Energy Aware Particle Swarm Optimization as Search Mechanism for Aerial Micro-robots

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

Christoph Steup
Faculty of Computer Science
University of Magdeburg
Germany
Email: steup@ovgu.de

Fabian Witt
Faculty of Computer Science
University of Magdeburg
Germany
Email: fabian.witt@st.ovgu.de

***Abstract*—This paper presents the Energy Aware PSO (EAPSO) as a search mechanism for aerial micro-robots with limited energy capacity. The proposed model is an extension of the search concept of Particle Swarm Optimization (PSO) that additionally considers the energy levels of the individuals for an efficient movement. One major contribution of this paper is that the energy efficiency results from a multi-criteria decision making process performed by the individuals. The energy consumption model in EAPSO is adapted from a real hardware scenario and has been tested on three known landscapes which are very similar to search terrains by the aerial micro-robots. The results show that EAPSO can reduce the total energy consumption of the swarm with negligible degradation of the search results.**

I. Introduction

Swarm Robotics (SR) has been the subject to research for almost over a decade. The major properties concern a large number of simple robotic systems and simple rules. The swarm is supposed to collectively learn a pre-defined given task [11]. One important challenge in micro-robotic systems is managing the energy resources, which is a crucial aspect in accomplishing a task by such autonomous systems [5]. This is notably evident with small aerial robots, which have severely limited battery of typically 10 to 15 minutes [7], [12]. Different to ground robots, aerial robots have significantly different energy dynamics, require substantially more energy to locomote [7], [9], and the small payload entails reduced sensing and processing capabilities. Dealing with limited energy levels has been addressed in the literature, for instance, an algorithm for indoor aerial swarm search that exploits the ability of flying robots to attach to ceilings and saves energy was developed by [7] and [10]. A novel strategy was studied by [9] which controls the density of flying robots. They illustrate an efficient way of reducing swarm energy costs while maintaining a rapid search. Other approaches consider Underwater robots [1] saving energy by staying on the surface. An additional aspect in swarm robotics is controlling deployment into unknown environments. If robots deploy to unnecessary locations, energy is wasted. Conversely, if an area receives insufficient robots the task may be unachievable or the performance is reduced. Nevertheless, a crucial feature is the autonomy of robotic systems, i.e., the individual robots need to be able to make decisions on their own by considering own and the performance of the other swarm members.

The goal of this paper is to propose a new model for a search mechanism in a swarm of small flying robots (aerial micro-robots) which is based on Particle Swarm Optimization method. The main research questions are to investigate whether PSO can be employed as a search mechanism for aerial micro-robots and how can we consider an efficient energy consumption for such systems with limited battery capacity. In this model we aim to additionally involve the amount of energy in the search mechanism and give the single individuals the ability to decide about their movements considering their energy levels. The decision making process of each single individual must additionally involve information from other swarm members and the overall progress of the swarm. We propose to use the concepts from multi-objective criteria decision making and let the individuals decide about their individual movements. To our knowledge this concept has not been studied so far in the literature about the search mechanisms for aerial autonomous swarm robots.

Decision making in the swarms has been studied in other contexts presented by [13], [14] in which the authors investigate different robot controllers for collective decision-making in the context of the speed-versus-accuracy trade-off. The proposed approach in this paper is different from collective decision making in swarms as we let the swarm members decide on their own by considering the local information.

The proposed method introduces a new PSO-based search mechanism called Energy Aware PSO (EAPSO) which additionally considers the amount of energy consumption for each individual. In contrast to the standard PSO, the individuals estimate the amount of required energy for moving to the next position, and decide about the movement by considering the trade-off between profit and energy consumption. The paper presents various models for multi-criteria decision making and the experiments and comparisons illustrate that with EAPSO the individuals are able to save a large amount of energy without degrading the quality of search. EAPSO is modeled based on energy consumption of a real hardware scenario. However, the presented work in this paper only considers the search mechanism and has met several assumptions.

This paper is structured as follows. We describe the background in Section II. Afterwards, we introduce our model in Section III. The leader selection mechanism and multi-criteria decision making are studied in Section IV. Subsequently, Section V contains the experiments and the evaluations. The paper is summarized in Section VI.

II. Background

In this section, we briefly describe the background about Particle Swarm Optimization and Multi-Criteria Decision Making.

