Research Article

Tuning PID Controller Using Multiobjective Ant Colony Optimization

Ibtissem Chiha,[1] Noureddine Liouane,[2] and Pierre Borne[3]

[1] *Ensi Enim, Monastir, Rabat, Tunisia*
[2] *Ensi Enim, Tunisia*
[3] *National Cancer Institute, 92513 Boulogne Billan Court Cedex, France*

Correspondence should be addressed to Ibtissem Chiha, chiha.ibtissem@yahoo.fr

This paper treats a tuning of PID controllers method using multiobjective ant colony optimization. The design objective was to apply the ant colony algorithm in the aim of tuning the optimum solution of the PID controllers (K_p, K_i, and K_d) by minimizing the multiobjective function. The potential of using multiobjective ant algorithms is to identify the Pareto optimal solution. The other methods are applied to make comparisons between a classic approach based on the "Ziegler-Nichols" method and a metaheuristic approach based on the genetic algorithms. Simulation results demonstrate that the new tuning method using multiobjective ant colony optimization has a better control system performance compared with the classic approach and the genetic algorithms.

1. Introduction

Proportional-integral-derivative (PID) controllers are frequently used in the control process to regulate the time domain behavior of many different types of dynamic plants. These controllers are extremely popular because of their simple structure and they can usually provide a good closed loop response characteristic. Despite its simple structure it seems so hard to find a proper PID controller [1]. Considering this problem, various methods have been proposed to tune these parameters.

Ziegler-Nichols tuning method is the most standard one but it is often difficult to find optimal PID parameters with these methods. Therefore many optimization methods are developed to tune the PID controllers such as fuzzy logic [2, 3], neural network [4], neural-fuzzy logic [5], immune algorithm [6], simulated annealing [7], and pattern recognition [8]. In addition, we have many other optimum tuning PID methods based on many random search methods such as genetic algorithm (GA) [9, 10], particle swarm optimization [11], and ant colony optimization [12].

In this work, we developed the problem of design PID controllers as a multiobjective optimization problem taking in consideration the ant colony optimization algorithm (ACO). Researchers have reported the capacity of ACO to efficiently search for and locate an optimum solution. This method was mainly inspired by the fact that ants are able to find the shortest route between their nest and a food source.

Ant colony optimization (ACO) [13, 14] is a recently developed metaheuristic approach for solving hard combinatorial optimization problems such as the travelling salesman problem TSP [15], quadratic assignment problem [16], graph coloring problems [17], hydroelectric generation scheduling problems [18], vehicle routing [19], sequential ordering, scheduling [20], and routing in Internet-like networks [21].

Ant colony optimization algorithms are especially suited for finding solutions to difficult optimization problems. A colony of artificial ants cooperates to find good solutions, which are an emergent property of the ants' cooperative in–teraction. Based on their similarities with ant colonies in nature, ant algorithms are adaptive and robust and can be applied to different versions of the same problem as well as to different optimization problems.

The main traits of artificial ants are taken from their natural model. These main traits are as follows: (1) artificial ants exist in colonies of cooperating individuals, (2) they communicate indirectly by depositing (artificial) pheromone
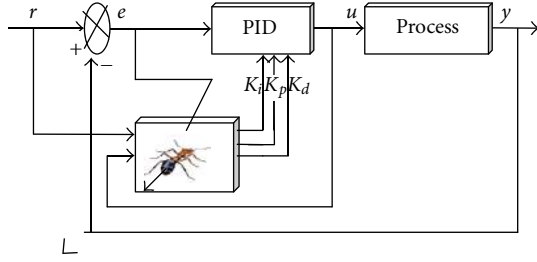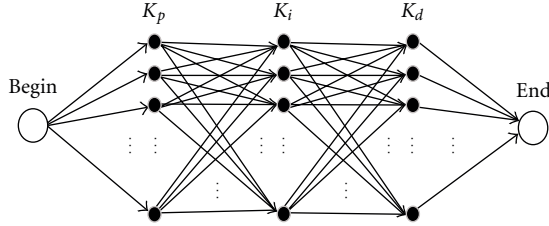
FIGURE 1: PID control system.

FIGURE 2: Ant colony optimization graph.

FIGURE 3: Comparison of step responses of the $G_1(s)$ plant.

(stigmergic communication), (3) they use a sequence of local moves to find the shortest path from a starting position, to a destination point (i.e., the optimal solution to a given problem), and (4) they apply a stochastic decision policy using local information only (i.e., they do not look ahead) to find the best solution. If necessary in order to solve a particular optimization problem, artificial ants have been enriched with some additional capabilities not present in real ants.

