# A HYBRID IWO/PSO ALGORITHM FOR FAST AND GLOBAL OPTIMIZATION

Hossein Hajimirsadeghi, *Student Member, IEEE*, Caro Lucas, *Member, IEEE*

***Abstract:*** **This paper presents a hybrid optimization algorithm which originates from Invasive Weed Optimization (IWO) and Particle Swarm Optimization (PSO). Based on the novel and distinct qualifications of IWO and PSO, we introduce IWO/PSO algorithm and try to combine their excellent features in this extended algorithm. The efficiency of this algorithm both in the case of speed of convergence and optimality of the results are compared with IWO, PSO, and some other evolutionary algorithms through a number of common multi-dimensional benchmark functions. Finally, a practical problem consisting design and optimization of an adaptive controller for a surge tank is simulated. The experimental results show that the proposed algorithm can be successfully employed as a fast and global optimization method for a variety of theoretical or practical purposes.**

***Index Terms*: Evolutionary Algorithms, Invasive Weed Optimization, Particle Swarm Optimization, Biomimicry.**

## I. INTRODUCTION

Recently there has been considerable amount of attention devoted to bioinspiration and biomimicry, for solving computational problems and constructing intelligent systems, including autonomous robots [1], [2], automated fabrication devices, smart structures, and also developing intelligent control strategies like distributed and cooperative control [3], [4], formation control for multi-agent systems [5], [6], and attention control [7]. Reviewing these achievements in the scope of computational intelligence suggests that there seems to be at least six main domains of intelligence in biological systems and wild life: swarming, Communication and collaboration, Reproduction and Colonization, Learning and Experience, Competition, Evolution.

There are many evidences of intelligence for the posed domains in animals, plants, and generally living systems. For example, ants foraging, birds flocking, fish schooling, bacterial chemotactics are some of the well-known examples in category of swarming [8]-[14]. Communication is present among most animals and plants. For example, the way honey bees share their information about the quality of nutrients makes one of the fastest mediums of communication in collective systems [15], [16]. Reproduction is also existed in all the creatures; However, some show an intellectual mechanism for breeding or reproducing like making fruiting body in some kinds of bacteria [17]. In plants and trees, roots and branches have deliberate and intelligent attitude toward colonization. Generally speaking, where the resources are abundant, branching increases and bushy structures are constructed, and where the nutrients are insufficient, growth is accelerated and branching decreases, so the density in the rich places becomes larger [18]. The same process is available in weeds accompanied by $r$ and $k$ selection approach [19], [20]. Competition is mostly a result of deficiency and can be seen in all species whether in the form of intraspecific or interspecific competition. Human beings and some other creatures are capable of cognitive learning and experience recording. For example, in lack of nutrients, a kind of bacteria is producing extra copies of itself as memento to remember the situation; thus, if shortage recurs later, the bacterium is better prepared [21]. Finally, we can see numerous evidences of evolution in nature that has made our present universe.

In this paper, we aim to merge the idea of intelligent swarming, social cooperation, competition, and reproduction in an optimization meta-algorithm. Particle Swarm Optimization is inspired from flocking birds (swarming and collaborative communication) and has been used in a large number of applications like neural network training [22], data mining [23], web content organizing [24], computing Nash equilibria in strategic games [25], etc. Invasive Weed Optimization is a novel ecologically inspired algorithm that mimics the process of weeds colonization and distribution. Despite its recent development, it has shown successful results in a number of practical applications like optimization and tuning of a robust controller [19], optimal positioning of piezoelectric actuators [26], developing a recommender system [27], antenna configuration [28], analysis of electricity markets dynamics [29], etc.

Section II provides steps for algorithm design with a quick review of PSO and IWO, introduction to IWO/PSO algorithm and discussion on parameter settings for the proposed algorithm. In section III, simulation results for optimization of some benchmark problems in comparison with other evolutionary algorithms are summarized. We present our experiment to design an adaptive controller for a surge tank in section IV, and finally, conclusions are drawn in section V.

## II. ALGORITHM DESIGN

### *A. Particle Swarm Optimization*

PSO was developed by kennedy and Eberhart in 1995 [30]. This algorithm aims to mimic foraging trend and communication behavior in flocks of birds when they are flying. Contrary to traditional evolutionary algorithms which only keep track of position, PSO maintains information regarding position and velocity [30]. The equations for



978-1-4244-3861-7/09/$25.00 ©2009 IEEE

calculating the next particle velocity and position are presented in (1) and (2).

$$V_i(t+1) = \omega V_i(t) + c_1 * \varphi_1 * \left(P_i(t) - X_i(t)\right) + c_2 * \varphi_2 * \left(P_g(t) - X_i(t)\right) \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

$P_i$ is the best previous position for that particle, and $P_g$ is the position of the best particle in the whole swarm up to that iteration. $c_1$ and $c_2$ called learning factors, are constants that determine the balance between acceleration toward local best (individual's experience, cognition, exploration) or global best (social collaboration or interaction, exploitation). $\varphi_1$ and $\varphi_2$ are uniform random numbers in the range of [0, 1]. $\omega$ is an inertia weight which determines the influence of velocity memory and is employed on the favor of global or local search [31]. It is also suggested to restrict the velocity to a specified range [-$V_{max}$, $V_{max}$] [32]. Until now, numerous versions of PSO with selection, reproduction, recombination, and mutation operators have been developed and the way on the improvement of PSO and generally swarm intelligence seems to be continued.

