Population size $N_p$ | 30 |
| Number of explorers $N_e$ | 20 |
| Velocity limit $v_{max}$ | 2 |
| Battery capacity $e_{max}$ | 100 |
| Angle range $\theta$ | 90° |
| Angle extension $\theta^*$ | 150° |
| Precision $k$ | 2 |

The flights are modeled in $d = 2$ dimensional search space. $N_p$ aerial robots are placed with random positions in search area (arena) defined as $x_1, x_2 \in [-15, 15]$. Three test problems such as Sphere, Ackley and Rosenbrock from the literature are used for the experiments. These test problems can very well simulate natural search terrains, i.e. while Sphere is for the simple tests, Ackley and Rosenbrock capture terrains with lots of local optima and a flat plateau respectively. The optimal solution of all these problems is shifted to $(-10, 10)$. The value of the optimal solution is the same as for standard test problems, i.e. equals zero. In the experiments 5 various vector fields (denoted by VF1 to VF5 in Table II) are used to model the external dynamics defined on the mesh of $31 \times 31$ units of the search space. VF1 to VF3 were already considered in [2], while VF4, defined as $V\vec{F}4(x_1, x_2) = (-0.7y, 0)$, and VF5, defined by random values $rand(-1, 1)$, are the new ones.

TABLE II: Preference values in decision-making process for five vector fields according to the Scenario.

| | Scenario 1 | | | Scenario 2 | |
|---|---|---|---|---|---|
| | $w_\alpha^1$ | $w_d^1$ | $w_\beta^1$ | $w_\alpha^2$ | $w_d^2$ |
| VF1: “Cross” | 0.8 | 0.15 | 0.05 | 0.75 | 0.25 |
| VF2: “Rotation” | 0.8 | 0.19 | 0.01 | 0.7 | 0.3 |
| VF3: “Sheared” | 0.75 | 0.15 | 0.1 | 0.7 | 0.3 |
| VF4: “Bi-directional” | 0.1 | 0.2 | 0.7 | 0.6 | 0.4 |
| VF5: “Random” | 0.5 | 0.2 | 0.3 | 0.7 | 0.3 |

The results are compared in terms of *fitness* (best function values obtained during the all iterations), the *total swarm energy* (calculated by Equation 6), *success* and *survival rate*. The *success rate* indicates the percentage of runs, in which *fitness* is smaller than a certain threshold (e.g., 0.01) by the end of the search process, while the *survival rate* shows the ratio (in percentage) of the particles which are still full of energy, i.e. $e_i^{battery} \neq 0$, by the end of the iterations. All the experiments are run for 100 times and the median values and the corresponding standard errors are reported in Table III.

TABLE III: Results for the three test problems in five vector fields (median values and standard errors (std)). “Energy” and “fitness” indicate the total amount of energy and the best function value obtained by the swarm. Bold values indicate the best fitness value (i.e. convergence) and the corresponding energy wastes.

| | | energy | ±std | fitness | ±std |
|---|---|---|---|---|---|
| Sphere | | | | | |
| VF1 | NW- | 545,532 | 25,188 | 3,132 | 0,727 |
| | CW- | 2501,371 | 26,752 | 5,711 | 0,781 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |
| VF2 | NW- | **1268,953** | 27,494 | **0,002** | 0,001 |
| | CW- | 2990,401 | 2,429 | 0,380 | 0,129 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |
| VF3 | NW- | 556,342 | 13,429 | 0,186 | 0,055 |
| | CW- | 1165,756 | 28,306 | 1,361 | 0,157 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |
| VF4 | NW- | **618,595** | 21,093 | **0,005** | 0,002 |
| | CW- | **1598,155** | 112,187 | **0,070** | 0,007 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |
| VF5 | NW- | **2424,924** | 16,623 | **0,000** | 0,000 |
| | CW- | **3000,00** | 0,00 | **0,001** | 0,000 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |
| Rosenbrock | | | | | |
| VF1 | NW- | 518,321 | 18,216 | 178,801 | 65,811 |
| | CW- | 2542,374 | 26,559 | 395,208 | 91,004 |
| | C- | 3000,000 | 0,000 | 0,002 | 0,002 |
| VF2 | NW- | **1299,938** | 21,839 | **0,074** | 0,013 |
| | CW- | 2968,305 | 5,923 | 27,293 | 11,098 |
| | C- | 2999,750 | 0,240 | 0,003 | 0,001 |
| VF3 | NW- | 757,959 | 28,640 | 2,147 | 0,315 |
| | CW- | 1207,998 | 31,455 | 178,021 | 173,392 |
| | C- | 2999,206 | 0,326 | 0,004 | 0,004 |
| VF4 | NW- | 541,178 | 15,989 | 0,606 | 0,106 |
| | CW- | 2597,197 | 45,318 | 1,989 | 0,416 |
| | C- | 3000,000 | 0,000 | 0,001 | 0,001 |
| VF5 | NW- | 2440,865 | 15,906 | 0,221 | 0,084 |
| | CW- | **3000,000** | 0,000 | **0,016** | 0,002 |
| | C- | 3000,000 | 0,000 | 0,001 | 0,000 |
| Ackley | | | | | |
| VF1 | NW- | 603,961 | 22,072 | 3,714 | 0,384 |
| | CW- | 2501,210 | 28,404 | 6,141 | 0,297 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |
| VF2 | NW- | **1232,115** | 25,596 | **0,101** | 0,027 |
| | CW- | 2987,347 | 3,078 | 1,252 | 0,182 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |
| VF3 | NW- | 659,721 | 17,090 | 2,014 | 0,228 |
| | CW- | 1140,529 | 27,352 | 4,225 | 0,118 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |
| VF4 | NW- | 631,198 | 20,776 | 0,519 | 0,087 |
| | CW- | 1483,747 | 108,136 | 1,612 | 0,108 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |
| VF5 | NW- | **2455,187** | 14,300 | **0,091** | 0,036 |
| | CW- | **3000,000** | 0,000 | **0,080** | 0,005 |
| | C- | 3000,000 | 0,000 | 0,000 | 0,000 |