An ant colony of finite size searches collectively for a good solution to a given optimization problem. Each individual ant can find a solution or at least part of a solution to the optimization problem on its own but only when many ants work together they can find the optimal solution. Since the optimal solution can only be found through the global cooperation of all the ants in a colony, it is an emergent result of such this cooperation. While searching for a solution the ants do not communicate directly but indirectly by adding pheromone to the environment. Based on the specific problem an ant is given a starting state and moves through a sequence of neighboring states trying to find the shortest path. It moves based on a stochastic local search policy directed by its internal state (private information), the pheromone trails, and local information encoded in the environment (together public information). Ants use this private and public information in order to decide when and where to deposit pheromones. In most applications the amount of pheromone deposited is proportional to the quality of the move an ant has made. Thus, the more pheromone, the better the solution found. After an ant has found a solution, it dies, that is, it is deleted from the system.

This paper is organized as follows: the implementation of ACO to design multiobjective optimization is developed in Section 2. In Section 3, the effectiveness of this approach is tested, compared with other methods which are the standard method (Ziegler-Nichols), metaheuristique method (Genetic Algorithm), and ant system in the same section the
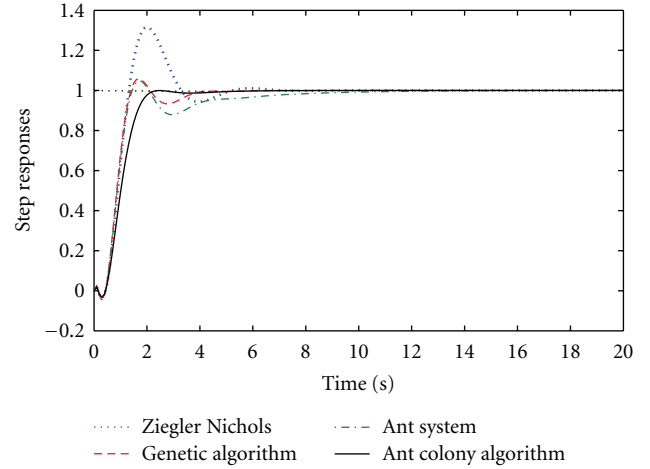
construction of a Pareto optimal set of solutions, and the convergence graph in the ACO method are shown. Finally, some conclusions are given in Section 4.

2. Design PID Controller Using Multiobjective Ant Colony Algorithm

The series controllers are very frequent because of higher order systems. The transfer function of PID controller is defined for a continuous system as:

$$G_c(p) = K_p + \frac{K_i}{s} + K_d s. \tag{1}$$

The design implies the determination of the values of the constants K_p, K_i, and K_d, meeting the required performance specifications.

The textbook version of the PID controller in continuous time is

$$e(t) = r(t) - y(t),$$

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d e(t) = u_p(t) + u_i(t) + u_d(t), \tag{2}$$

where $e(t) = r(t) - y(t)$ is the difference between the reference signal $r(t)$ and the output, $y(t)$ of the controlled process.

The PID controller is implemented to improve the dynamic response in addition to reduce or eliminate the steady-state error. To characterise the performance of the PID controller systems, we compute the indexes' performance of the transient response such as rise time (t_r), overshoot (O_s), settling time (t_s), the integral square error (ISE). . . .

The problem is how to tune the parameters of the PID controllers using the multiobjective ant colony optimization as indicated in Figure 1.

As shown in Figure 3, the gains K_p, K_i, and K_d of the PID controller are generated by the multiobjective ACO algorithm for a given plant.

In order to exploit the ACO algorithm, it would be better to represent our optimization problem by a direct way in the form of construction graph.

The population is represented by $100 * 3$ matrix, where the ant select the optimum parameters K_p, K_i, and K_d of the PID control system by minimizing the objective function L^A. The graph shown in Figure 2 illustrates the design PID problem using ant colony algorithm.

In this study, each parameter of K_p, K_i, and K_d is coded by 100 numbers (nodes), respectively. Therefore, only one node represents the optimum solution values of the parameters K_p, K_i, and K_d.

The basic step in applying optimization method is to choose the optimization criteria that are used to evaluate fitness. Since the PID controller has many indexes performance of the transient response, then we can combine them into one objective function composed of the weighted sum of objectives.