## B. *Invasive Weed Optimization*

Invasive weed optimization was developed by Mehrabian and Lucas in 2006 [19]. IWO algorithm is a bioinspired numerical optimization algorithm that simply simulates natural behavior of weeds in colonizing and finding suitable place for growth and reproduction. Some of the distinctive properties of IWO in comparison with other evolutionary algorithms are the way of reproduction, spatial dispersal, and competitive exclusion [19].

In Invasive Weed Optimization algorithm, the process begins with initializing a population. It means that a population of initial solutions is randomly generated over the problem space. Then members of the population produce seeds depending on their relative fitness in the population. In other words, the number of seeds for each member is beginning with the value of $S_{min}$ for the worst member and increases linearly to $S_{max}$ for the best member. For the third step, these seeds are randomly scattered over the search space by normally distributed random numbers with mean equal to zero and an adaptive standard deviation. The equation for determining the standard deviation (SD) for each generation is presented in (3).

$$\sigma_{iter} = \frac{(iter_{max} - iter)^n}{(iter_{max})^n}\left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \tag{3}$$

where $iter_{max}$ is the maximum number of iterations, $\sigma_{iter}$ is the SD at the current iteration and $n$ is the nonlinear modulation index. The produced seeds, accompanied by their parents are considered as the potential solutions for the next generation. Finally, a competitive exclusion is conducted in the algorithm, i.e., after a number of iterations the population reaches its maximum, and an elimination mechanism should be employed. To this end, the seeds and their parents are ranked together and those with better fitness survive and become reproductive.

## C. *The Hybrid IWO/PSO Algorithm*

From the two previous sections it can be concluded that IWO and PSO have two different approaches for optimization. IWO offers good exploration and diversity, while PSO is an algorithm with fairly deliberate and to the point movements in each iteration. In this section, we mix the two algorithms and present a hybrid algorithm.

In hybrid IWO/PSO algorithm, colonization is beginning in the same way as IWO, however, the seeds are located like the equations in PSO for flying particles. It means that after reproducing the seeds, the velocity is updated with (4), and temporary position of seeds is estimated by (5), and finally these seeds are randomly distributed the same as the process used in IWO to construct the next population. Pseudo-code for this algorithm is presented in Fig. 1.

$$V_{i,s}(t+1) = \omega \,.\, V_i(t) + c_1 \,.\, \varphi_{1,s} \,.\, \left(P_i(t) - X_i(t)\right) + c_2 \,.\, \varphi_{2,s} \,.\, \left(P_g(t) - X_i(t)\right) \tag{4}$$

$$X_{i,s}(t+1) = X_i(t) + V_{i,s}(t+1) \tag{5}$$

1. Genearte random population of $N_0$ solutions;
2. For $iter$ = 1 to the maximum number of generations;
   a. Calculate fitness for each individual;
   b. Compute maximum and minimum fitness in the colony;
   c. Set $P_g$ as the best position of all individuals;
   d. For each individual $w \in W$;
      i. Set $P_i$ as the best position of individual $w$ in comparison with its predecessors;
      ii. Compute number of seeds of $w$, corresponding to its fitness;
      iii. For each seed $s$;
         1) Calculate the velocity according to (4);
         2) Update the position according to (5);
      iv. Randomly distribute generated seeds over the search space with normal distribution around the parent plant ($w$);
      v. Add the generated seeds to the solution set, $W$;
   e. If ($|W| = N) > p_{max}$;
      i. Sort the population $W$ in descending order of their fitness;
      ii. Truncate population of weeds with smaller fitness until $N = p_{max}$;
3. Next $iter$;

Fig. 1 – Psuedocode for IWO/PSO algorithm

$V_{i,s}$ and $X_{i,s}$ are the velocity and position for the $s$th seed of the $i$th member. It is important to say that random distribution of the seeds is still necessary in this algorithm. Indeed, we tried the algorithm without this dispersal step and the algorithm failed to achieve acceptable rates of success mostly because of inability in escaping from local optima. The reason is that in IWO, colonization imposes considerable amount of exploitation, and employing PSO and IWO without adding any exploration operator like random distribution of seeds decreases the diversity in the algorithm. Moreover, there is another possible approach for distribution of the seeds in which the seeds are distributed randomly and then the movements are performed by (4) and (5). We tried this approach either, and unsatisfactory results were obtained. In fact, all the seeds after being randomly distributed move toward the best members, and so this algorithm doesn't help to increase the diversity, and it is easily trapped in local optima.

In our hybrid algorithm, we consider two cases for reproduction. In IWO, the number of seeds for each plant increases linearly proportionate to its fitness in the population. Although this mechanism seems intuitive at first, but there is no strong and perfect reasoning for that, especially when there is a competitive exclusion in the next step, i.e., considering the elimination process applying to the plants with worse fitness, these individuals must have more chance to produce offspring with better fitness to survive. This approach can also be observed in nature among bacteria when they start the process of sporulation in nutritional stress as a survival mechanism [12], [14], [33]. This view makes more sense when a straight forward technique for fitness improvement and surviving is existed by following the best individuals in the population which is the procedure provided in IWO/PSO algorithm.