A. Particle Swarm Optimization (PSO)

PSO is a search mechanism based on the movement of a population of N individuals in a n-dimensional search space defined by S [3]. Each individual i has a position $\vec{x}_i(t) \in S$ and a velocity $\vec{v}_i(t)$ at time step t. The individuals move in the search space by considering three factors: their own velocity vector at $t-1$, their own best obtained position $\vec{P}_{best}$ and the position of the globally (or locally) best individual from the population $\vec{x}_g$:

$$\vec{v}_i(t+1) = w\vec{v}_i(t) + C_1\phi_1(\vec{P}_{best} - \vec{x}_i(t)) + \quad (1)$$
$$+ C_2\phi_2(\vec{x}_g - \vec{x}_i(t))$$
$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \quad (2)$$

where ϕ_1 and ϕ_2 are two random vectors $\in [0,1]^n$. C_1 and C_2 are constants and determine the attraction rates. The globally best individual $\vec{x}_g$ can be defined using different communication topologies which is known to have a large impact on the convergence rate of PSO [2]. As selecting the globally best solution (leader) defines the amount of distance an individual might fly (depending on the random value ϕ_1), the topology can implicitly influence the energy consumption of the individuals. Ideally, the closer the globally best solution is located, the less energy is required to reach that point. In the following we take the *k-Nearest-Neighbor* neighborhood topology, where $k = N-1$ refers to the fully connected network. In the experiments, the size k will be analyzed.

B. Multi-criteria Decision Making

Multi-criteria decision making methods usually involve several conflicting objective functions $f_i(\vec{x})$ for $i = 1, \cdots, m$ and $\vec{x} \in S$ which have to be optimized at the same time. The solution of multi-objective optimization problems is usually a set of so called Pareto-optimal solutions from which the user has to select one. 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. Selecting one of the Pareto-optimal (or non-dominated) solutions depends on the preferences of the user and can vary accordingly [4], [6]. Weighted sum approach is known to be the most straight forward mechanism to incorporate the preferences of the user in a weight vector $(w_1, \cdots, w_m)$ where $\sum_{i=1}^{m} w_i = 1$. Each w_i indicates the relative preference towards the objective function i. Different vector values lead to different selection preferences.

III. Proposed model

Our proposed approach models a population of N autonomous aerial robots which move (fly) in a search space $S \subset R^2$. The search mechanism involves the concept of PSO. Nevertheless, in order to employ PSO as a search mechanism for aerial micro-robots, we need to extend the PSO by a new energy model and additionally address the discrete time movements (as shown in Equations (1) and (2)).

A. Modeling the movement

In order to better capture the main concept, we model the flight behavior of the individuals using a simple finite state machine, as shown in Figure 1.

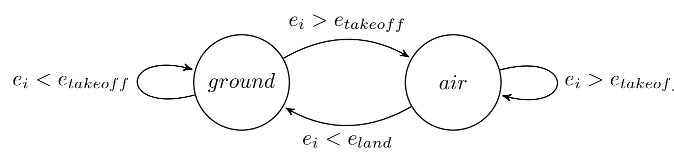

Fig. 1. Transition graph depicting the flight model for an individual i depending on its energy level e_i

The individuals operate in two states: *Ground* or *Air*. In the Ground state the individuals can decide between two different actions. If an individual i has enough energy e_i (i.e., $e_i > e_{takeoff}$), it can start and switch to the Air state. In case of low energy level ($e_i <= e_{takeoff}$), the individual stays in the Ground state. For simplicity, we assume that staying in Ground state is coupled with charging. This is a very strong assumption which cannot be easily implemented in real-scenarios.

In the Air state the individuals either change state and go to Ground state if $e_i < e_{land}$ or they move (fly) towards a certain direction. In fact the direction towards which the individuals move has a great impact on the energy consumption of the individuals. In this case, the individuals need to decide on the distance they will move in the next iteration.

B. Discrete time flight

Even though the real-world is continuous in time, robotic systems work in a time-discrete way. They follow a strict input, compute, output scheme with a fixed time interval. In the case of the real aerial micro-robots (quadcopters) different functional blocks have different time intervals. Our reference copters [8] for example stabilize their attitude approximately 100 times per second, but they only decide about the next movement command approximately 10 times per second. Consequently, our simulation follows the time-discrete approach and models the micro-robots to only take decisions in certain time intervals. Each time interval is a snapshot of the system on which the next decisions will be taken. To generalize the simulation we abstracted away the time interval and simply count the amount of decisions taken, which also yields a monotonic increasing time measurement. This allows for an easier comparison with PSO Equations (1) and (2), since the equations are measured regarding the number of iterations passed. Therefore we call our time interval count iterations.