The objective function must be set:

$$L^A = \min(\Phi F), \tag{3}$$

where $F = [f_1 \ f_2 \ f_3 \ f_4 \ f_5 \ f_6 \ f_7]^T$: vector of objective functions, f_1: setting time (t_s), f_2: overshoot (OS), f_3: rise time (t_r), f_4: integral absolute error (IAE), f_5: integral square error (ISE), f_6: integral time absolute error (ITAE), f_7: integral time square error (MSE), and $\Phi = [\lambda_1 \ \lambda_2 \ \lambda_3 \ \lambda_4 \ \lambda_5 \ \lambda_6 \ \lambda_7]$: vector of nonnegative weights.

The goal of multiobjective optimization problems is to find the best compromise between multiple and conflicting objectives. Considering all objectives in these problems, there will be more than one solution that optimizes simultaneously all the objectives and there is no distinct superiority between these solutions. Usually there is not a single best solution being better than the remainder with respect to every objective. Therefore, we face with a set of solutions which are better than remainder solutions called the Pareto front. Among the feasible solutions, solutions belonging to the Pareto front are known as nondominated solutions, while the remainder solutions are known as dominated. Since none of the Pareto set solutions is absolutely better than the other nondominated solutions, all of them are equally acceptable as regards the satisfaction of all the objectives [22].

ACO uses a pheromone matrix $\tau = \{\tau_{ij}\}$ for the construction of potential good solutions. The initial values of τ are set $\tau_{ij} = \tau_0$ for all (i, j), where $\tau_0 > 0$.

The probability $P_{ij}^A(t)$ of choosing a node j at node i is defined in (4). At each generation of the algorithm, the ant constructs a complete solution using (4), starting at source node.

$$P_{ij}^A(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{i,j \in T^A}\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}, \quad \text{if } i, j \in T^A, \tag{4}$$

where η_{ij} representing heuristic functions, α and β are constants that determine the relative influence of the pheromone values and the heuristic values on the decision of the ant, and T^A: is the path effectuated by the ant A at a given time.

The pheromone evaporation is a way to avoid unlimited increase of pheromone trails. Also it allows the forgetfulness of the bad choices:

$$\tau_{ij}(t) = \rho\tau_{ij}(t-1) + \sum_{A=1}^{NA} \Delta\tau_{ij}^A(t), \tag{5}$$

where $\Delta\tau_{ij}^A$ the quantity of pheromone on each path, NA: number of ants, ρ: the evaporation rate $0 < \rho \leq 1$.

Implementation Algorithm. The proposed algorithm can be described by the following general algorithm.

Begin

Step 1. Initialize randomly a potential solutions of the parameters (K_p, K_i, K_d) by using uniform distribution.
   Initialize the pheromone trail, and the heuristic value.
   Initialize the Pareto set to an empty set.

Step 2. Place the Ath ant on the node.
Compute the heuristic value associated in the multiobjective L^A.

   Choose the successive node with probability:

$$P_{ij}^A(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{i,j \in T^A}\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}, \quad \text{if } i, j \in T^A, \tag{6}$$

where $\eta_{ij} = 1/K_j$, $j = [p, i, d]$: representing heuristic functions, T^A: is the path effectuated by the ant A at a given time.

   The quantity of pheromone $\Delta\tau_{ij}^A$ on each path may be defined as:

$$\Delta\tau_{ij}^A = \begin{cases} \dfrac{L^{\min}}{L^A}, & \text{if } i, j \in T^A, \\ 0, & \text{else,} \end{cases} \tag{7}$$

where L^A is the value of the objective function found by the ant A. $L^{\min}$ is the best solution carried out by the set of the ants until the current iteration.

Step 3. Use pheromone evaporation given by (5) to avoid unlimited increase of pheromone trails and allow the forgetfulness of bad choices:

$$\tau_{ij}(t) = \rho\tau_{ij}(t-1) + \sum_{A=1}^{NA} \Delta\tau_{ij}^A(t), \tag{8}$$

where NA: number of ants, and ρ: the evaporation rate $0 < \rho \leq 1$.

Step 4. Evaluate the obtained solutions according to the different objectives.
   Update the Pareto archive with the nondominated ones.
   Reduce the size of the archive if necessary.

Step 5. Display the optimum values of the optimization parameters.

Step 6. Globally update the pheromone, according to the optimum solutions calculated at Step 5.
   Iterate from Step 2 until the maximum of iterations is reached.
End

TABLE 1: Simulation results.