To use the best approach in our proposed algorithm, an experiment is performed in which the two above-mentioned cases are compared. The results for this experiment are presented in Table 1. The first case is the mechanism that the number of seeds and the fitness are proportionate and the second case is when they are disproportionate. Table 1 shows that the second case outperforms the first case in optimality of the solutions. As the number of function evaluation for the two cases with the same parameters are different for some of the instances in this experiment (i.e., evaluation number for the second case is greater than the first case's), comparison between the two cases might be imperfect. So, Table 1 also provides some other instances with nearly equal evaluation number to make logical inferences.

The transient behavior of the two cases might help us to have better understanding of their routine. Evolutionary process for these two cases is demonstrated in Fig. 2. Note that the average number of function evaluation for both cases is roughly set to be equal in this experiment to have a fair comparison. It can be observed that convergence of the first approach is faster at first, however in the vicinity of the optimal point, the second approach has better progress. The reason for this behavior might be as follows: in the first case, colonization around the best solutions helps to have bigger steps toward the optimal area at first generations, while in the optimal area, immediate exclusion of the worst individuals decreases exploration in this area, degrades diversity and leads to immature convergence.

Table 1. Comparison Between the two Cases for Reproduction Process

| $P_{max}$ | $it_{max}$ | $S_{min}$ | $S_{max}$ | The 1st case | | The 2nd case | |
|---|---|---|---|---|---|---|---|
| | | | | Mean | Eval. Num. | Mean | Eval. Num. |
| 40 | 300 | 1 | 3 | 0.0138 | 16616 | 0.0060 | 20166 |
| 50 | 300 | 1 | 3 | 0.0130 | 20515 | - | - |
| 40 | 300 | 1 | 4 | 0.0176 | 20411 | - | - |
| 60 | 200 | 1 | 3 | 0.0137 | 16161 | 0.0076 | 20411 |
| 40 | 200 | 1 | 3 | 0.0191 | 10947 | 0.0088 | 13518 |
| 50 | 200 | 1 | 3 | 0.0165 | 13566 | - | - |
| 40 | 200 | 1 | 4 | 0.0184 | 13468 | - | - |
| 20 | 200 | 1 | 3 | 0.0228 | 5692 | 0.012 | 6622 |
| 60 | 100 | 1 | 3 | 0.0369 | 7893 | 0.0264 | 9738 |
| 60 | 300 | 1 | 3 | 0.0183 | 24630 | 0.0100 | 29808 |
| 60 | 200 | 0 | 3 | 0.0240 | 10530 | 21.94 | 438 |
| 60 | 200 | 0 | 5 | 0.0382 | 18110 | 0.0117 | 33607 |
| 20 | 200 | 0 | 5 | 0.0512 | 6649 | 0.0575 | 10492 |
| 60 | 300 | 0 | 3 | 0.0324 | 16379 | 22.75 | 735 |

## D. Parameter Settings

Despite the numerous advantages of IWO/PSO algorithm, it has many parameters to be tuned, comparing with algorithms like PSO. Although our simulations show robustness for parameters selection, however, it seems somehow challenging for a practitioner to set the parameters effectively. In this section, we present some guidelines for tuning the parameters and compare them with those in PSO and IWO.

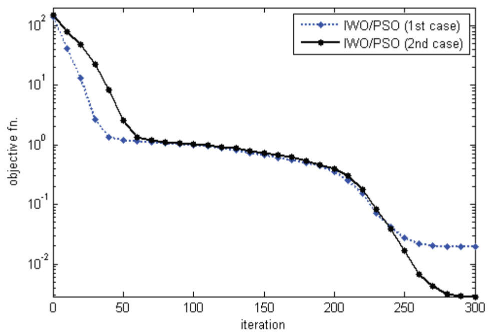


Fig. 2 – Comparison between two approaches for computing the number of seeds (the results are averaged over 20 runs):
Number of seeds is proportionate to fitness (1st case).
Number of seeds is disproportionate to fitness (2nd case)

Firstly, in PSO it is suggested that $c_1$ and $c_2$ are about 2. However, for IWO/PSO this estimation doesn't work. In fact, in IWO/PSO, there are two ways for tracking the global best particle, whether by increasing $S_{max}$ or by increasing $c_2$. So for escaping from local optima and increasing the exploration, we suggest $c_2$ to be much less than $c_1$ ($c_1 \epsilon\, [1.5, 2]$,

$c_2 \epsilon$ [0, 0.5]). Furthermore, comparing with IWO, in IWO/PSO we have more deliberate strategy (not just a random distribution as in IWO) to reproduce the next population so we can have more number of seeds in each iteration and also we can start the algorithm with a smaller standard deviation. In addition, as described in previous section, in our proposed algorithm, number of seeds is disproportionate to fitness in reproduction process, so the trace of some valuable particles may vanish after a number of iterations. To come up with this problem, we suggest that $S_{min}$ is set to 1, so each particle will have a representative in next population that is probably better than itself. Finally, like IWO, speed of convergence for the algorithm can be set by $n$ or $iter_{max}$. But, although increase of $n$ or decrease of $iter_{max}$ makes a fast convergence, however the possibility of trapping in local optima also increases.