C. Energy Computations

In the following we introduce a simplified model of energy consumption for the individuals which is modeled based on a real scenario from [8]. The energy of an individual is defined in units per iteration. In each iteration each individual can use at most one energy unit. Based on this energy unit several constants are defined, which represent the energy consumption of different actions the aerial robot can execute, such as: take off, hover, control, move and land. Each constant is defined as a fraction of an energy unit. The resulting constants are $e_{takeoff}$, e_{hover}, $e_{control}$, e_{move} and e_{land}.

$e_{takeoff}$ and e_{land} require a relatively fixed amount of energy and can be calculated by considering the required energy related to the potential energy:

$$E_{pot} = m \cdot g \cdot h \tag{3}$$

where the mass (m), the gravitation (g) and the target height h are considered to be constant as we let the aerial robots move at the same height (either the target height while flying or zero when landed). Therefore, we can conclude that $e_{takeoff} = E_{pot}$ and $e_{land} = -E_{pot}$.

$e_{control}$ indicates the amount of energy which is required for computations and communication and can be estimated to be a constant value per iteration for computation.

The main sources of energy consumption which can be additionally influenced by the individual itself, are the e_{hover} and e_{move}. The more time an individual spends in the hovering state, it will consume more energy. This also holds for the flight distance and its corresponding energy consumption denoted by e_{move}.

In order to keep the model as simple as possible, we have met the following assumptions. In our model, we set a maximum distance that each individual can move in one iteration. All robots fly at the same height, have identical weight and move in constant time steps. To calculate a simplified movement cost, the flight has an acceleration and a deceleration phase. With this assumptions the cost for *move* depends on the flight distance which is calculated using the Euclidean distance values. The individuals can recharge their batteries while in Ground state. The charging rate is defined in percentage of the energy unit.

D. Energy Aware PSO

Algorithm 1 illustrates the main building blocks of our proposed approach called Energy Aware PSO. In this algorithm, we consider the flight physics in the PSO computations. The goal is to define a population of individuals, which search for an optimal solution in a defined search space. The individuals have physical constraints in terms of their energy values. In this algorithm, the individuals can decide about the amount of their movement in the search space based on both their current energy level and the performance of the other individuals in their neighborhood. The algorithm starts with initialization

Algorithm 1: Energy-aware PSO

```
Input : N Individuals
t = 0
Initialize the individuals
for i = 1 to N do
    x⃗_i(t) = StartPosition(S)
    v⃗_i(t) = 0
    e_i(t) = Random(e_min, e_max)
end
while Stopping Criterion not fulfilled do
    for i = 1 to N do
        state_i(t) = DecideState(x⃗_i(t), e_i(t))
        if state_i(t) == Ground then
            charge: e_i(t) = e_i(t) + c
        end
        else
            x⃗_g(t) = LeaderSelection(state_i(t), e_i(t))
            v⃗_i(t+1) = ComputeVelocity (x⃗_i(t), x⃗_g(t))
            x⃗_i(t+1) = UpdatePosition(v⃗_i(t), x⃗_i(t))
            e_i(t+1) = ComputeEnergy(v⃗_i(t+1))
        end
    end
    t = t + 1
end
```

of the individuals at $t = 0$ using certain start positions $\vec{x_i(t)}$, initial velocity values $\vec{v}_i(t)$ and certain energy level $e_i(t)$ for individual i.

After the initialization, the individuals make a decision about their functionality and select a state either as Air or Ground. The function *DecideState* considers the amount of the energy $e_i(t)$ available to individual i and in case there is a certain minimum value of e_{min}, the individual takes off. Otherwise, the individual's state ($state_i(t)$) remains in Ground and gets charged. Here we take a simple additive recharging mechanism with a constant value c.

In the case that the individual takes off, it needs to find the globally best individual (denoted at the *LeaderSelection*) to be able to perform the PSO movement. However, this depends on the amount of available energy $e_i(t)$. The individuals with enough energy values can perform as in the standard PSO, while the others with low energy values can only perform a local search. This decision has a large impact on the convergence of the approach and will be studied in Section IV. The next steps after the leader selection mechanism are straight forward. Each individual computes its velocity vector and moves accordingly using the PSO equations. In addition to this, if the calculated velocity is less than a certain minimum threshold, e.g. if the individual is stuck in a local optima, we use the so called turbulence factor: those individuals take a randomly generated velocity vector. Additionally, we assign a maximal velocity value V_{max} as a threshold and in case the velocity vector is larger than V_{max}, it will set to V_{max}. After each movement, the individuals compute their energy consumption in *ComputeEnergy*. This process is performed iteratively until a stopping criterion is fulfilled. We set the maximum number of iterations (time) as the stopping criterion.