	Ziegle-Nichols [23]	Genetic algorithm [24]	Ant system [12]	Multiobjective ant colony optimization
$G_1(s) = e^{-0.5s}/(s+1)^2$	$K_p = 2.808$ $K_i = 1.712$ $K_d = 1.151$ $f_1: t_s = 4.78$ $f_2: OS = 31.59\%$ $f_3: t_r = 0.664$ $f_5: ISE = 0.854$	$K_p = 2.391$ $K_i = 1.072$ $K_d = 1.458$ $f_1: t_s = 3.63$ $f_2: OS = 5.84\%$ $f_3: t_r = 0.676$ $f_5: ISE = 0.797$	$K_p = 2.4911$ $K_i = 0.8158$ $K_d = 1.3540$ $f_1: t_s = 5.9$ $f_2: OS = 4.95\%$ $f_3: t_r = 0.701$ $f_5: ISE = 0.809$	$K_p = 1.905$ $K_i = 0.903$ $K_d = 0.989$ $f_1: t_s = 3$ $f_2: OS = 0\%$ $f_3: t_r = 0.7$ $f_5: ISE = 0.772$
$G_2(s) =$ $4.228/(s+0.5)(s^2+1.64s+8.456)$	$K_p = 2.190$ $K_i = 2.126$ $K_d = 0.565$ $f_1: t_s = 6.6$ $f_2: OS = 16.46\%$ $f_3: t_r = 0.8$ $f_5: ISE = 0.785$	$K_p = 1.637$ $K_i = 0.964$ $K_d = 0.387$ $f_1: t_s = 5.97$ $f_2: OS = 3\%$ $f_3: t_r = 2.45$ $f_5: ISE = 0.588$	$K_p = 2.517$ $K_i = 2.219$ $K_d = 1.151$ $f_1: t_s = 6.51$ $f_2: OS = 16\%$ $f_3: t_r = 0.627$ $f_5: ISE = 0.684$	$K_p = 2.1604$ $K_i = 1.8546$ $K_d = 1.6920$ $f_1: t_s = 5.22$ $f_2: OS = 0.09\%$ $f_3: t_r = 2.54$ $f_5: ISE = 0.448$
$G_3(s) = 27/(s+1)(s+3)^3$	$K_p = 3.072$ $K_i = 2.272$ $K_d = 1.038$ $f_1: t_s = 8.2473$ $f_2: OS = 51.47\%$ $f_3: t_r = 22.5$ $f_5: ISE = 0.66$	$K_p = 1.772$ $K_i = 1.061$ $K_d = 0.772$ $f_1: t_s = 7.3959$ $f_2: OS = 30.7\%$ $f_3: t_r = 15.3$ $f_5: ISE = 0.7311$	$K_p = 2.058$ $K_i = 1.137$ $K_d = 0.746$ $f_1: t_s = 7.0311$ $f_2: OS = 19.82\%$ $f_3: t_r = 8.9$ $f_5: ISE = 0.708$	$K_p = 1.702$ $K_i = 1.061$ $K_d = 0.772$ $f_1: t_s = 6.973$ $f_2: OS = 6\%$ $f_3: t_r = 6.2$ $f_5: ISE = 0.501$

3. Simulation Results

In this section, we presented the numerical results to improve the performance of the proposed solution algorithm.

All the computation is implemented with Matlab/Simulink. The values of the parameters in ACO are $m = 500$ (numbers of ants), $\alpha = 0.5$, $\beta = 0.5$, $\rho = 0.5$, and maximum generation = 300.

In this study, we utilised two examples in order to illustrate the efficiency of the proposed algorithm.

Example 1. The performance of the algorithm developed was tested with three transfer functions with different order. The chosen performance criterion is often a weighted combination of various performance characteristics such as rise time, settling time, overshoot, and integral of the square of the error [24].

We chose the following objective function:

$$L^A = \min(\lambda_1 f_1 + \lambda_2 f_2 + \lambda_5 f_5). \tag{9}$$

The objective function here is f_1: the setting time to measure the performance of the closed-loop system, f_2: the overshoot, and f_5: the integral square error that should be minimized.

Therefore the vector of weights is $\Phi = [1\ 1\ 0\ 0\ 1\ 0\ 0]$.

The closed loop PID controller cascaded with the process was tuned for the values K_p, K_i, and K_d first by using the Ziegler-Nichols method [23], genetic algorithm [24], ant system [12] and then by our multiobjective ant colony algorithm. So that, the percent maximum overshoot, the settling time, the rise time, and the integral of the squared error were computed in both cases and given in Table 1.

