International Journal on Artificial Intelligence Tools
Vol. 25, No. 1 (2016) 1660003 (19 pages)

DOI: 10.1142/S0218213016600034

Fine-Tuning of UAV Control Rules for Spraying Pesticides on Crop Fields: An Approach for Dynamic Environments

Bruno S. Faiçal*, Gustavo Pessin†, Geraldo P. R. Filho*,
André C. P. L. F. Carvalho*, Pedro H. Gomes‡ and Jó Ueyama*

*University of São Paulo (USP), São Carlos, SP, Brazil
†Applied Computing Lab, Vale Institute of Technology, Belém, PA, Brazil
‡Autonomous Network Research Group (ANRG)
University of Southern California, Los Angeles, CA, USA
{bsfaical, geraldop, andre, joueyama}@icmc.usp.br
†gustavo.pessin@itv.org
‡pdasilva@usc.edu

Received 27 April 2015
Accepted 21 December 2015
Published 23 February 2016

Brazil is an agricultural nation whose process of spraying pesticides is mainly carried out by using aircrafts. However, the use of aircrafts with on-board pilots has often resulted in chemicals being sprayed outside the intended areas. The precision required for spraying on crop fields is often impaired by external factors, like changes in wind speed and direction. To address this problem, ensuring that the pesticides are sprayed accurately, this paper proposes the use of artificial neural networks (ANN) on programmable UAVs. For such, the UAV is programmed to spray chemicals on the target crop field considering dynamic context. To control the UAV flight route planning, we investigated several optimization techniques including Particle Swarm Optimization (PSO). We employ PSO to find near-optimal parameters for static environments and then train a neural network to interpolate PSO solutions in order to improve the UAV route in dynamic environments. Experimental results showed a gain in the spraying precision in dynamic environments when ANN and PSO were combined. We demonstrate the improvement in figures when compared against the exclusive use of PSO. This approach will be embedded in UAVs with programmable boards, such as Raspberry PIs or Beaglebones. The experimental results demonstrate that the proposed approach is feasible and can meet the demand for a fast response time needed by the UAV to adjust its route in a highly dynamic environment, while seeking to spray pesticides accurately.

Keywords: Unmanned aerial vehicle; agricultural applications; dynamic environments; neural networks; evolutionary algorithms.

1. Introduction

Pesticides, also known as agrochemicals, are generally applied in agricultural crop fields to increase productivity, improve quality and reduce production costs.

However, prolonged contact (either directly or indirectly) with these products can cause various diseases to humans, such as several types of cancers, complications to the respiratory system and neurological diseases.[1] It is estimated that about 2.5 million tons of pesticides are used each year throughout the world and this amount is growing.[2] Much of the pesticide is wasted during the spraying process due to the type of technology employed. Evidence show that the drift of pesticides is generally found at a distance of 48 m to 800 m from the target crop field; the deviation can range from a distance of 5 km to 32 km downwind.[3]

The use of unmanned aerial vehicles (UAV) to carry out the task of spraying pesticides can have several benefits, including (i) to reduce human contact with the chemicals, which helps to preserve human health; and (ii) to improve the performance of the spraying operation, by avoiding the presence of chemicals outside the designated areas, which is important to protect the neighboring fields that may have other crops, and protect nature reserves or water sources. The sets of control rules to be employed in an autonomous UAV are very hard to put into effect and even harder to fine-tune to each environmental feature. Due to the technical features of each UAV, a fine-tuning phase must include the parameters of the algorithm. This process must also take into account the type of crop being handled and the type of pesticide being used.

The proposed architecture employs an UAV that has an attached spraying system and is able to communicate with a wireless sensor network (WSN), which is arranged in a matrix-like grid on the crop field. The WSN sends feedback on the weather conditions and determines how the pesticide is actually being applied on the target crop field. On the basis of the received information, the UAV appropriately adopts a policy that allows it to correct its route. Hence, the main contributions of this research are: (i) to investigate an evolutionary methodology capable of minimizing human contact with pesticides, (ii) to evaluate an evolutionary approach that is able to reduce errors when spraying pesticides in areas where vegetables and fruits are grown, (iii) to investigate techniques able to maximize quality in agricultural production, and (iv) to increase the autonomy of the architecture proposed by Faiçal *et al.*,[4] in which the policy parameters were set out empirically and could be applied regardless of the weather conditions.

This paper extends the previous work[5] by presenting a proposal and an evaluation on how UAVs can be controlled in a highly dynamic environment, such as environments with sudden changes in the speed and direction of the wind. To this aim, we devised an ANN to be employed in real-world operations, which was built with evolved values employing a PSO approach. We employ the PSO to find near-optimal parameters for static environments and then train a neural network to interpolate the PSO solutions in order to improve the UAV route in dynamic environments. Neural networks have an intrinsic mapping and generalization features, which make them a good choice for dynamic environments,[6,7] while the evolutionary approach is a good mean to discovering non-trivial parameters.[8,9] Combining the evolutionary technique with the neural approach in this work allowed us to leverage the best capabilities of each technique. In such a way, we propose the use of an ANN for quick decision-making, since in real environments the weather conditions change suddenly and at short intervals of time. The new proposal provides a significant advance in the optimization of an UAV route which can be used in real environments, as a trained ANN is faster than running the evolutionary process of PSO technique over and over again whenever the weather conditions are changeable. Moreover, even if the employed hardware has enough resources to perform the technique PSO quickly, the ANN will enable the intensity of the route adjustment to be adjusted in a shorter time.

This paper is divided into six sections. Section 2 examines other studies related to this paper. Following this, Section 3 provides an outline of the architecture to clarify the scope of this paper and the optimization methodology proposed in this work. The experiments and results are analyzed and discussed in Section 4, and then compared with the results found in the literature. Finally, Section 5 summarizes the conclusions obtained from the results and suggests how this paper might encourage further studies in this field.

2. Related Work

There are several studies that suggest how UAVs or WSNs can be employed for monitoring agricultural production, occasionally by integrating both technologies.[10–12] However, this work differs in so far as it proposes a particle swarm optimization algorithm to optimize the control rules of the UAV at runtime, based on feedback provided by WSN about weather conditions in the agricultural field.