## III. SIMULATION STUDIES

### *A. Comparing IWO/PSO with Five Recent Evolutionary Algorithms*

To evaluate performance of the proposed algorithm, Griewank function (6) is considered as a benchmark and the simulation results are compared with results of five evolutionary algorithms (EAs): Particle Swarm Optimization (PSO), Invasive Weed Optimization (IWO), Genetic Algorithms (GAs), Memetic Algorithms (MAs), and Shuffled Frog Leaping (SFL) Which were reported in [19] and [34]. As it is mentioned in [19], it was tried to find the best possible parameters for the above-mentioned algorithms to have a reasonable comparison and providing guidelines for determining the best operators for each algorithm.

$$f(X) = 1 + \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} (\cos(x_i/\sqrt{i})) \qquad (6)$$

Simulation results for Griewank function with 10 and 20 dimensions which are declared to be more complex than the other standard dimensions of this function are provided in Table 2. It shows that the hybrid algorithm outperforms all the other algorithms, including its components which are IWO and PSO. In simulation of IWO/PSO and IWO, the process stopped when maximum allowable iteration was reached, but for other EAs, a different termination criterion was used in [34]. The parameters for the best results, i.e., one with the best solution and one with less evaluation number but a suboptimal solution are listed in Table 3.

To illustrate the transient performance of IWO/PSO algorithm, trace of mean objective values of Griewank10 for 20 runs is depicted in Fig. 3. Obtaining Comparable results, parameters for all three algorithms are tuned so that the evaluation number of the objective function is roughly the same for all of them and is equal to ~20000.

Table 2. Results of the Griewank Function for Comparison with 5 EAs

| **Comparison Criteria** | **Algorithm** | **$P_{max}$** | **$iter_{max}$** | **dim 10** | **dim 20** |
|---|---|---|---|---|---|
| **% success** | IWO/PSO | 20 | 200 | **100** | 80 |
| | IWO/PSO | 40 | 300 | **100** | **100** |
| | IWO | 20 | 120 | 95 | - |
| | IWO | 20 | 200 | - | 95 |
| | PSO | - | - | 30 | 80 |
| | GAs (Evolver) | - | - | 50 | 30 |
| | MAs | - | - | 90 | 100 |
| | SFL | - | - | 50 | 70 |
| **Mean Solution** | IWO/PSO | 20 | 200 | 0.012 | 0.0365 |
| | IWO/PSO | 40 | 300 | **0.006** | **0.0087** |
| | IWO | 20 | 210 | 0.0163 | - |
| | IWO | 20 | 200 | - | 0.0494 |
| | PSO | - | - | 0.093 | 0.081 |
| | GAs (Evolver) | - | - | 0.06 | 0.097 |
| | MAs | - | - | 0.014 | 0.013 |
| | SFL | - | - | 0.08 | 0.06 |

Table 3. IWO/PSO parameter values for Griewank Function Optimization

| Symbol | Quantity | Value: Eval. Num. ≅20000 | Value: Eval. Num. ≅6500 |
|---|---|---|---|
| $N_0$ | Number of initial population | 5 | 5 |
| $iter_{max}$ | Maximum number of iterations | 300 | 200 |
| $dim$ | Problem dimension | 10, 20 | 10, 20 |
| $P_{max}$ | Maximum number of plant | 40 | 20 |
| $S_{max}$ | Maximum number of seeds | 3 | 3 |
| $S_{min}$ | minimum number of seeds | 1 | 1 |
| $n$ | Nonlinear modulation index | 3 | 3 |
| $\sigma_{init}$ | Initial value of standard deviation | 30 | 30 |
| $\sigma_{final}$ | Final value of standard deviation | 0.005, 0.01 | 0.01 |
| $x_{ini}$ | Initial search area | [-512,511] | [-512,511] |
| $c_1$ | Acceleration coefficient 1 | 0.5 | 0.5 |
| $c_2$ | Acceleration coefficient 1 | 2 | 2 |
| $\omega$ | Inertia weight | 0.7 | 0.7 |
| $V_{max}$ | Maximum velocity | 20 | 20 |

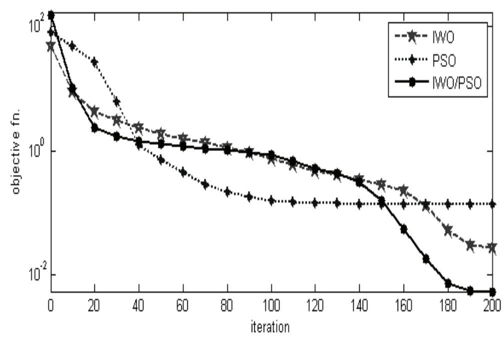


Fig. 3 – Optimization process of the Griewank10 for IWO, PSO, and IWO/PSO

### B. *Comparing IWO/PSO with Optimized PSO and Fast PSO*

For evaluating the computational cost and also the optimality of the proposed algorithm another simulation is performed. In this process, IWO/PSO is used to optimize another benchmark, Rastrigin function (7), which is known as a challenging benchmark. Results are compared with those in [35] and [36] which were pursuing for optimized and fast PSO, respectively. In [35], an Optimized Particle Swarm Optimization (OPSO) is introduced in which the optimal parameters of the swarms are calculated with a number of super swarms. The results of our simulation and the ones reported in [35] are provided in Table 4. It is shown that IWO/PSO has much better results than Standard type PSO, Construction type PSO, and even Optimized PSO. In [35], the number of particles was set to 20 and the maximum number of iterations was set to 1000, so the maximum evaluation number of the objective function was 20000. For this reason, the parameters of IWO/PSO in our simulation are tuned to have approximately the same number of function evaluation. The parameter settings for this experiment are provided in Table 5.