In this table OS represents the percent maximum overshoot, t_s is the 5 percent settling time, t_r is the rise time, and ISE is the integral of the squared error.
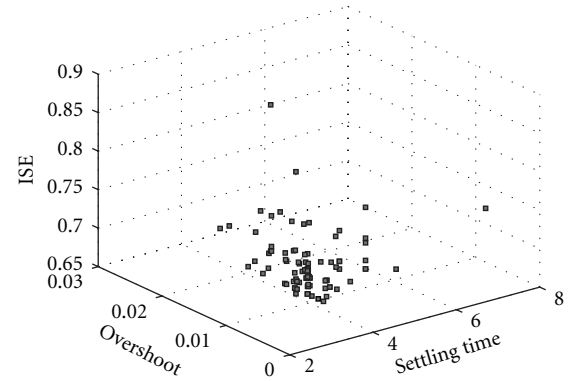

FIGURE 4: Multiobjective optimization Pareto-sets of the $G_1(s)$ plant.

The objective function used in [24] and [12] is defined using the performance indices: the response overshoot (OS), the 5% settling time t_s, and integral of the square of the error (ISE).

As shown in results of Table 1 and in all the cases tested, the value of the maximum overshoot is quite small, nearly zero percent and the values of the settling times, the rise time, and the integral of the squared error obtained by multiobjective ant colony optimization were much less than those values by the other methods.

The graphs of the obtained three-dimensional Pareto optimal fronts (the settling times, the overshoot, and the squared error) for the generated problem corresponding in each transfer function are depicted in Figures 4, 7, and 10. So that, it is possible to find a well-distributed set of nondominated solutions along the Pareto optimal front.
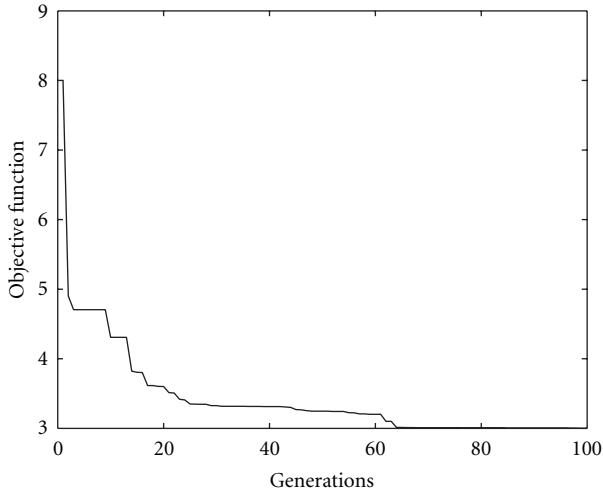
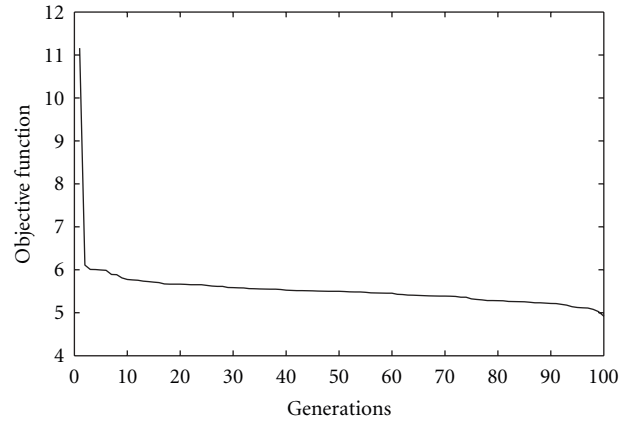FIGURE 5: Convergence graph in the ACO method of the $G_1(s)$ plant.

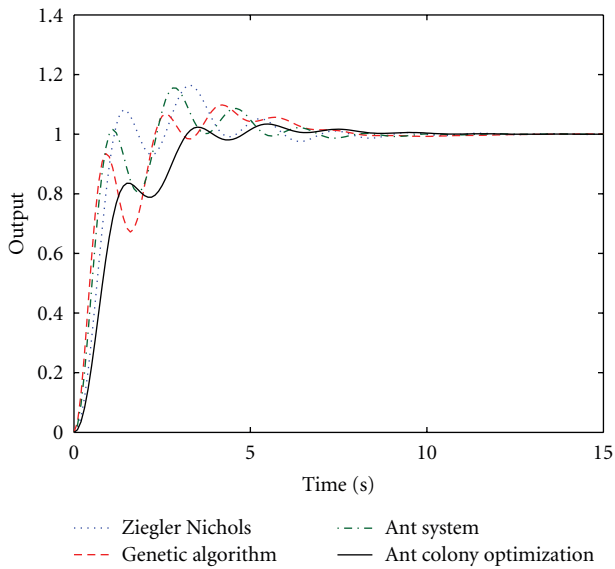FIGURE 8: Convergence graph in the ACO method of the $G_2(s)$ plant.