### B. Results

Table III shows the results obtained during 100 iterations. In all of the experiments **C-PSO** delivers the best fitness values (i.e. good convergence) as it behaves like standard PSO. However, the results for energy are the worst among the others as all individuals are run out of batteries by the end of iterations. This is caused by small movements which are made in the swarm after they have found needed global best until they do not really converge to one point, i.e. velocity set to zero. Due to the arising perturbations from vector fields, these small movements take a lot of effort in terms of compensative behavior of C-PSO. Since the maximum total amount of energy is used, i.e. $E^{total} = 3000$, we do not consider C-PSO in further analysis. Considering **NW-PSO** in terms of energy, we observe the best results among the others approaches as expected regardless the structure either of vector field or the search terrain. Considering NW-PSO and the shifted Sphere function, we observe the lowest energy wastes in VF1, VF3, VF4, while in VF2 and VF5 we get the highest ones (i.e. the whole swarm is run out of energy). For the shifted Rosenbrock, the results are a bit different: in VF1,VF4,VF3 the swarm spend almost the same lowest amount of energy which follows by higher expenses (but not the greatest possible) in VF2 and VF5. The order for the Ackley is the same as for Rosenbrock function. The results for **CW-PSO** energy wastes are always higher than for NW-PSO and have a different order: VF3, VF4, VF1, VF2 and VF5, which is the same for Sphere and Ackley with slight difference for Rosenbrock, i.e. in Rosenbrock VF1 is before VF4.

Vector fields VF2 and VF5 are battery expensive for both approaches. This can be justified by high fitness values of both in Table III and success rate in Table IV for all three search terrains. Low energy wastes in VF1, i.e. "Cross", for NW-PSO are caused by its difficult structure (described in [2]). Since particles are trying to follow the flow which blows them away to the corners of the search space, as the result the search process is totally misleading by navigation. So still only by luck the particles, which are initialized at up-right corner, could be pushed by the flow towards the solution, but afterwards again blown away since navigation behavior. That confirms the reasonable good values of success rate in Table IV, but bad fitness values for VF1 in Table III.

Vector field VF4, so called "Bi-directional", is in the middle rank of energy wastes for both NW- and CW-PSO regardless the considered objective functions and indicates reasonable good results as well as for fitness and success rates. The reason consists in the vector fields structure, i.e. in the middle it has a non wind band area symmetrical to which we consider an uniform flow in different directions (to the left side upside the band zone, i.e. towards the solution, and to the right - downside the band). According to this, particles are tend to be pushed to the search borders from which, due to the small random movements, they can escape to non wind area, from where they are already able to correct their course again towards the optimum.

Referring to Table III, for the Ackley it is still very difficult to accomplish the goal as the particles are more often occurred to visit multiple local optima. However, due to the vary continuous redirections, i.e. as chaotic in VF5 and repeating circulation in VF2, the individuals are constantly kicked by the vector field during the navigation process that allows them not to be trapped in bad search areas, i.e. show the trend to converge. The energy costs in this case are in the range of middle values.