$$f(X) = \sum_{i=1}^{N} (x_i^2 - 10\cos(2\pi x_i) + 10) \tag{7}$$

Table 4. Simulation Results of Rastrigin30 Optimization for comparison with SPSO, CPSO, and OPSO

| Method | Mean error | Standard deviation | Median error | Eval. Num. | Success % |
|---|---|---|---|---|---|
| Standard type PSO (SPSO) | 99.5 | 27 | 98.2 | 20000 | 55 |
| Construction type PSO (CPSO) | 86.2 | 23 | 84.6 | 20000 | 70 |
| OPSO | 46.5 | 13.1 | 44.8 | 20000 | 100 |
| IWO/PSO | 31.55 | 8.59 | 31.19 | 19189 | 100 |

Table 5. IWO/PSO parameters value for comparison with SPSO, CPSO, and OPSO

| Symbol | value | Symbol | value |
|---|---|---|---|
| $N_0$ | 10 | $\sigma_{final}$ | .001 |
| $it_{max}$ | 100 | $x_{ini}$ | [-5.12,5.12] |
| $P_{max}$ | 60 | $c_1$ | 0.5 |
| $S_{max}$ | 5 | $c_2$ | 2 |
| $S_{min}$ | 1 | $\omega$ | 0.7 |
| n | 3 | $V_{max}$ | 4 |
| $\sigma_{init}$ | 1 | - | - |

In [36], a new Fast Particle Swarm Optimization (FPSO) is proposed in which traditional and additional convex combination methods are used to estimate objective values of the child swarms, decreasing the evaluation number for convergence and hence speeding up the optimization process. Here, we compare the results of IWO/PSO algorithm, both in the case of objective value and average number of function evaluation for Rastrigin function with those reported in [36]. The simulation results of Rastrigin function with 30 dimensions are depicted in Table 6, and the IWO/PSO parameters for this study are shown in Table 7. From Table 6 it can be seen that with approximately the same number of function evaluation, IWO/PSO outperforms SPSO and also different types of FPSO.

## IV. ADAPTIVE CONTROL OF A SURGE TANK

In [19], the authors introduced Invasive Weed Optimization as a robust algorithm and showed its performance to design a robust controller. As discussed in former sections, IWO/PSO algorithm is a fast and efficient algorithm in which the parameters can be set so that a fairly acceptable fitness is achieved after a limited number of iterations. This characteristic suggests an outstanding mechanism for online processing and computation, and so it can be employed for fast identification and adaptation in dynamic systems. Biomimicry of bacterial foraging to estimate models in adaptive control was discussed in [8]. In this section, we design the same controller as reported in [8], using IWO/PSO for tuning the multiple identifier models [37], and demonstrate the efficiency of IWO/PSO algorithm for searching in the parameter space both in the case of optimality and speed.

Table 6. Simulation results of Rastrigin30 Optimization for comparison with SPSO and FPSO

| Algorithm | Mean | Std | Eval. Num. | Addi Num. | Trad Num. |
|---|---|---|---|---|---|
| Standard type PSO | 31.10 | 8.98 | 500000 | 0 | 499800 |
| IWO/PSO | 16.29 | 4.25 | 498160 | 0 | 0 |
| FPSO3 | 29.42 | 7.53 | 105043 | 99165 | 400635 |
| FPSO4 | 28.26 | 8.31 | 98105 | 70653 | 429147 |
| IWO/PSO | 23.52 | 5.69 | 98682 | 0 | 0 |

Table 7. IWO/PSO parameters value for comparison with SPSO and FPSO

| Symbol | Value | | Symbol | Value | |
|---|---|---|---|---|---|
| | Eval. Num.= 490151 | Eval. Num.= 97381 | | Eval. Num.= 490151 | Eval. Num.= 97381 |
| $N_0$ | 10 | 10 | $\sigma_{final}$ | .001 | .001 |
| $it_{max}$ | 400 | 200 | $x_{ini}$ | [-5.12,5.12] | [-5.12,5.12] |
| $P_{max}$ | 300 | 150 | $c_1$ | 0.5 | 0.5 |
| $S_{max}$ | 6 | 5 | $c_2$ | 2 | 2 |
| $S_{min}$ | 1 | 1 | $\omega$ | 0.7 | 0.7 |
| n | 3 | 3 | $V_{max}$ | 4 | 4 |
| $\sigma_{init}$ | 1 | 1 | - | - | - |

Equation (8) shows the nonlinear model of a surge tank for liquid level control. $h(t)$ is liquid level (saturated), $u(t)$ is the input (saturated), c and d are the constants and $A\big(h(t)\big)= \ |ah(t)+b|$ is the unknown tank cross-sectional area in which $a$ and $b$ are also constants. Each identifier model is an affine mapping (i.e., $\alpha(t)+\beta(t).u(t)$) that matches plant nonlinearities. Using IWO/PSO, the identifier model parameters represent the particle's position. The cost function is defined to be the sum of squares of N=100 past identifier error (error between the model output and the plant output), for each identifier model.

$$\frac{dh(t)}{dt}=\frac{-d\sqrt{2gh(t)}}{A(h(t))}+\frac{c}{A\big(h(t)\big)}u(t) \tag{8}$$

At each step, the best model is selected and used in a standard certainty equivalence approach for controller design. This approach is similar to indirect genetic adaptive control approaches [38], and direct adaptive control strategies can be investigated using the mechanism in [39] and [40]. Considering the discretized model of the system, we use our algorithm to optimize the cost function over the time like the process described in [8].