IV. Leader Selection and Multi-Criteria Decision Making

Leader selection mechanism (cf. Algorithm 1) contains the multi-criteria decision making process for each individual in the population. An individual i must select a leader according to several factors such as its energy level e_i, the amount of overall work to be done and the status of the other individuals in the neighborhood. The main steps for selecting the leaders are shown in Algorithm 2.

Algorithm 2: Leader Selection for individual i

```
Input : state_i(t) and e_i(t)
Output : Globally best position x_g(t)
if state_i(t) = Air then
    for k = 1 to N-1 do
        x_g^k(t) = FindBest(k)
        Cost(k) := ComputeCost(x_g^k(t))
        Profit(k) := ComputeProfit(x_g^k(t))
    end
    x_g(t) = MCDM(Profit, Cost, e_i(t))
end
```

The individual i goes through the multi-criteria decision making for leader selection mechanism only when it is in the Air state. The first step is to find the globally best solutions for several neighborhood topologies with $k = 1, \cdots, N-1$ using the k-Nearest-Neighbor. In this case we can have $N-1$ different possible globally best solutions: $x_g^1(t)$ to $x_g^{N-1}(t)$. The individual computes (simulates) cf. IV-A its next position $\vec{x}_i(t+1)$ by considering each of the possible $N-1$ globally best solutions. In order to select one of them, it computes the *cost* and *profit* in terms of energy consumption for each of the possible next positions. Cost simply captures the amount of required energy to reach $\vec{x}_i(t+1)$.

Profit means the difference between the quality of the current position $f(\vec{x}_i(t))$ and the next one $f(\vec{x}_i(t+1))$. This value must be approximated as the PSO equations involve several random values such as ϕ_1 and ϕ_2 and the quality of the position for $t+1$ is not known. Section IV-A describes the approximation.

The values related to cost and profit are in conflict with each other; the solutions with high profit can cause a large energy consumption. In this case, the individual must select one of the $N-1$, $x_g^k(t)$, $k = 1, \cdots, N-1$, using concepts from multi-criteria decision making (denoted as MCDM (profit, cost, $e_i(t)$) in the Algorithm 2).

In this paper we take the weighted sum approach from Section II. Each individual i is assigned a weight vector for the two criteria *profit* and *cost*: $w_i = (r_i, 1-r_i)$, where r_i indicates the amount of *risk* in terms of energy consumption an individual would spend to achieve a large profit. For instance, $w_i = (1, 0)$ depicts the preference to select new possible position which delivers a large amount of profit and requires a large amount of energy. The values for r_i can be selected using different mechanisms:

1) Randomly: Each of the individuals in the population has a random preference.
2) Constant: All the individuals have the same value such as (0.5, 0.5), (1, 0) and (0, 1).
3) Adaptive: The individuals select their preferences based on the amount of available energy.

After setting the preferences for the individuals, each individual ranks each of the possible $N-1$ new positions at $t+1$ according to its weight vector as follows:

$$Rank(k) \quad = \quad r_i \cdot profit(k) + (1-r_i) \cdot cost(k)$$

Where $k = 1, \cdots, N-1$. The position with the lowest rank will be selected as the $\vec{x}_g(t)$ by the the individual.

The above multi-criteria decision making approach for each single individual implicitly implies that the individuals with low values of risk (e.g., $r_i = 0$) will perform small movements in the search space and hence a local search. On the other hand the individuals with $r_i = 1$ select the leaders which are far away from them and require a large amount of energy. Considering the amount of profit in the decision making process implicitly involve the status of the other individuals in the neighborhood. If all of the individuals in a neighborhood have more or less the same function value, the amount of profit will degrade.