Valente *et al.*[13] describe a WSN-based system and UAV to monitor vineyards. The WSNs collect information about weather, soil and planting conditions and then make it available to farmers. However, a field crop may be hundreds of meters away from other fields and sometimes there are barriers (e.g. rivers and roads) that separate two crop fields. Thus, it may not be feasible or cost-effective to use cables to connect the WSN. Although the use of powerful wireless devices allows communication between WSNs, this solution leads to higher energy consumption and involves reducing the lifetime of the nodes. One solution that can be adopted to overcome these limitations is the employment a UAV to fly over the crop fields and gather information from each WSN, which can then be conveyed to a processing center. Although this study demonstrates that UAVs and WSNs can be integrated to provide efficient solutions or improvements in an agricultural setting, no methodology is employed for optimization at runtime. Additionally, a UAV is used as a mobile node in a WSN without having any adverse effects on the environment.

Huang *et al.*[14] devise a particular system for spraying pesticide. This system should be coupled with a UAV that is capable of carrying approximately 22.7 kg. The model used in this work is UAV SR200 (manufactured by Rotomotion). The spraying system consists of four main components: (i) a metal tube with nozzles;

(ii) a tank to store pesticide; (iii) a pump to move the liquid; and (iv) a mechanism for controlling the activation of the spray. The spraying system can carry up to 5 kg of pesticide, which is enough to spray 14 ha; and it has a flight time of around 90 min. The main objective of this study is to validate the proposed system and evaluate different types of spray nozzles. However, the weather conditions were not taken into account. Additionally, it does not include a discussion of an evolutionary methodology that is able to optimize control of this activity.

Faiçal *et al.*[4] proposed an architecture formed of a UAV and WSN to spray pesticide in crop fields. It is known that adverse weather conditions, such as high-speed winds, can cause errors in the spraying process. The study shows how the recommended architecture can reduce the risk of errors and increase control over this activity. With the aid of feedback from the WSN on pesticide concentrations, the route is gradually changed until the sensor node can identify the correct application of the product. However, the parameters set for the route change are applied in different weather conditions, which might impair the performance of this architecture. As mentioned earlier, this paper addresses this limitation by evaluating a methodology that is employed for the fine-tuning of a parameter that ponders the changes in the intensity of the route followed by the UAV.

3. Proposed Approach

3.1. *UAV and WSN architecture for spraying on crop fields*

Figure 1 illustrates how the UAV acts as an agent on the crop fields. The UAV is equipped with a spraying system and a communication module, which enables data exchange with a WSN arranged on the crop fields; it flies over the area and sprays the pesticide in its entire length. The WSN is only depicted within the targeted crop fields and is bounded by two dark dashed lines (from top left to bottom right) to simplify the viewing image. At the top of Fig. 1, there are two arrows that indicate

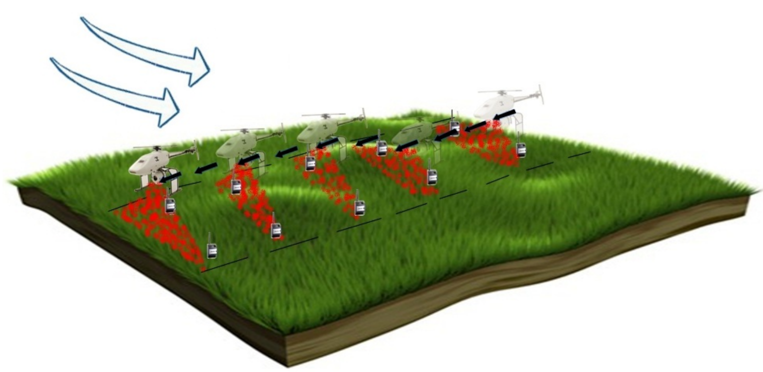


Fig. 1. Example of spraying in crop fields with the architecture proposed by Faiçal *et al.*[4] This architecture consists of a UAV (to spray) and WSN (to monitor). If the WSN detects an unbalanced spray on its sensor nodes, the UAV changes its route to correct the spraying of the pesticide.

the wind direction at a particular location. Through its communication link with the WSN, the UAV is able to obtain information about the weather (e.g. speed and direction of the wind) and the concentration of pesticides sprayed on the crops. If an imbalance is detected in this concentration (e.g. the sensor on the left identifies a higher concentration than the sensor on the right), possibly caused by the wind, the UAV adopts a policy that involves changing its route to balance the application of pesticides throughout the whole extent of the targeted crop fields. This policy also helps to prevent overlapping when the chemical is applied. In Fig. 1, the correction of the route is represented by small arrows between the images of the UAV.

A parameter called *routeChangingFactor* is employed in the route change function to set the degree of intensity (e.g. mild or sharp) so that the change can be made. However, despite the importance of this parameter to ensure the success of the spraying, its value is set empirically before the beginning of the flight and is used for all weather conditions that occur during the spraying process. This characteristic can affect the quality of the spraying; for example, a sharp correction might be made in an environment where a low wind speed has been detected. Moreover, an increase in the complexity of this environment might cause variable behavior. In other words, the weather conditions can change during the activity, and this is detrimental to the whole architecture if it has a static configuration.

The *routeChangingFactor* parameter is a weighting variable used in the calculation of the period of time assigned for a UAV route change.[4] It defines if the route change will be of mild intensity (low value, resulting in more time for a change of route) or high intensity (high value, resulting in a short time to be re-routed). Equation (1) illustrates the time that the UAV remains in change of route is set. In this equation, ls (left sensor) and rs (right sensor) are data received from the pair of sensors deployed inside the plantation and located in the spraying tracks (see Fig. 1), τ is the *routeChangingFactor* and Δ corresponds to the period of time assigned for the route change.

$$\Delta = \frac{|ls - rs|}{\tau}\,. \tag{1}$$

This equation is used by the UAV control policy, which sets a minimum threshold for the difference between the values from the pair of sensor to decide whether the route change should occur. If the difference is larger than the threshold, the UAV control policy re-defines the duration of the route change (based on Eq. (1)), the angle and the direction required for the aircraft.