Ziegler Nichols
Genetic algorithm
Ant system
Ant colony optimization

FIGURE 6: Comparison of step responses of the $G_2(s)$ plant.
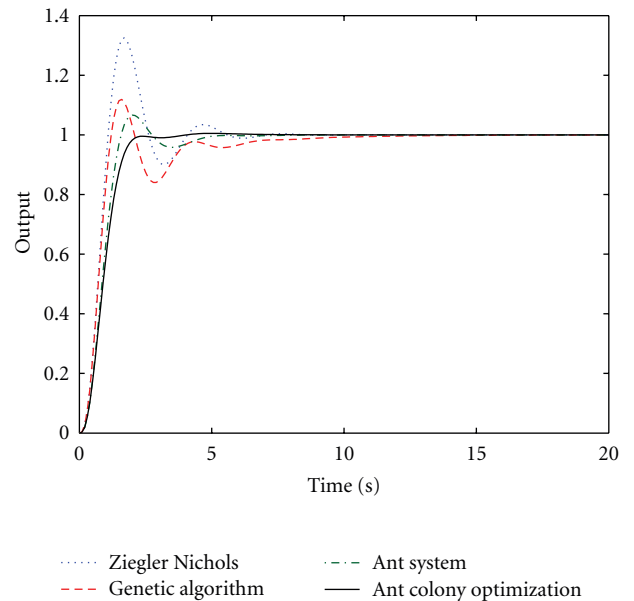

Ziegler Nichols
Genetic algorithm
Ant system
Ant colony optimization

FIGURE 9: Comparison of step responses of the $G_3(s)$ plant.
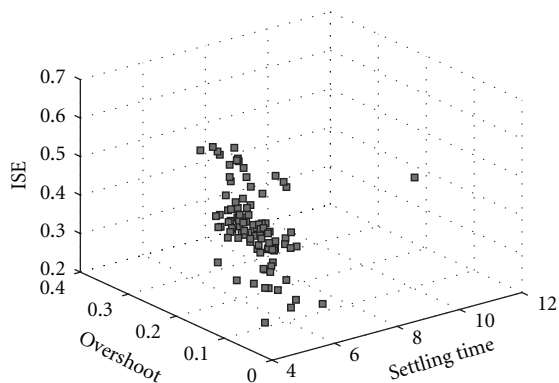

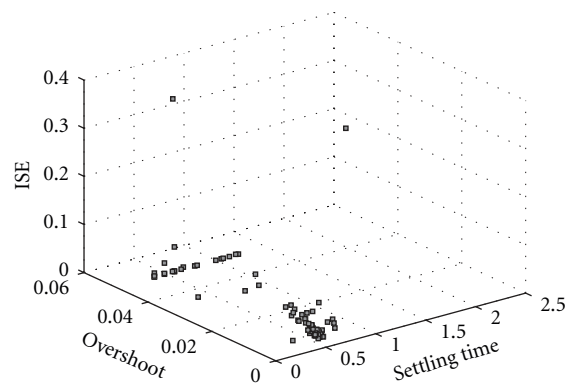FIGURE 7: Multiobjective optimization the Pareto sets of the $G_2(s)$ plant.

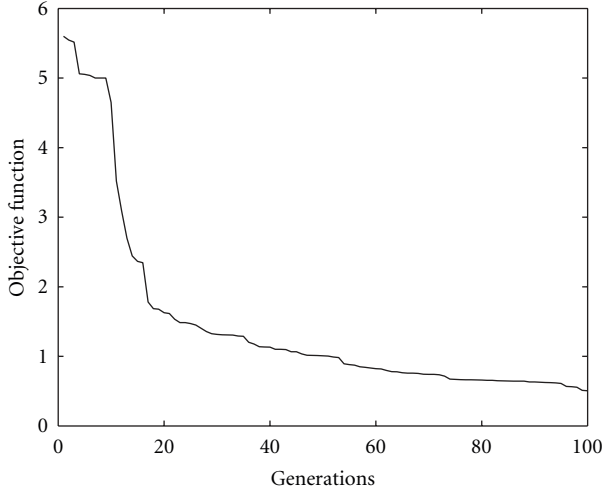FIGURE 10: Multiobjective optimization Pareto sets of the $G_3(s)$ plant.