A. Profit approximation

In this section we describe the approximation of the profit by the individuals. As described in the last section, each individual simulates the next $N-1$ possible steps in order to make a decision. The output of this process is a set of parameters. This set contains the next state of the individual, the action performed by the individual, the cost for moving, the velocity vector and the profit of the movement. Since we deal with an unknown environment, the function value of a none visited position is not known. Therefore, the function value of the possible next positions must be approximated. Here, we use the information given by the neighborhood around each individual who is able to access the previous visited points and corresponding function values of all individuals in its neighborhood. With this information the individual is able to approximate the unknown landscape and can calculate the function value of the simulated goal. For the approximation, we use ordinary least-squares regression to fit a quadratic model with constant, linear, interaction, and squared terms. In order to save memory and improve the approximation, each individual collects points with distance greater than a certain threshold (here 0.1) in the neighborhood.

V. Experiments

The main idea of this paper is motivated by a real case scenario of an aerial micro-robotic swarm. The proposed model and the corresponding features are meant to provide an algorithmic design for the energy consumption and search of the aerial swarm. Therefore the goal of the experiments is to provide a baseline for further realistic tests. The parameters are selected based on a model of the FINken-III micro aerial robot [8] as follows: $e_{takeoff} = 10$, $e_{hover} = 28$, $e_{control} = 20$, $e_{move} = 22$ and $e_{land} = -10$.

The flight of the aerial robots are modeled in a $n = 2$ dimensional search space. The goal is to analyze if a swarm with limited energy can find an optimal solution in the search space. We take the standard PSO [3] as the baseline algorithm and denote it as *default PSO*. As the search space only contains two parameters, the default PSO is easily able to solve the problem. We take $C_1 = C_2 = 1$, $w = 0.5$ and population size of 30.

Three test problems such as Sphere, Ackley and Rosenbrock from the literature are being used for the experiments. These test problems can very well simulate the terrain in which aerial swarms can fly and search, while Sphere is only for simple tests, Ackely and Rosenbrock respectively capture search terrains with lots of local optima and a flat plateau. The main focus of our experiments is on the different multi-criteria decision making approaches and the total energy consumption. All the experiments are run for 30 times and the median values and the corresponding standard errors are reported.

The experimental area (arena) is defined as $x_1, x_2 \in [-10, 10]$. All individuals start in a defined area with $x_1 \in [-10, 10]$ and $x_2 \in [-10, -8]$ at a random position (this is a realistic assumption for aerial robotic systems). the initial state is set to be the Ground state. The initial velocity is set to $v_0 = 0$. The optimization stops after 500 iterations.

In all the experiments the best function values (denoted as *fitness*), the total amount of movement in the swarm (denoted as *distance*) and the total amount of available energy in the swarm (denoted as *energy*) are measured. The total amount of movement is meant to capture the amount of energy consumption, while the total amount of energy can be used to estimate the charging behavior and its frequency during the 500 iterations. In the experiments, we compare default PSO with EAPSO with different values for r_i as 1.0, 0.5 and 0.0. Additionally, we perform a random leader selection mechanism as a baseline for a random decision making and the adaptive variant denoted as *adaptive* in which the individuals select a leader according to their available energy level $e_i(t)$.

A. Results

Table I shows the results after 500 iterations. In all of the experiments the $r_i = 0$ delivers the worst fitness value as expected. In this case, the individuals always select the closest better individual as their global best and therefore they save lots of energy and distance. Considering the Sphere function, we observe that all of the other EAPSO variants can find the optimal solution where the particles in default PSO move the

TABLE I

Results for the three test problems (median values and standard errors (std)). "fitness" refers to the best function value obtained by the swarm, "energy" and "distance" indicate the total amount of energy and the distance moved by the swarm

risk r_i	fitness	± std	energy	± std	distance	± std
			Sphere			
default	0.000	0.000	4924.679	11.950	2274.928	38.020
0.00	5.598	0.485	4375.483	1.499	415.959	5.337
0.50	0.000	0.000	4430.815	1.157	729.013	2.684
1.00	0.000	0.000	4449.644	1.203	799.311	3.406
random	0.000	0.000	4425.916	2.541	684.370	8.266
adaptive	0.000	0.000	4404.301	1.315	568.176	6.851
			Ackeley			
default	0.001	0.120	4942.563	4.235	2317.692	13.242
0.00	11.61	0.162	4505.048	3.383	935.639	9.089
0.50	0.000	0.554	4531.115	8.800	1008.798	22.184
1.00	0.001	0.171	4993.991	6.649	2491.168	25.109
random	0.001	0.203	4773.944	12.115	1766.415	37.688
adaptive	0.002	0.852	4449.128	5.755	775.828	19.086
			Rosenbrock			
default	0.000	0.000	4929.129	9.580	2291.180	31.136
0.00	1095	76.57	4391.300	2.108	491.280	8.549
0.50	1.483	68.10	4432.276	7.299	714.238	28.774
1.00	0.102	0.296	4899.941	17.155	2179.118	58.146
random	0.018	0.174	4549.582	12.328	1029.446	41.034
adaptive	0.067	0.014	4387.826	0.774	433.425	2.552