The tracking performance and the best cost are shown in Fig. 4 and Fig. 5. In Fig. 4, a small population is recruited (approximately equal to the population size in [8]) for fast adaptation. It can be observed that acceptable tracking is achieved after only ten seconds (100 iteration as the sampling time is 0.1 sec), and also the cost decreases over the whole process. In Fig. 5, more individuals are used and so complexity of the process is increased, but it is evident that precision and accuracy of the estimator and the controller is much better than the previous experiment. Note that the results for the simulation in [8] are also depicted in Fig. 6.

## V. CONCLUSION

IWO/PSO is a numerical stochastic optimization algorithm which comprises some intelligent natural behaviors of biological systems like swarming, collaborative communication, colonization, and competition. The efficiency of IWO/PSO is surveyed by comparing it with recent evolutionary algorithms through a set of well-known multi-dimensional benchmark functions. The simulations indicate that the proposed algorithm has outstanding performance in speed of convergence and precision of the solution for global optimization, i.e., it has the capability to come up with non-differentiable objective functions with a multitude number of local optima in a reasonable time limit. Finally, IWO/PSO is employed for optimization of an engineering problem and its applicability is demonstrated in excellent estimation of the parameters for identifier models and accurate control of the system.

Based on fast convergence of this algorithm and its ability for switching between exploration and exploitation, it is suggested to use this algorithm for machine learning and other applications that need real-time processing of the information.

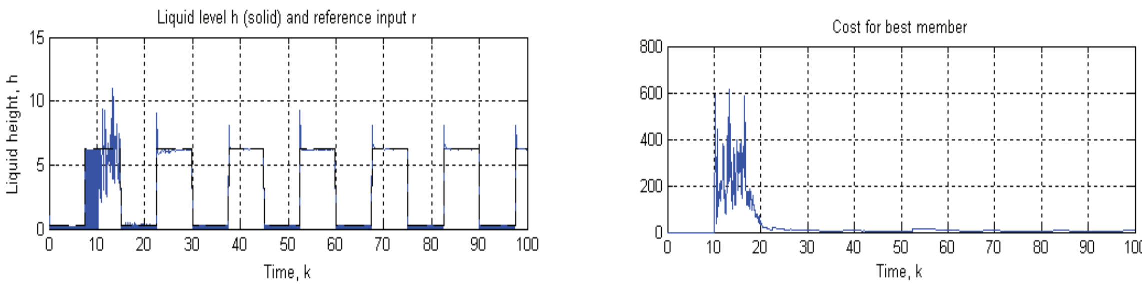


Fig. 4 – IWO/PSO for adaptive control of a surge tank with a small population size

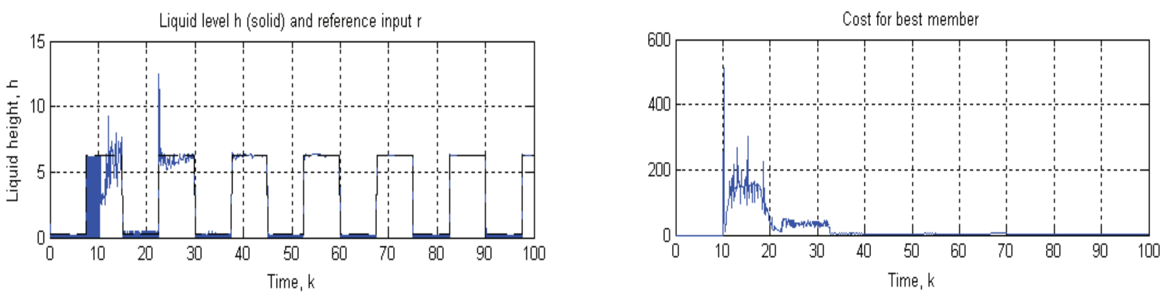


Fig. 5 – IWO/PSO for adaptive control of a surge tank with a greater population size

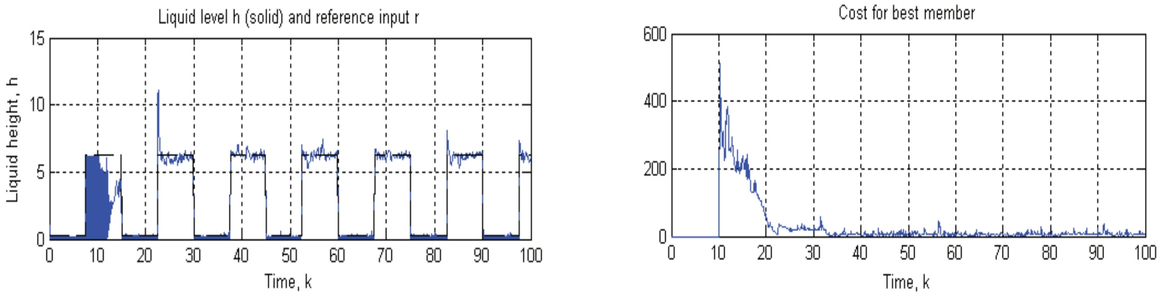


Fig. 6 – Results of the experiment in [8], using Bacterial foraging Algorithm.