TABLE IV: Average values of Success and Survival Rate for NW-PSO, CW-PSO and C-PSO. DARK GRAY indicates high values, i.e. $\geq 50\%$. LIGHT GRAY indicates low values in the range of $[15\%, 50\%)$.

| | | Sphere | | Rosenbrock | | Ackley | |
|---|---|---|---|---|---|---|---|
| | | success | alive | success | alive | success | alive |
| VF1 | NW- | 51% | 95% | 10% | 96% | 27% | 96% |
| | CW- | 1% | 51% | 0% | 44% | 0% | 49% |
| | C- | 99% | 0% | 98% | 0% | 99% | 0% |
| VF2 | NW- | 93% | 94% | 39% | 96% | 49% | 96% |
| | CW- | 48% | 4% | 17% | 12% | 25% | 5% |
| | C- | 99% | 0% | 94% | 0% | 99% | 0% |
| VF3 | NW- | 60% | 96% | 7% | 87% | 18% | 94% |
| | CW- | 1% | 99% | 0% | 97% | 0% | 99% |
| | C- | 99% | 0% | 97% | 0% | 99% | 0% |
| VF4 | NW- | 90% | 100% | 15% | 100% | 25% | 100% |
| | CW- | 19% | 61% | 1% | 36% | 1% | 66% |
| | C- | 99% | 0% | 97% | 0% | 99% | 0% |
| VF5 | NW- | 99% | 52% | 20% | 52% | 75% | 51% |
| | CW- | 100% | 0% | 51% | 0% | 0% | 0% |
| | C- | 99% | 0% | 97% | 0% | 99% | 0% |

The obtained results reveal that proposed model of energy computations is correct, i.e. the individuals arrange their movement relying on the vector field, and that NW-PSO can find valid approximation of the global optimum with low energy wastes relative to the other considered above approaches. Energy costs vary only within vector fields structures and do not depend on the search terrain. Although, in order to get good convergence performance, reasonable expenses are still needed.

### C. Analysis of Survival Rate

In this section we want to investigate how individual loss of particles influences the collective search. Table IV summarizes the average values of **Survival Rate** (denoted as *alive*) and **Success Rate** (denoted as *success*) in percentage for each algorithm by the end of iterations among 100 runs. From this table it can be seen that in NW-PSO, as expected, a very small amount of particles run out of energy independent on either the search landscape or the vector field. For NW-PSO we observe high *alive* values, i.e. almost more than $95\%$. The exception is only vector field VF5 (so-called "Random"), where about the half of population is recharged by the end of the search. Nevertheless, the success rate in this vector field is the highest among the others. Especially for the shifted Ackley function in VF5 we observe $75\%$ of successful runs, which is the best obtained value along with survival of $51\%$ among the considered vector fields.

While the Survival Rate does not vary much among vector fields (as the main goal of NW-PSO is to save energy), the Success Rate differs between both the vector fields structures and objective functions. For Sphere by **NW-PSO** we always obtain high values of success despite the vector field, while results for Rosenbrock are very low. The reason is that while trapping into the area of flat plateau during the navigation process, the particles in that region mostly go directly with the corresponding flow without almost any corrections for the search. So only those individuals, which by occasion follow profitable for search directions, pass through the optimum. The results of Success Rate for Ackley are also low, but higher than for Rosenbrock. This is because NW-PSO particles never stop their movement (i.e. do not run off energy), so they are always blown away from the local optima no matter as frequent they are trapped by them. The results for **CW-PSO** present very few values of Success Rate suitable for consideration, while those which indicate at least low values go along with almost 0% Survival Rate. At the same time 0% Survival Rate of **C-PSO** by the end of iterations with almost 100% Success Rate reveals that the particles accomplish the task already before the first loss is occurred. From this we deduce, that the premature loss of certain individuals destroys essential for the search connections between the others (like the same problem as re-initialization of the population after dynamic change [3]). The another issue is that CW-PSO behavior highly depends on the *Information Map*, while NW-PSO could deal well in unknown environment as it navigates mostly according to the current position.

## V. Conclusion and Future Work

In this paper we presented Navigation Wind PSO (NW-PSO) method as a search mechanism under energy constraints for aerial swarms acting in environments with unknown perturbations modeled by vector fields. The main idea of the paper is motivated by a real case scenario of an aerial micro-robotic swarm with limited battery capacity. The proposed model and the corresponding features are meant to provide an algorithmic design to estimate effects of the external unknown dynamics on energy consumption of the aerial swarm during the search process. The presented experiments were made to provide a baseline for further realistic tests. The analysis reveal that the proposed model of energy computations is correct and according to it the individuals of NW-PSO can arrange their movements relying on the vector field and thereby save the energy. Provided that NW-PSO always outperforms CW-PSO in energy conservation along with optimization of the fitness landscape, i.e. it was justified that in most of the cases NW-PSO can also find good approximation of the global optimal solution which is better than this one found by CW-PSO. Moreover, NW-PSO prevents the premature particle loss, which as was shown has a very large impact on the mechanism of search. According to the obtained results, it is very sufficient to find a balance between energy conservation and search mechanism which depends on either the search landscape or vector field structure. Further research will be needed in order to determine how the individuals have to switch between the behavioral strategies in order to accomplish the task as efficient as possible under the corresponding conditions. This will provide the basis for derivation of profitable for search vector fields structures under unknown conditions. As the part of the future work, also deeper statistical analysis of the difference in performance between particles behavior will be held.