To overcome the problems previously mentioned, this paper proposes a methodology based on Particle Swarm Optimization to optimize the parameter *routeChangingFactor*. As previously mentioned, the parameter of route change has a large influence on spraying and, in addition, the architecture is employed in a dynamic environment. Thus, it is worth investigating a methodology that is able to find a value for the parameter *routeChangingFactor* (and is close to an optimal solution). Figure 2 shows the behavior of the architecture when the optimization

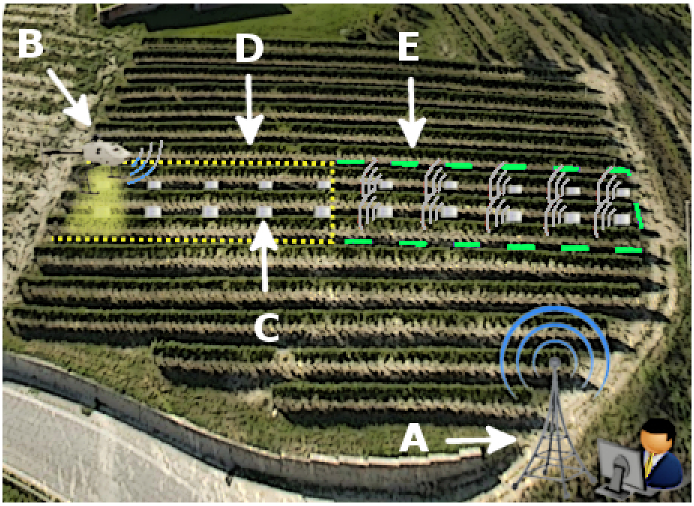


Fig. 2. Behavior of the architecture that employs the proposed optimization methodology. The Control Station (A) is installed outside the target crop field, in a zone within communication range of the UAV (B). During the spraying of the current crop field (D), the UAV sends a request for weather information about the next crop field (E) to the WSN (C). When the requested information is received, the UAV sends it to Control Station (A) where it will be used by the optimization methodology. At the end of the optimization, the Control Station sends the new configuration back to the UAV. The settings will be updated when the spraying of the current crop field has been completed and the spraying of the next crop field is about to begin.

methodology is used. It assumes that a crop field is composed of several small virtual subareas with a rectangular shape. Thus, if all the subareas are sprayed, this results in a complete spraying of the crop field. Each subarea will be called a "crop field" during this study. The UAV's flight plan is designed to ensure that the next crop field will be sprayed right after the work on the current crop field has been completed. The route change, as described earlier, is made in the current crop field (D). Running parallel with this activity, the UAV (B) queries the WSN (C) about the weather conditions in the next crop field (E). At this stage the request can reach the nodes that are deployed inside the next crop field by using multi-hop links (not shown in the diagram). Only the endpoints of the communication (source and destination) are shown for a clear image. As soon as the UAV obtains weather information, this is sent to Control Station (A) to optimize the parameter *routeChangingFactor*. At this time, the optimization methodology proposal is executed on the basis of the weather information. At the end of the optimization, the best value of the parameter is sent back to the UAV. When the spraying of the current crop field (D) is finalized, the UAV updates its settings so that the spraying of the next crop field (E) can start. It should be highlighted that the use of a Control Station provides more powerful computation and, in addition, allows a pilot (on the ground) to oversee the flight and, if necessary, intervene in the control of the UAV.

3.2. *Optimization of control rules*

The optimization methodology proposed in this paper is essentially composed of an algorithm based on PSO.[15,16] This algorithm searches for a non-optimal value for the parameter *routeChangingFactor* and in one computation model of the environment evaluates the accuracy of spraying by applying the weather information received from the WSN. Lastly, the algorithm returns the best solution (value per parameter) and this is assessed so that it can be applied in the next crop field. One important condition of this algorithm is that the computational cost (runtime) should be lower than the time required for spraying a single crop field (subarea). Hence, the search space is restricted to one zone that has values of different acuteness (e.g. abrupt, smooth and moderate). Additionally, the delimitation of the search space allows a faster convergence.

The optimization process is conducted in two ways simultaneously: (i) through cooperation (group learning) and (ii) competition (single learning), by considering the particles of a swarm. Each particle is initialized in a random position (possible solution) within a search space. In each iteration of the algorithm, the velocity and position of the particles are updated. The position found by the swarm with best fitness (as well as the positions with best fitness found by each particle individually) are considered for updating. As the positions of the particles are possible values for the parameter *routeChangingFactor* contained in the search space, the velocity of the particle indicates how far and in what direction this value will move (to a new position). The new position of each particle is obtained by Eq. (2) (where: X_{id} is the position and V_{id} is the velocity of particle $_i$ in an instant $_d$), while the velocity is updated in each iteration with Eq. (3) (where: w_i is the inertia, C_1 and C_2 establish the importance of social trend or individual (cooperation or competition), P_{id} is the best position found by individual particle, P_{gd} is the best position found by the swarm and, finally, $rand()$ and $Rand()$ are different random values for a good exploration of search space).[17]

$$X_{id+1} = X_{id} + V_{id}\,, \tag{2}$$

$$V_{id} = w_i * V_{id} + C_1 * rand() * (P_{id} - X_{id}) + C_2 * Rand() * (P_{gd} - X_{id})\,. \tag{3}$$

Algorithm 1 shows details of the optimization process. The particles are initialized in random positions inside the search space. The stop condition is defined by the amount of iteration that the algorithm has to run. This stop condition allows the average runtime to be analyzed in the worst case scenarios, when all the iterations have been executed to find one possible solution. Following this, one stop condition can be added with the aim of finalizing the algorithm after confirming that convergence has occurred. It should be noted that the runtime in worst cases should be shorter than the time required for spraying a crop field (subarea). In each iteration, all the particles will have their positions evaluated and if the "fitness" of a particle is the best found by the swarm so far, the algorithm stores this position. On the other hand, if the position is not the best globally, but is the best of the

Algorithm 1: Proposed algorithm to optimize the *routeChangingFactor* parameter.

```
1:  InitializeParticles(RandomPosition[1, 10])
2:  for MAX_ITERATION do
3:    PARTICLES ← FirstParticle()
4:    for ALL_PARTICLES do
5:      Result ← FuncObjetive(PARTICLES)
6:      if Result is best particle then
7:        Stores the position in particle
8:      end if
9:      if Result is the best in the swarm then
10:       Stores the position in swarm
11:     end if
12:     UpdateVelocity(PARTICLES)
13:     NewPosition(PARTICLES)
14:     PARTICLES ← NextParticle()
15:   end for
16: end for
17: return BestGlobalPosition
```

particle, the algorithm also stores this position in the particle. Later on, the velocity and the position of each particle are updated. When the algorithm achieves maximum interaction, it is finalized and the best position found by the swarm is returned.