largest distance and consume more energy than the others. Among the EAPSO approaches, the adaptive method saves the most amount of energy, while $r_i = 1.0$ has the highest energy consumption. Considering the Ackeley function with lots of local optima, the default PSO obtains the best result in term of the fitness value (and the corresponding std. error values). The adaptive variant is not as good as the other variants, nevertheless its distance and the energy values are the best among the others. The same results can be observed in Rosenbrock function with a large flat plateau. Due to the small amount of profit which can be obtained in a local neighborhood, the individuals reduce the amount of movement and save energy while not making the effort of moving. This leads to a degradation in the function value. In order to better analyze the results, we investigate the convergence plots, the energy and distance measures over the iterations. Figures 2 to 4 illustrate the fitness, distance and energy values over 500 iterations. For better analysis and comparison of the of the results we only show the results of default PSO and adaptive. As expected the standard PSO is able to converge much faster than the EAPSO variants for all the three test problems.

The results get very interesting once we look at the distance plots i.e., the total amount of movement by the swarm, as shown in Figure 3. We observe that the movement of the individuals in default PSO never stops, even if the PSO has already obtained the optimal solution. In contrast to this, in the adaptive EAPSO the individuals reduce their movements (and therefore the energy consumption) to a large extent. Both

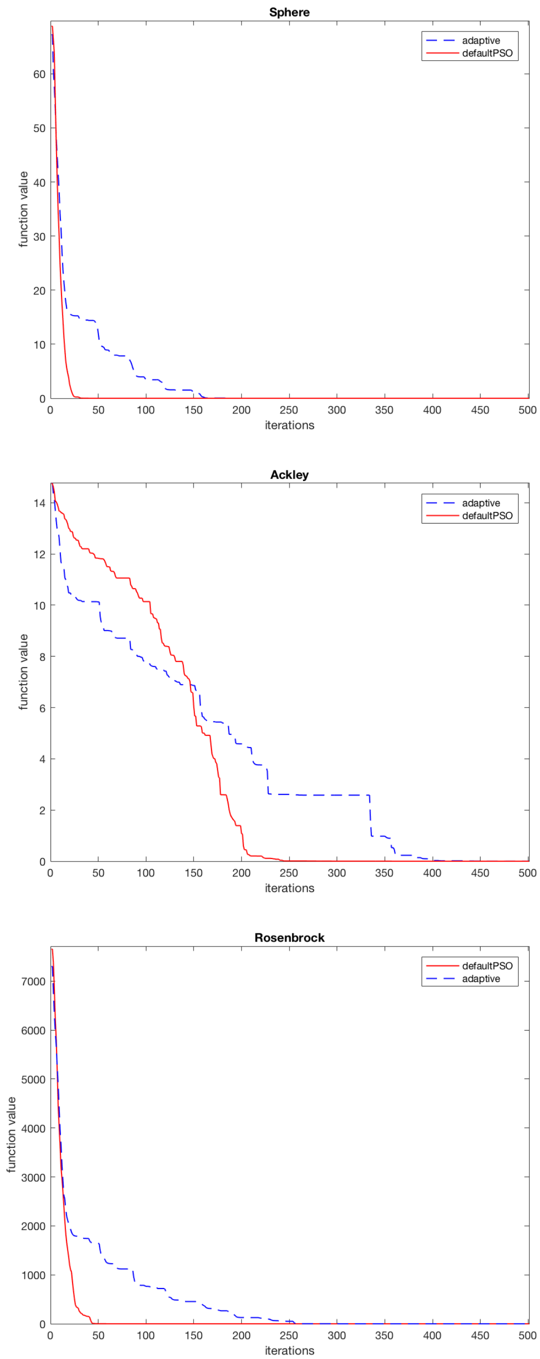

Fig. 2. Convergence plots (Average of the best function value over iterations)