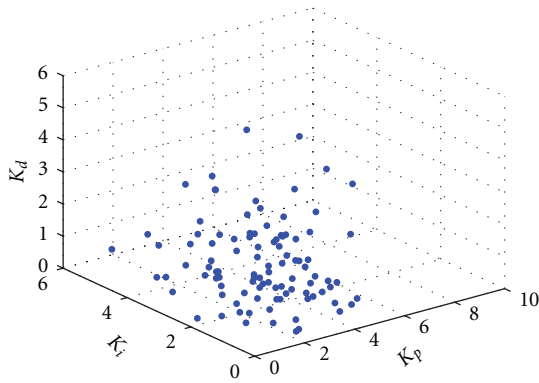FIGURE 11: Convergence graph in the ACO method of the $G_3(s)$ plant.

FIGURE 12: The initial search space of parameters (K_p, K_i, K_d).

FIGURE 13: The Intermediate search space of parameters (K_p^{best}, K_i^{best}, K_d^{best}).

FIGURE 14: The final search space of parameters (K_p^{opt}, K_i^{opt}, K_d^{opt}).

Figures 3, 6, and 9 show the step responses of $G_1(s)$, $G_2(s)$, and $G_3(s)$, respectively, which are ploted with the optimum values of the parameters K_p, K_i, and K_d obtained by the proposed algorithm. the results obtained by using Ziegler-Nichols, genetic algorithm, and ant system algorithm are presented for comparison.

In all cases, the proposed algorithm produces better responses than that obtained using the other methods. So, we can say that the multiobjective ant colony algorithm well improves the performance of the PID controllers.

Figures 5, 8, and 11 report the evaluation of the objective function on the above three transfer functions. It is observed that the objective function value decreases substantially.

Initially, each parameter (K_p, K_i, K_d) is randomly and uniformly distributed with an average value which is equal to the value founded by Ziegler-Nichols of the transfer function $G_1(s)$, the search space of these parameters is shown in Figure 12.

After several iterations, the multiobjective ant colony algorithm generated the best solutions of the PID parameters (K_p^{best}, K_i^{best}, K_d^{best}), the search space of these parameters is
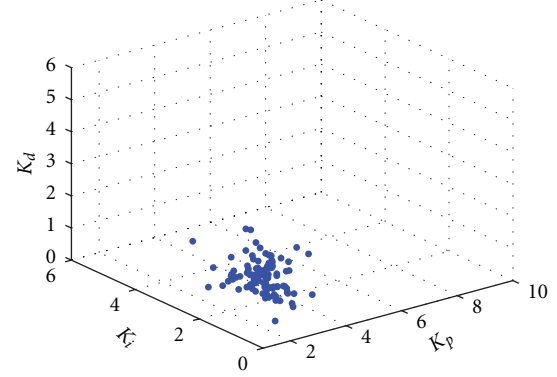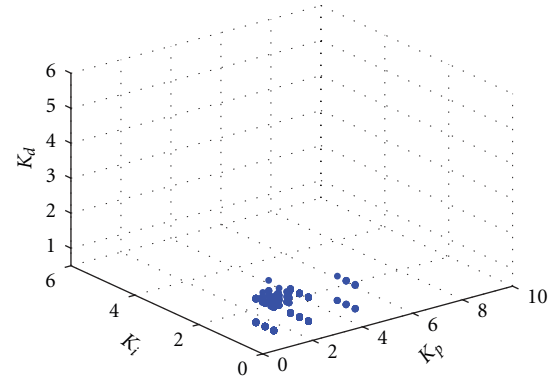
shown in Figure 13, it may be noted that these solutions are somewhat condensed.

After that, each parameter ($K_p^{\text{best}} K_i^{\text{best}} K_d^{\text{best}}$) is randomly and uniformly distributed with an average value which is equal to the value founded in the last generation, the multiobjective ant colony algorithm generated the optimal solutions ($K_p^{\text{opt}} K_i^{\text{opt}} K_d^{\text{opt}}$), as shown in the Figure 14, and these solutions are more condensed than that found in the last generation.
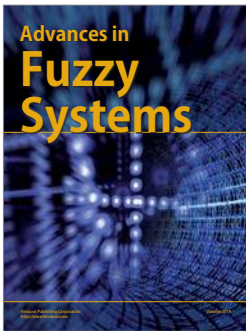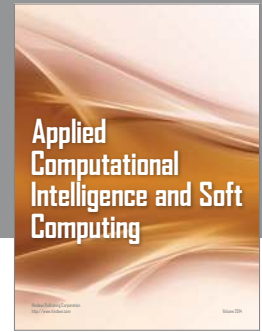
4. Conclusions

In this study, a tuning PID method based on the multiobjective ant colony optimization is developed for getting good performances and tunes the optimal PID parameters. In contrast to the single-objective algorithms, which try to find a single solution of the problem, the multiobjective technique searches for the optimal Pareto set directly. The aim of the multiobjective ACO algorithm is to determine the optimal solutions of the PID controller parameters by minimization the multiobjective function and to identify the Pareto optimal solution. This method is able to ind the optimum solution of the PID controller's parameters (K_p, K_i, and K_d) that they allow to guarantee the performance of the system.