The objective function (FuncObjetive) contained in the Algorithm, cited in Line 5 of Algorithm 1, refers to an interaction with one project inside the OMNeT++ software. The project is an implementation of a computational model to evaluate the spraying.[4] This interaction tests and analyzes the quality of spraying in each position of all the particles. The OMNeT++[a] is a simulator of discrete events based on C++ language to model networks, multiprocessors and other distributed and parallel systems.[18] The OMNeT++ can be used to model several types of networks, such as networks of queues, wireless and peer-to-peer types.[19] Because of its generic design, OMNeT++ has several frameworks established for specific networks, such as Mixim[b] for modeling wireless networks. This framework provides detailed models for wireless channels, wireless connections, mobility models, models for dealing with obstacles and several communication protocols, especially for MAC.[20] Figure 3 shows the connection between the algorithm and OMNeT++.

[a] OMNeT++ Network Simulation Framework, http://www.omnetpp.org
[b] MiXiM project, http://mixim.sourceforge.net

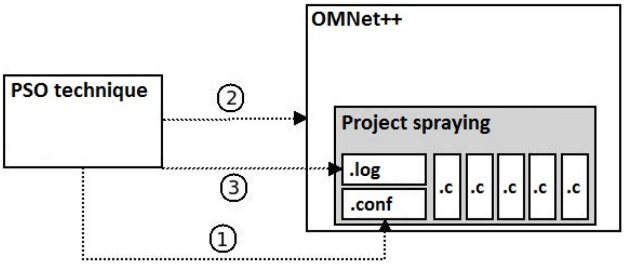


Fig. 3. Interaction between PSO technique and OMNeT++.

Initially the algorithm changes the settings and files of "Project spraying" so that the position of the particle can be used as *routeChangingFactor*, apart from the addition of real weather information (Stage 1). After that, the algorithm runs "Project spraying" in OMNeT++ (Stage 2) and, finally, analyzes the log file to determine the results of the spraying (Stage 3). In the source code of "Project spraying" there is a dispersion model to estimate the movement of pesticide until it reaches the planting [plantation ?].[4] The fitness is calculated by estimating the amount of pesticide sprayed outside of the target crop field.

Thus, the objective function used by the PSO technique consists of two stages: (i) the execution of the computational model for the spraying of the agricultural field with the parameters set by the algorithm; and (ii) an analysis of the concentration of pesticide deposited in the agricultural field. In the first stage, the algorithm adjusts the computational model to the received weather conditions and the parameter *routeChangingFactor* being analysed, and runs the simulator to estimate how the spraying will be performed in these conditions. This execution returns a matrix with dimensions proportional to the size of the agricultural field and element values representing the concentration of the product deposited in each square meter. It must be observed that the value of the parameter *routeChangingFactor* will be changed during the optimization process. In the second stage, the pesticide concentration matrix is analyzed and the amounts deposited outside the target area are added to be used as the fitness value. Thus, the smaller the fitness, the better (more accurate) is the spraying carried out with the considered *routeChangingFactor*.

3.3. *Proposed approach for dynamic environments*

One of the characteristics of the PSO is that the search for the best values occurs in static environments. However, the evolutionary approach is often very time-consuming, and hence, it is not trivial to employ it in embedded software for dynamic operations. The operation in this case is dynamic since the UAV can change its speed and height or there may also be a change in the wind itself. Neural networks have intrinsic mapping and generalization features, which make them a good choice for dynamic environments while the evolutionary approach is a good means of discovering non-trivial parameters.

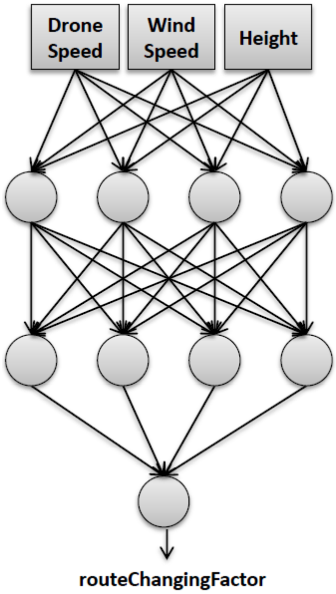


Fig. 4. ANN topology.

For an approach which can handle dynamic environments, we designed and evaluated how a neural network can be built upon data from the evolutionary algorithm. Hence, we ran the evolutionary technique in 27 static different environments and used its results to train the neural network. The 27 different scenarios were built in the light of the following variations: UAV speed (m/s) {10, 15, 20}; wind speed (km/h) {0, 10, 20} and UAV height of operation (m) {10, 15, 20}. We ran the evolutionary algorithm 10 times for each scenario, and obtained 270 different values. These values were then used for training the ANN. It should be highlighted that for each static scenario, the values obtained by the PSO were not the same, but often similar. We evaluated 5 ANN with different topologies to investigate which is the smallest neural network that can achieve the highest degree of accuracy.

Figure 4 shows the ANN topologies. The ANN inputs are the speed of the UAV, wind speed and UAV height and the output is the parameter *changeRouteFactor*. The results of the evaluation are given in Section 4.3.

4. Evaluation and Analysis of the Experimental Results

This section includes a description of our evaluations and examines our results. It is subdivided into three subsections which aim to explain (i) the evaluation of the optimization of the *routeChangingFactor*, (ii) the comparison resulting from the evolutionary approach with pre-programmed rules, i.e. without optimization of rule controls for route changes, as discussed by Faiçal *et al.*,[4] and (iii) evaluation of the application of the neural network for dynamic environments.

Table 1. Results of the optimization of the *routeChangingFactor* parameter. The first column shows the set of evaluated PSO as P#I# meaning P (number of particles) and I (number of interactions).