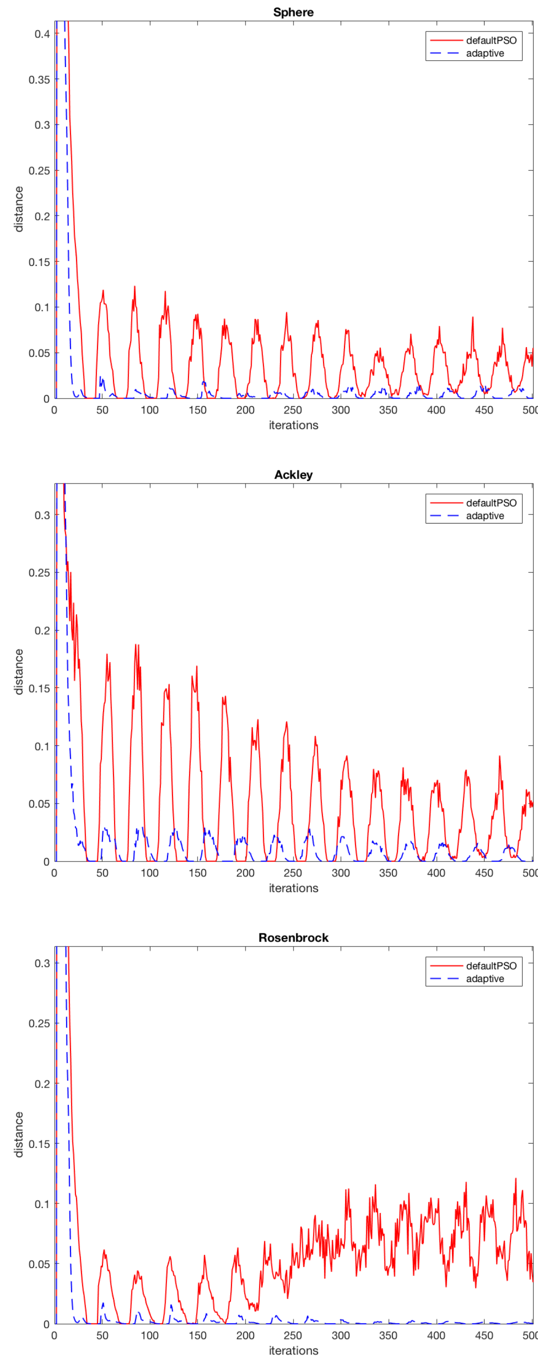

Fig. 3. Average movement of the entire swarm over iterations

of the approaches have a cyclic behavior in the distances due to the recharging effect in the model. Since the individuals all start with a certain high battery level, many of them require a recharging at the same time steps.

Figure 4 illustrates the total amount of available energy in the entire swarm. We observe that both of the methods have a cyclic energy level. Starting with a large amount of energy, the individuals get synchronized over the iterations, i.e., they all recharge at the same time steps (at the lower peaks). This effect is more visible in EAPSO than the default PSO. This interesting side effect can be explained by the fact that in the adaptive EAPSO the individuals only move if the trade-off between profit and cost is large enough. In case of Ackley function the synchronized behavior is less visible than the other two test problems as the individuals can be trapped in several local optima and build clusters with different properties. By careful observations, we can conclude that the