## REFERENCES


[1] I. Suzuki and M. Yamashita, “Distributed anonymous mobile robots: formation of geometric patterns,” *SIAM J. Comput.*, vol. 28, no. 4, pp.1347–1363, 1999.

[2] B. W. Andrews, K. M. Passino, and T. A. Waite, “Foraging Theory for Autonomous Vehicle Decision-making System Design,” *J. of Intelligent Robot Syst*, vol.49, pp. 39–65, Mar. 2007.

[3] I. Suzuki and M. Yamashita, “Distributed anonymous mobile robots: formation of geometric patterns,” *SIAM J. Comput.*, vol. 28, no. 4, pp. 1347–1363, 1999.

[4] N. Quijano, A. E. Gil, and K. M. Passino, “Experiments for dynamic resource allocation, scheduling, and control,” *IEEE Control Syst. Mag.*, vol. 25, no.1, pp. 63–79, 2005.

[5] F. Giulietti, L. Pollini, and M. Innocenti, “Autonomous formation flight,” *IEEE Control Syst. Mag.*, vol. 20, pp. 34–44, Dec. 2000.

[6] T. Balch and R. C. Arkin, “Behavior-based formation control for multirobot teams,” *IEEE Trans. Robot. Automat.*, vol. 14, pp. 926–939, Dec. 1998.

[7] H. Fatemi Shariatpanahi and M. Nili Ahmadabadi, “Biologically Inspired Framework for Learning and Abstract Representation of Attention Control,” *in Proc. 4th International Workshop on Attention in Cognitive Systems*, Hyderabad, India, 2007, pp 63–80.

[8] K. M. Passino, “Biomimicry of Bacterial Foraging for Distributed Optimization and Control,” *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, June 2002.

[9] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford Univ. Press, 1999.

[10] A. Okubo, “Dynamical aspects of animal grouping: swarms, schools, flocks, and herds,” *Adv. Biophys.*, vol. 22, pp. 1–94, 1986.

[11] S. Gueron and S. A. Levin, “The dynamics of group formation,” *Math Biosci.*, vol. 128, pp. 243–264, 1995.

[12] E. Ben-Jacob, “Bacterial self-organization: co-enhancement ofcomplexification and adaptability in a dynamic environment,” *Phil. Trans. R. Soc. Lond. A*, vol. 361, pp. 1283–1312, May 2003.

[13] E. Ben-Jacob and H. Levine, “Self-engineering capabilities of bacteria,” Journal of The Royal Society Interface, 3 (6), pp. 197–214, 2006.

[14] E. Ben-Jacob, I. Becker, Y. Shapira, and H. Levine, “Bacterial linguistic communication and social intelligence,” *TRENDS in Microbiology*, vol.12 no.8 August 2004.

[15] A. Trewavas, “Aspects of Plant Intelligence,” *Annals of Botany*, vol. 92, pp. 1–20, 2003.

[16] K. M. Passino, and T.D. Seeley, “Modeling and Analysis of Nest-Site Selection by Honey Bee Swarms: The Speed and Accuracy Trade-off,” *Behavioral Ecology and Sociobiology*, vol. 59, no. 3, pp. 427–442, Jan. 2006.

[17] N. F. Britton, N. R. Franks, S. C. Pratt, and T. D. Seeley, “Deciding on a new home: how do honey bees agree?” in *Proc. Roy. Soc. Lond. (B)*, vol. 269, pp. 1383–1388, 2002.

[18] A. Stevens, “A stochastic cellular automaton, modeling gliding and aggregation of myxobacteria,” *SIAM J. Appli. Math.*, vol. 61, no. 1, pp. 172–182, 2000.

[19] A. R. Mehrabian and C. Lucas, “A novel numerical optimization algorithm inspired from weed colonization,” *Ecological Informatics*, vol. 1, pp. 355–366, 2006.

[20] R. Dawkins, *The extended phenotype: the long reach of the gene.* Oxford University Press, Oxford, 1999 (First edition, 1982), pp. 156–164.

[21] S. M. Hoffer, H. V. Westerhoff, K. J. Hellingwerf, P. W. Postma, and J. Tommassen, “Autoamplification of a Two-Component Regulatory System Results in Learning Behavior,” *J. of Bacteriology*, vol. 183, no. 16, pp. 4914–4917, Aug. 2001.

[22] R. C. Eberhart. and Y. Shi, “Evolving artificial neural networks,” in *Proc. the 1998 International Conference on Neural Networks and Brain*, Beijing, China, 1998, pp. 5–13.

[23] Q. Y. Li, Z. P. Shi, J. Shi, and Z. Z. Shi, “Swarm intelligence clustering algorithm based on attractor,” *LNCS 3612*, Changsha, China, 2005, pp. 496–504.

[24] S. Hassas, “Using swarm intelligence for dynamic web content organizing,” in *Proc. the 2003 IEEE Swarm Intelligence Symposium, Indianapolis*, USA, 2003, pp.19–25.

[25] N.G. Pavlidis, K.E. Parsopoulos, and M.N. Vrahatis, “Computing Nash equilibria through computational intelligence methods,” J. of Computational and Applied Mathematics, vol. 175, no. 1, pp. 113–136, 2005.