Settings	Convergence Rate (%)	Average Time of Evolutions (s)
P3I20	96.77	18.617 ± 0.371
P3I50	100.00	45.927 ± 0.649
P3I100	100.00	93.854 ± 1.555
P5I20	100.00	30.705 ± 0.506
P5I50	100.00	77.162 ± 0.766
P5I100	100.00	158.995 ± 3.143
P10I20	100.00	62.549 ± 0.912
P10I50	100.00	157.957 ± 2.976
P10I100	100.00	313.335 ± 1.488
P15I20	100.00	93.606 ± 0.799
P15I50	100.00	235.189 ± 1.816
P15I100	100.00	480.359 ± 14.762
P20I20	100.00	125.088 ± 1.059
P20I50	100.00	312.894 ± 2.058
P20I100	100.00	628.324 ± 2.251

4.1. *Optimization of the routeChangingFactor parameter*

In this stage, the algorithm will search for the best possible value when applying it as the parameter of route changes (taking into account the feedback obtained from the weather information). The evaluated settings are called as: P#I#, meaning P (number of particles) and I (number of interactions). Each configuration is replicated thirty times to obtain a greater confidence level for future statistical analysis. The algorithm is defined so that it will prefer the social trend ($C_2 = 0.75$) to the individual trend ($C_1 = 0.25$) in the search. Another important parameter for running the algorithm is *Inertia*, which is used to strike a balance between local and global searches, and is set to carry out local searches ($w_i = 0.1$). This configuration aims at a "quick pull" of the swarm of particles to a place considered promising because it contains a better intensity than the others found so far. Moreover, it is expected that the particles will carry out a thorough search in the region where they are located. It is notable that both the ability not to remain stuck in local minima and the convergence of the algorithm were considered in this study, which showed a satisfactory performance.

Due to the low communication time, measured in Ref. 4, it can be assumed that the communication time between the UAV and Control Station does not have a significant influence on the total runtime. Thus, it can be assumed from this experiment that the weather information is already in the Control Station.

This subsection shows the results when the PSO-based algorithm described in Section 3.2 is employed. Table 1 shows the results of the first stage. Apart from

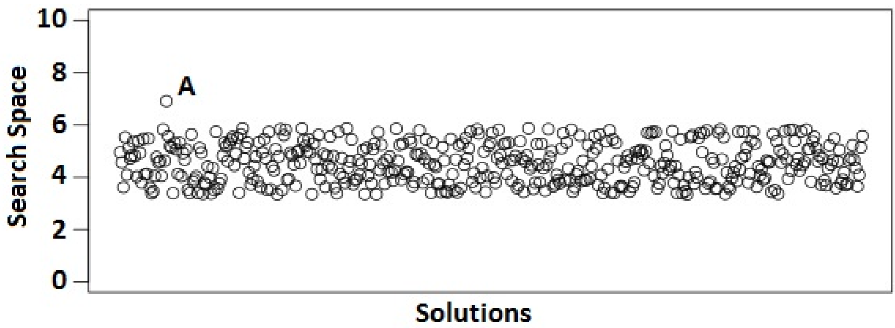


Fig. 5. Representation of the solutions found by the algorithm in the search space.

P3I20 setting, that has a 96.77% convergence rate, all the others have a 100% convergence rate for the same value of fitness. Owing to particular features of the problem, it is possible that a group of solutions has a fitness that is similar but not the same, since the difference between the values of the parameter *routeChangingFactor* may be low enough to have no significant influence on the spraying in specific situations.

It can be seen in Table 1 that the P3I20 setting is the only configuration that does not have a convergence rate of 100%. Another important point in Table 1 is the average time ± standard deviation (in seconds) for each setting of the algorithm. The spraying of a target crop field is carried out in ≈ 65 seconds (in accordance with the speed of the UAV) and as mentioned previously the runtime must be less than the time required for spraying a target crop field. Hence, the settings that are feasible for this application are P3I50, P5I20, and P10I20. These settings allow the optimization of the parameter *routeChangingFactor* with an appropriate time (less than 65 s) and with a convergence rate of 100%.

In conducting an analysis of the position of the solutions in search space and visualizing the non-convergent solution, we have plotted all the solutions on the basis of their value in search space (see Fig. 5). It can be seen that the proposed algorithm is capable of finding a region in search space where values are appropriate for the parameter *routeChangingFactor* in specific climatic conditions. This region in search space is closely connected with features of the environment and tends not to be an appropriate region for the next crop field, since it is a dynamic environment. Thus, the algorithm should run before the spraying in each crop field is started to reduce the risk of making a wrong decision. The non-converged solution originating from the P3I20 setting, is marked as "A" in Fig. 5. Despite its proximity, this solution does not belong to the region of appropriate solutions for the weather conditions reported by the WSN.

After analyzing the optimization of the parameter *routeChangingFactor*, we conducted experiments aimed at evaluating the precision of the spraying by using the solution indicated by the algorithm.

4.2. *The spray operation on crop fields*

This stage involves the use of the solution which has best fitness (found in the previous stage) to evaluate the spraying on a target crop field. This selection criterion is used to evaluate the best solution in the group of alternatives generated by replications. If all the replications converge in a group of solutions with equal fitness, one of the solutions is randomly selected. The spraying is carried out by using the value selected as the parameter *routeChangingFactor* and the result is compared with the results without optimization from Faiçal *et al.*,[4] where a fixed value was employed. It is worth noting that the environmental features are the same for all the experiments and this is called *Constant Light Wind* by Faiçal *et al.*[4] This environment has a constant wind at a speed of 10 Km/h. The crop field used has an area of 1100 m $\times$ 150 m and the area of the target crop field is 1000 m $\times$ 50 m. The WSN has twenty-two nodes spread across the target crop field and the UAV initializes the spraying at a height of 20 meters above ground and at a constant speed of 15 m/s. At intervals of ten seconds, the UAV makes requests to the WSN to obtain information about the quality of the spraying. These experiments are replicated seventy times, to obtain a greater level of confidence for future statistical analysis. In the following subsection, the results are shown and discussed.