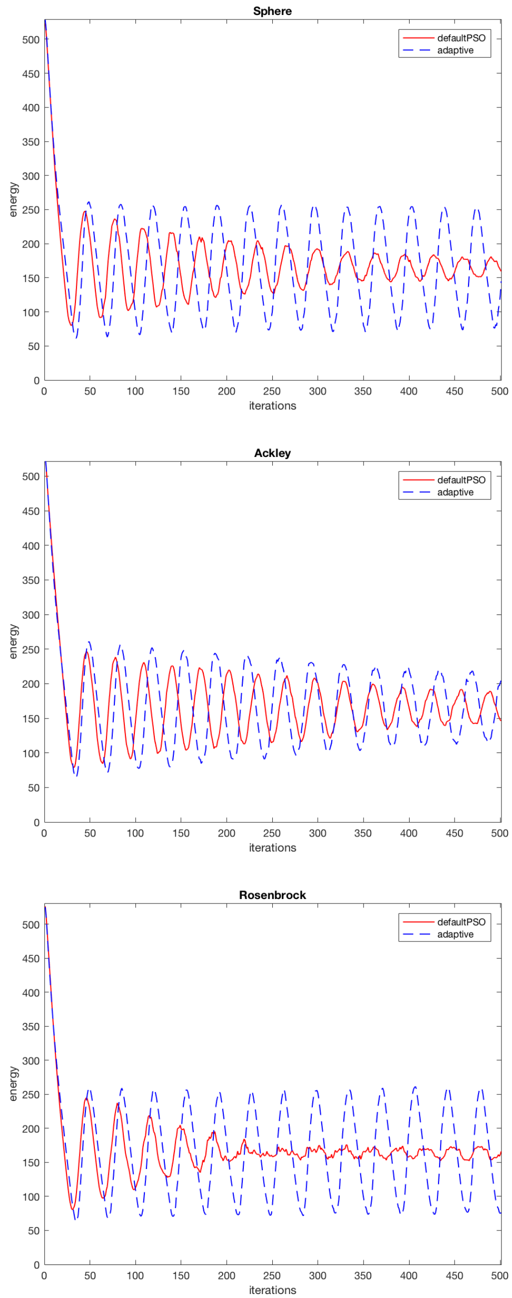

Fig. 4. Average total amount of energy in the swarm over iterations.

EAPSO individuals require less re-charging cycles than the individuals in default PSO.

VI. Conclusion and Future Work

In this paper we presented EAPSO (Energy Aware PSO) method as a search mechanism for swarm of aerial micro-robots. The paper is the first attempt which contains several assumptions. Nevertheless we have captured several important features from a real hardware scenario. In particular we have addressed the energy consumption and the discrete time flight (from PSO) in the search mechanism. The proposed model is built upon the default PSO with an additional multi-criteria decision making aspect for the individuals which make a decision before starting a movement. The decision is made based on two objectives, profit in terms of the overall gain in search process and cost in terms of the energy consumption. We have used weighted sum approach and an adaptive version for the decision making. The experiments on three test problems show that PSO can be used as a search mechanism for swarm of aerial microrobots and integrating the decision making process in the optimization can extensively reduce the energy consumption, while the quality of search will be influenced. This work has opened a large number of research questions for future work. The next step is to work on the assumptions and implement the search mechanism on Hardware platform (FINken-III). Addressing several tasks in the swarm, adaptation of the parameters and analyzing the fitness landscapes are among the next steps.

References

[1] A. Amory, T. Tosik, and E. Maehle. A load balancing behavior for underwater robot swarms to increase mission time and fault tolerance. In *Parallel Distributed Processing Symposium Workshops (IPDPSW)*, pages 1306–1313, 2014.

[2] A. P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. Wiley, 2005.

[3] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann, 2001.

[4] K. Miettinen. *Non-linear Multiobjective Optimization*. Kluwer, International Series in Operations Research and Management Science, 1999.

[5] K. O'Hara, R. Nathuji, H. Raj, K. Schwan, and T. Balch. Autopower: toward energy-aware software systems for distributed mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2757–2762, 2006.

[6] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang. A review of hybrid evolutionary multiple criteria decision making methods. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1147–1154, 2014.

[7] J. Roberts, J.-C. Zufferey, and D. Floreano. Energy management for indoor hovering robots. In *Intelligent Robots and Systems (IROS)*, pages 1242–1247, 2008.

[8] C. Steup, S. Mai, and S. Mostaghim. Evaluation platform for micro aerial indoor swarm robotics. In *13th International Symposium on Distributed Autonomous Robotic Systems*, page submitted, 2016.

[9] T. Stirling and D. Floreano. Energy-time efficiency in aerial swarm deployment. In A. e. a. Martinoli, editor, *Distributed Autonomous Robotic Systems*, volume 83 of *Springer Tracts in Advanced Robotics*, pages 5–18. Springer Berlin Heidelberg, 2013.

[10] 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.

[11] V. Trianni. *Evolutionary Swarm Robotics: Evolving Self-Organising Behaviours in Groups of Autonomous Robots (Studies in Computational Intelligence) (Studies in Computational Intelligence)*. Springer Publishing Company, Incorporated, 1 edition, 2008.

[12] M. Valenti, B. Bethke, J. How, D. de Farias, and J. Vian. Embedding health management into mission tasking for uav teams. In *American Control Conference*, pages 5777–5783, 2007.

[13] G. Valentini, H. Hamann, and M. Dorigo. Efficient decision-making in a self-organizing robot swarm: On the speed versus accuracy trade-off. In R. B. et al., editor, *Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1305–1314, 2015.

[14] M. Wahby, A. Weinhold, and H. Hamann. Revisiting BEECLUST: Aggregation of swarm robots with adaptiveness to different light settings. In *Int. Conf. on Bio-inspired Information and Communications Technologies*, 2015.