[26] A. R. Mehrabian and A. Yousefi-Koma, “Optimal positioning of piezoelectric actuators of smart fin using bio-inspired algorithms,” *Aerospace Science and Technology*, vol. 11, pp. 174–182, 2007.

[27] H. Sepehri-Rad and C. Lucas, “A recommender system based on invasive weed optimization algorithm,” in *Proc. IEEE Congress on Evolutionary Computation*, 2007, pp. 4297–4304.
[28] B. Dadalipour, A. R. Mallahzadeh, Z. Davoodi-Rad, “Application of the invasive weed optimization technique for antenna configurations,” in *Proc. Loughborough Antennas and Propagation Conf.*, Loughborough, pp. 425-428, Mar. 2008.
[29] M. Sahraei-Ardakani, M. Roshanaei, A. Rahimi-Kian, C. Lucas, “A Study of Electricity Market Dynamics Using Invasive Weed Optimization,” in *Proc. IEEE Symposium on Computational Intelligence and Games*, Perth, Australia, Dec. 2008.
[30] J. Kennedy and R.C. Eberhart, “Particle swarm optimization,” in *Proc. IEEE Int. Conference on Neural Networks,* Piscataway, NJ, pp. 1942–1948, 1995.
[31] Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” *in Proc. IEEE Int. Conference on Evolutionary Computation*, Piscataway, NJ, pp 69–73, 1998.
[32] R.C. Eberhart, P. Simpson, and R. Dobbins, *Computational intelligence PC tools*. Academic Press Professional, San Diego, CA, 1996, pp. 212–226.
[33] C. Stephen, “Bacterial sporulation: aquestion of commitment?” *Curr. Biol*., vol. 8, pp. 45–48, 1998.
[34] E. Elbeltagia, T. Hegazyb, and D. Grierson, “Comparison among five evolutionary-based optimization algorithms,” *Advanced Engineering Informatics*, vol. 19, pp. 43–53, 2005.
[35] M. Meissner, M. Schmuker, and G. Schneider, “Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training,” *BMC Bioinformatics*, vol. 7, no. 125, 2006.
[36] Z. Cui1, J. Zeng, and G. Sun, “A Fast Particle Swarm Optimization,” *Int. J. of Innovative Computing, Information and Control*, vol. 4, no. 6, pp. 1365–1380, 2006
[37] R. Murray-Smith, T.A. Johansen, Eds., *Multiple Model Approaches to Nonlinear Modeling and Control*. London, UK: Taylor and Francis, 1996.
[38] K. Kristinsson and G. Dumont, “System identification and control using genetic algorithms,” *IEEE Trans. Syst., Man, Cybernet.*, vol. 22, no. 5, pp. 1033-1046, 1992.
[39] L.L. Porter and K. Passino, “Genetic adaptive and supervisory control,” *Int. J. Intell. Contr. Syst.*, vol. 12, no. 1, pp. 1-41, 1998.
[40] W. Lennon and K. Passino, “Genetic adaptive identification and control,” *Eng. Applicat. Artif. Intell.*, vol. 12, pp. 185-200, Apr. 1999.

**Hossein Hajimirsadeghi** received the B.S. degree in electrical engineering from University of Tehran, Iran, in 2008. He is currently pursuing the M.S. degree with the Center of Excellence for Control and Intelligent Processing at the Department of Electrical and Computer Engineering, University of Tehran, Iran. His research interests include biomimicry, artificial life, computational intelligence, multi-agent systems, machine learning, cooperative and distributed control, ambient intelligence, and game theory with application in control systems, aerospace, robotics, economics, and cognitive sciences.

**Caro Lucas** received the M.S. degree from the University of Tehran, Iran, in 1973, and the Ph.D. degree from the University of California, Berkeley, in 1976. He is a Professor, and the Director of Center of Excellence for Control and Intelligent Processing at the Department of Electrical and Computer Engineering, University of Tehran, Iran, as well as a Researcher at the School of Cognitive Science, Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran. He has served as the Director of SIS (1993-1997), Chairman of the ECE Department at the University of Tehran (1986-1988), Managing Editor of the Memories of the Engineering Faculty, University of Tehran (1979-1991), Reviewer of Mathematical Reviewers (since 1987), Associate Editor of Journal of Intelligent and Fuzzy systems (1992-1999), and Chairman of the IEEE, Iran section (1990-1992). He was also a Visiting Associate Professor at the University of Toronto (summer, 1989-1990), University of California, Berkeley (1988-1989), an Assistant Professor at Garyounis University (1984-1985), University of California at Los Angeles (1975-1976), a Senior Researcher at the International Center for Theoretical Physics and the International Center for Genetic Engineering and Biotechnology, both in Trieste, Italy, the Institute of Applied Mathematics Chinese Academy of Sciences, Harbin Institute of Electrical Technology, a Research Associate at the Manufacturing Research Corporation of Ontario, and a Research Assistant at the Electronic Research Laboratory, University of California, Berkeley. His research interests include biological computing, computational intelligence, uncertain systems, intelligent control, neural networks, multi-agent systems, data mining, business intelligence, financial modeling and knowledge management. Professor Lucas has served as the chairman of several International Conferences. He was the founder of the SIS and has assisted in founding several new research organizations and engineering disciplines in Iran. He is the recipient of several research grants at the University of Tehran and SIS.