This subsection shows the results of the second stage of experiments. This involved analyzing and discussing the results of spraying in a crop field by using the solutions found by the PSO. In this stage, the experiments were conducted to support the assessment of the proposal, which entailed optimizing the parameter *routeChangingFactor* and ran parallel with the spraying of a crop field (in the first stage) and applied the results of the optimization to subsequent crop fields (the second stage). The results of spraying where the optimization method was used, are compared with the results when there was no optimization as discussed by Faiçal *et al.*[4]

The following settings were adopted: *CL10*, interval of ten seconds between each of the requests of weather information from UAV to WSN; *CL30*, interval of thirty seconds between each of the requests of weather information from UAV to WSN; *CLNO* does not change its route. The settings that use an optimization parameter are *P5I20*, *P10I20*, and *P3I50*. These results are obtained by the PSO.

Figure 6 and Table 2 show the results of spraying on target crop field, and compare the results from Faiçal *et al.*[4] with the results of the proposed PSO. It is clear that there is an increase in the area with a correct application of pesticides when the evolved *routeChangingFactor* parameter was applied. The *CL10* is the setting with the smallest error rate among all the non-optimized settings. However, all the optimized settings surpass the precision rate usually achieved when spraying a target crop field. Figure 7 displays a heat map to represent the chemicals sprayed on the crop at the end of the simulation.

The Shapiro Wilk method, employed for the statistical analysis, shows that the hypothesis of normality is rejected for one of the sets when there is a confidence level

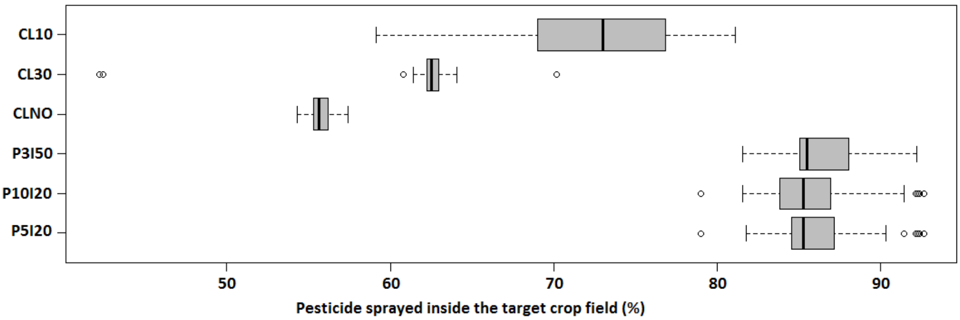


Fig. 6. Percent of pesticide spraying inside the target crop field. In this boxplot, the first three results come from Faiçal *et al.*[4] and the last three results were obtained in this work by the proposed PSO.

Table 2. Correct spraying (%) in the target crop field.

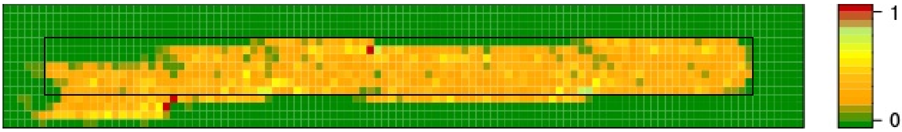
Settings	Area with Correct Coverage (%)
CL10	72.871 ± 4.659
CL30	62.113 ± 3.591
CLNO	55.697 ± 0.657
P3I50	86.220 ± 2.538
P5I20	85.811 ± 2.894
P10I20	85.777 ± 2.520

of 95%. In view of this, we decided to use non-parametric tests in the subsequent analysis.

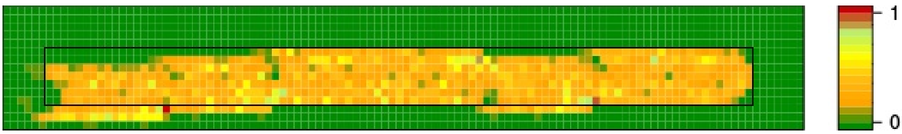
The pairwise comparisons were performed by means of the Wilcoxon Rank Sum Test (see Table 3) and show that there are significant differences between the results that employ the methodology for optimization and the results when this methodology is not used. However, no significant differences were found when only the settings based on the optimization methodology were analyzed. Additionally, the Friedman Rank Sum Test is also applied to this data and shows a p-value of 0.000, which suggests that there are significant differences between the results shown in Fig. 6. As a result, it can be concluded that the use of the optimization method for the parameter *routeChangingFactor* increases the efficiency of the control rules, and reduces the errors when spraying in a crop field.

4.3. *Use of ANNs for dynamic environments*

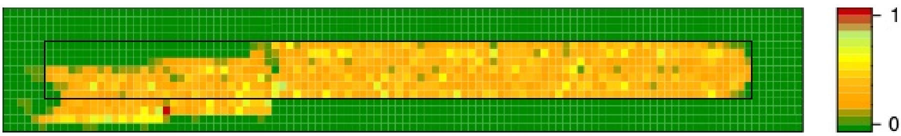
This section analyzes the ANN trained to interpolate and generalize the data from 27 static scenarios evolved by the PSO. As previously stated, the 27 different scenarios were built in the light of the following variations: UAV speed (m/s) {10, 15, 20}; wind speed (km/h) {0, 10, 20} and UAV height of operation (m) {10, 15, 20}.



(a) $routeChangingFactor = 3.000$



(b) $routeChangingFactor = 6.000$



(c) $routeChangingFactor = 7.164$

Fig. 7. (Color online) A heat map to represent the chemicals sprayed on the crop at the end of the simulation. The green colour represents no pesticide and red represents the most concentrated places. The thin black lines show the crop field that needs to have chemicals sprayed. (a) and (b) Evaluations with empirical values. (c) Evaluation with *routeChangingFactor* obtained by the PSO. We can see that when employing the *routeChangingFactor* obtained by the PSO we have the best adjusts in the UAV track, attempting to keep the chemicals within the boundary lane. It is worth to highlight that, as the simulation starts with wind, the UAV always starts the dispersion of the chemicals outside the boundary.

Table 3. Results of Wilcoxon Rank Sum Test. There are evidences of difference between the evolved values (P*) and the non-evolved values (C*) from Faiçal *et al.*; (p-values less than 0.05). There are no evidences of difference among evolved values (p-values greater than 0.05).