## References


[1] W. H. Al-Sabban, L. F. Gonzalez, R. N. Smith, and G. F. Wyeth. Wind-energy based path planning for electric unmanned aerial vehicles using markov decision processes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, February 2012.

[2] P. Bartashevich, L. Grimaldi, and S. Mostaghim. Pso-based search mechanism in dynamic environments: Swarms in vector fields. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, June 2017.

[3] J. Branke. *Evolutionary Optimization in Dynamic Environments*. Genetic Algorithms and Evolutionary Computation. Springer US, 2012.

[4] J. W. Chapman, R. H. G. Klaassen, V. A. Drake, S. Fossette, G. C. Hays, J. D. Metcalfe, A. M. Reynolds, D. R. Reynolds, and T. Alerstam. Animal orientation strategies for movement in flows. *Current Biology*, 21(20):861–870, 2011.

[5] C. Di Franco and G. C. Buttazzo. Energy-aware coverage path planning of uavs. In *Autonomous Robot Systems and Competitions (ICARSC 2015), International Conference on*, Villa Real, Portugal, 2015. IEEE.

[6] D. Jones and G. A. Hollinger. Real-time stochastic optimization for energy-efficient trajectories.

[7] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[8] S. Kernbach. *Handbook of collective robotics - fundamentals and challenges*. 2013.

[9] O. Kim, I.Y.and de Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multi-disciplinary Optimization*, 29(2):149–158, Feb 2005.

[10] D. Kularatne, S. Bhattacharya, and M. A. Hsieh. Time and Energy Optimal Path Planning in General Flows. *Rss*, 2016.

[11] T. Luukkonen. Modelling and control of quadcopter. *Independent research project in applied mathematics, Espoo*, 2011.

[12] J. McLaren. *Optimal bird migration: accumulating fuel, negotiating wind and trade-offs in fitness*. PhD thesis, 2017.

[13] J. D. McLaren, J. Shamoun-Baranes, and W. Bouten. Wind selectivity and partial compensation for wind drift among nocturnally migrating passerines. *Behavioral Ecology*, 2012.

[14] J. D. Mclaren, J. Shamoun-Baranes, A. M. Dokter, R. H. G. Klaassen, and W. Bouten. Optimal orientation in flows: providing a benchmark for animal movement strategies.

[15] S. Mostaghim, C. Steup, and F. Witt. Energy aware particle swarm optimization as search mechanism for aerial micro-robots. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7, Dec 2016.

[16] J. F. Roberts, J.-C. Zufferey, and D. Floreano. Energy management for indoor hovering robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1242–1247. IEEE, 2008.

[17] T. Stirling, S. Wischmann, and D. Floreano. Energy-efficient indoor search by swarms of simulated flying robots without global information. *Swarm Intelligence*, 4(2):117–143, 2010.

[18] C.-M. Tseng, C.-K. Chau, K. M. Elbassioni, and M. Khonji. Flight tour planning with recharging optimization for battery-operated autonomous drones. *CoRR*, abs/1703.10049, 2017.

[19] J. A. Vrugt, J. Van Belle, and W. Bouten. Pareto front analysis of flight time and energy use in long-distance bird migration. *Journal of Avian Biology*, 38(4):432–442, 2007.

[20] S. Waharte and N. Trigoni. Supporting search and rescue operations with uavs. In *International Symposiom on Robots and Security*, 2010.

[21] S. Waharte, N. Trigoni, and S. J. Julier. Coordinated search with a swarm of uavs. In *Sixth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2009.

[22] T. Wang, O. P. L. Maitre, H. Ibrahim, and O. M. Knio. Path planning in uncertain flow fields using ensemble method. *Ocean Dynamics*, 66(10), 2016.

[23] T. Wang, B. Wang, H. Wei, Y. Cao, M. Wang, and Z. Shao. Staying-alive and energy efficient path planning for mobile robots. In *Proceedings of the American Control Conference*, pages 868–873. IEEE Press, 2008.

[24] C. Yoo, R. Fitch, and S. Sukkarieh. Online task planning and control for fuel-constrained aerial robots in wind fields. *The International Journal of Robotics Research*, 35(5):438–453, 2015.

[25] D. Zorbas, T. Razafindralambo, D. P. P. Luigi, and F. Guerriero. Energy efficient mobile target tracking using flying drones. *Procedia Computer Science*, 19(Ant):80–87, 2013.