	CL10	CL30	CLNO	P3I50	P5I20
CL30	0.000				
CLNO	0.000	0.000			
P3I50	0.000	0.000	0.000		
P5I20	0.000	0.000	0.000	0.52	
P10I20	0.000	0.000	0.000	0.52	0.79

We ran the evolutionary algorithm 10 times for each scenario, and obtained 270 different values.

We sought to obtain the smallest ANN that would provide the most accurate values, since this also reduces the chance of overfitting during the training and improves the generalization of the ANNs. Hence, we started evaluating neural networks with one hidden layer and with 1 to 5 neurons. No ANN with these topologies

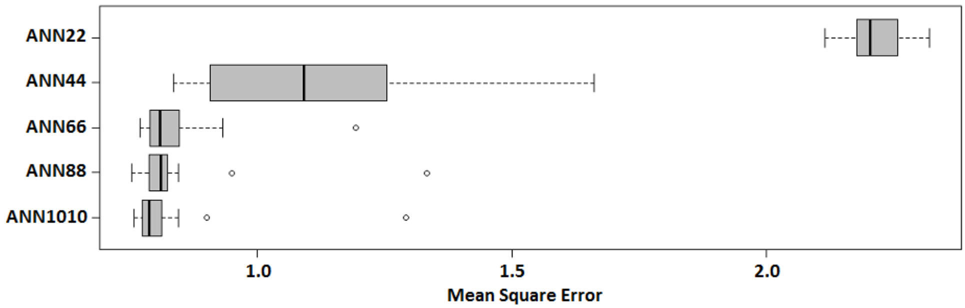


Fig. 8. Mean square error for 30 runs of each ANN topology.

was able to learn an accurate model from the data. therefore, the number of neurons and the number of layers were increased, leading to the following topologies for the first and second hidden layers: $\{2 \times 2, 4 \times 4, 6 \times 6, 8 \times 8, 10 \times 10\}$. The input layer has 3 neurons and the output has one neuron (as described in Section 3.3).

The evaluated ANNs are feed-forward multi-layer perceptron and are trained with the resilient backpropagation algorithm. The ANNs were built and trained by employing the Stuttgart Neural Network Simulator (SNNS).[c] We ran the training 30 times for each of the ANN topologies and employed 3-fold cross validation. The ANNs were trained for 2000 cycles, although we used the values of the best generalization point. The results as mean square error (MSE) can be seen in Fig. 8.

The distributions were evaluated with statistical tests (Shapiro-Wilk) that showed that most of the distributions cannot be accepted as normal distributions. Hence, the comparison between the distributions was carried out with the Wilcoxon-Mann-Whitney test. When 1% of significance is considered, the comparisons between ANN88 and ANN1010 are equivalent. No other comparison of distribution showed equivalence with the ANN1010 distribution. We can see that there is an improvement from ANN22 to ANN88; however, as the statistical test showed that ANN88 and ANN1010 are equivalent, the ANN88 was considered for the deployment.

Figure 9 displays an execution of the ANN88. The black dots represent the expected (original) values and the blue dots represent the values obtained by the ANN. It can be seen that there is a good fit for most of the points; however, there are points in which the obtained values differ from the expected. The reason for this is that the PSO does not obtain single values while performing the evolution, i.e. there is a group of good solutions within a range. Figure 5 can enable us to understand which good solutions are between ≈ 3 and 6, and thus, this PSO response can be interpreted as if the function being evolved has plateau regions. The current ANN topology allows unique outputs for the same inputs, which might be interpreted as a disperse value, although, the type of dispersion shown in the diagram does not

[c]Stuttgart Neural Network Simulator, http://www.ra.cs.uni-tuebingen.de/SNNS

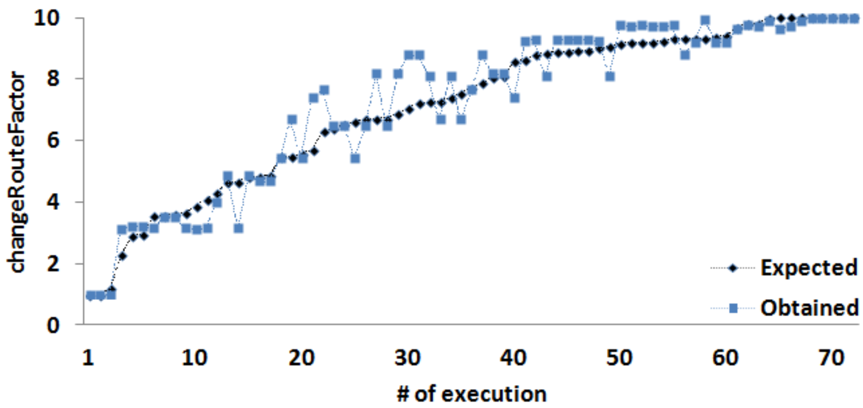


Fig. 9. (Color online) Results of execution of the ANN88 for ≈ 75 different inputs. The black dots represent the expected (original) values and blue dots represent the values obtained by the ANN.

lead to failure in the spraying operation because the obtained values are within a suitable range.

5. Conclusions and Suggestions for Future Work

In this paper, we have proposed and evaluated a methodology based on PSO, for fine-tuning the control rule of a UAV, and on an ANN to increase the support for high dynamic environments. The simulations with PSO provide the optimization of the parameter *routeChangingFactor* and thus reduce the error rate when spraying pesticides on crop fields. In the first experiments, we evaluated a broad set for the optimization method and the results show that it is possible to obtain 100% of convergence. Applying such evolutionary methodology allowed us to increase the precision of spraying pesticides so that ≈ 86% of the product can be applied within a target crop field. The reason for this is that the optimization is performed during the application and thus the parameter can be adapted to the weather conditions of each target crop field. Although, taking into account that the spraying operation might occur in highly dynamic environment due to changes in wind speed and direction, we devised an ANN to be employed in the real-world operations. The proposed ANN is trained with a dataset of near-optimal parameters obtained by the PSO that evolves for a limited set of static environments. The ANN training process allows it to interpolate the results as so it can be applied dynamically for any configuration of the environment. Combining the evolutionary technique with the neural approach in this work allowed us to leverage the best capabilities of each technique. The presented proposal provides a significant advance in the optimization of an UAV route which can be used in real environments, as a trained ANN is faster than running the evolutionary process of PSO technique over and over again whenever the weather conditions are changeable. Moreover, even if the employed hardware has enough resources to perform the technique PSO quickly, the

ANN will enable the intensity of the route adjustment to be adjusted in a shorter time.

On the basis of the results obtained the following are recommended for further studies: (i) an investigation of how more parameters can be optimized (e.g. the height and speed of the UAV, the best starting-position for the next crop field, and the pressure of the spraying system); (ii) an investigation of different methodologies for the fine-tuning control rules of UAV (e.g. Differential Evolution,[21] Genetic Algorithms,[22] Hill-Climbing,[23] NSGA-II[24]); (iii) an analysis of the feasibility of embedding the optimization methodology in the UAV, leading to an autonomous architecture; (iv) an investigation of the methodologies required for a weather-aware router planner.

Acknowledgments

Bruno S. Faiçal would like to acknowledge the funding received for your doctoral project by FAPESP (process No. 2013/18859-8). The authors would like to acknowledge the financial support granted by FAPESP (processes ID 2012/22550-0). We also would like to thank the Office of Naval Research Global (No. 62909-14-1-N241). Finally, the authors would like to acknowledge CAPES (Capes Foundation, Ministry of Education of Brazil), CNPq (Brazilian National Counsel of Technological and Scientific Development) and FAPESP (Sao Paulo Research Foundation).

References

1. D. D. Weisenburger, Human health effects of agrichemical use, *Human Pathology* **24**(6) (1993) 571–576.
2. M. I. Tariq, S. Afzal, I. Hussain and N. Sultana, Pesticides exposure in Pakistan: A review, *Environment International* **33**(8) (2007) 1107–1122.
3. D. Pimentel, Amounts of pesticides reaching target pests: Environmental impacts and ethics, *Journal of Agricultural and Environmental Ethics* **8**(1) (1995) 17–29.
4. B. S. Faiçal *et al.*, The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides, *Journal of Systems Architecture* **60**(4) (2014) 393–404.
5. B. Faiçal, G. Pessin, G. Filho, A. Carvalho, G. Furquim and J. Ueyama, Fine-tuning of uav control rules for spraying pesticides on crop fields, in *IEEE 26th Int. Conf. on Tools with Artificial Intelligence* (*ICTAI*) (2014), pp. 527–533.
6. J. A. Suykens, J. P. Vandewalle and B. L. de Moor, *Artificial Neural Networks for Modelling and Control of Non-Linear Systems* (Springer Science & Business Media, 2012).
7. B. Yegnanarayana, *Artificial Neural Networks* (PHI Learning Pvt. Ltd., 2009).
8. M. Clerc and J. Kennedy, The particle swarm — Explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation* **6** (February 2002) 58–73.
9. L. M. Rios and N. V. Sahinidis, Derivative-free optimization: A review of algorithms and comparison of software implementations, *Journal of Global Optimization* **56**(3) (2013) 1247–1293.
10. H. Xiang and L. Tian, Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV), *Biosystems Engineering* **108**(2) (2011) 174–190.

11. B. Li, R. Liu, S. Liu, Q. Liu, F. Liu and G. Zhou, Monitoring vegetation coverage variation of winter wheat by low-altitude UAV remote sensing system, *Nongye Gongcheng Xuebao/Transactions of the Chinese Society of Agricultural Engineering* **28**(13) (2012) 160–165.
12. J. Makynen, H. Saari, C. Holmlund, R. Mannila and T. Antila, Multi- and hyperspectral UAV imaging system for forest and agriculture applications, in *Proc. of SPIE — The International Society for Optical Engineering*, Vol. 8374 (Baltimore, MD, United States, 2012), The Society of Photo–Optical Instrumentation Engineers (SPIE).
13. J. Valente, D. Sanz, A. Barrientos, J. Cerro, A. Ribeiro and C. Rossi, An air-ground wireless sensor network for crop monitoring, *Sensors* **11**(6) (2011) 6088–6108.
14. Y. Huang, W. C. Hoffmann, Y. Lan, W. Wu and B. K. Fritz, Development of a spray system for an unmanned aerial vehicle platform, *Applied Engineering in Agriculture* **25**(6) (2009) 803–809.
15. R. C. Eberhart and J. Kennedy, A new optimizer using particle swarm theory, in *Proc. of the Sixth Int. Symp. on Micro Machine and Human Science*, Vol. 1 (New York, 1995), pp. 39–43.
16. L. A. Santana and A. M. Canuto, Particle swarm intelligence as feature selector in ensemble systems, in *2013 Brazilian Conference on Intelligent Systems* (*BRACIS*) (IEEE2013), pp. 89–94.
17. G. Pessin, F. S. Osório, J. R. Souza, J. Ueyama, F. G. Costa, D. F. Wolf, D. Dimitrova, T. Braun and P. A. Vargas, Investigation on the evolution of an indoor robotic localization system based on wireless networks, *Applied Artificial Intelligence* **27**(8) (2013) 743–758.
18. A. Varga, OMNeT++, in *Modeling and Tools for Network Simulation*, eds. K. Wehrle, M. Günes and J. Gross (Springer Berlin Heidelberg, 2010), pp. 35–59.
19. J. G. K. Wehrle and M. Günes, *Modeling and Tools for Network Simulation* (Springer, 2005).
20. A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte and S. Valentin, Simulating wireless and mobile networks in OMNeT++ the MiXiM vision, in *Proc. of the 1st Int. Conf. on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops SIMUTools '08* (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2008), pp. 71:1–71:8.
21. K. Price, R. M. Storn and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization* (Springer Science & Business Media, 2006).
22. S.-F. Kuo, G. P. Merkley and C.-W. Liu, Decision support for irrigation project planning using a genetic algorithm, *Agricultural Water Management* **45**(3) (2000) 243–266.
23. H. Muhlenbein, Evolution in time and space — The parallel genetic algorithm, in *Foundations of Genetic Algorithms* (1991).
24. K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* **6**(2) (2002) 182–197.