Simulation results demonstrate that the new tuning method using multiobjective ant colony optimization has a better control system performance compared with classic approach, the genetic algorithms and ant system. The multiobjective ACO algorithm is able to undertake local search with a fast convergence rate. From the simulation study it has been found that this method converges to the global optimum.

References

[1] A. Oonsivilai and P. Pao-La-Or, "Application of adaptive tabu search for optimum PID controller tuning AVR system," *WSEAS Transactions on Power Systems*, vol. 3, no. 6, pp. 495–506, 2008.

[2] S. Tzafestas and N. P. Papanikolopoulos, "Incremetal fuzzy expert PID control," *IEEE Transactions on Industrial Electronics*, vol. 37, no. 5, pp. 365–371, 1990.

[3] A. Visioli, "Tuning of PID controllers with fuzzy logic," *IEE Proceedings: Control Theory and Applications*, vol. 148, no. 1, pp. 1–8, 2001.

[4] C. Cao, X. Guo, and Y. Liu, "Research on ant colony neural network PID controller and application," in *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07)*, pp. 253–258, 2007.

[5] T. L. Seng, M. B. Khalid, and R. Yusof, "Tuning of a neuro-fuzzy controller by genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics Part B*, vol. 29, no. 2, pp. 226–236, 1999.

[6] D. H. Kim, "Tuning of a PID controller using a artificial immune network model and local fuzzy set," in *Proceedings of the Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS '01)*, vol. 5, pp. 2698–2703, 2001.

[7] G. Zhou and J. D. Birdwell, "Fuzzy logic-based PID autotuner design using simulated annealing," in *Proceedings of the IEEE/IFAC Joint Symposium on Computer-Aided*, pp. 67–72, 1994.

[8] P. Wang and D. P. Kwok, "Optimal design of PID process controllers based on genetic algorithms," *Control Engineering Practice*, vol. 2, no. 4, pp. 641–648, 1994.

[9] P. Wang and D. P. Kwok, "Optimal design of PID process controllers based on genetic algorithms," *Control Engineering Practice*, vol. 2, no. 4, pp. 641–648, 1994.

[10] Y. Mitsukura, T. Yamamoto, and M. Kaneda, "A design of self turning PID controllers'using a genetic algorithm," in *Proceedings of the American Control Conference*, pp. 1361–1365, San Diego, Calif, USA, 1999.

[11] S. E. Selvan, S. Subramanian, and S. T. Solomon, "Novel technique for PID tuning by particle swarm optimization," in *Proceedings of the 7th Annual Swarm Users/Researchers Conference (SwarmFest '03)*, 2003.

[12] Y. T. Hsiao, C. L. Chuang, and C. C. Chien, "Ant colony optimization for designing of PID controllers," in *Proceedings of the IEEE lnternational Symposium on Computer Aided Control Systems Design*, Taipei, Taiwan, 2004.

[13] M. Dorigo and G. Di Caro, "The ant colony optimization meta-heuristic," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds., pp. 11–32, McGraw Hill, London, UK, 1999.

[14] M. Dorigo, G. Di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.

[15] G. Reinelt, *The Traveling Salesman: Computational Solutions for TSP Applications*, vol. 840 of *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 1994.

[16] T. Stützle and M. Dorigo, "ACO algorithms for the quadratic assignment problem," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds., pp. 33–50, McGraw Hill, London, UK, 1999.

[17] D. Costa and A. Hertz, "Ants can colour graphs," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 295–305, 1997.

[18] S. J. Huang, "Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches," *IEEE Transactions on Energy Conversion*, vol. 16, no. 3, pp. 296–301, 2001.

[19] L. M. Gambardella, E. D. Taillard, and G. Agazzi, "MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds., pp. 63–76, McGraw Hill, London, UK, 1999.

[20] L. M. Gambardella and M. Dorigo, "An ant colony system hybridized with a new local search for the sequential ordering problem," *INFORMS Journal on Computing*, vol. 12, no. 3, pp. 237–255, 2000.

[21] G. Di Caro and M. Dorigo, "Ant colonies for adaptive routing in packetswitched communications networks," in *Proceedings of the Proceedings of 5th International Conference on Parallel Problem Solving from Nature (PPSN '98)*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds., vol. 1498 of *Lecture Notes in Computer Science*, pp. 673–682, Springer, Berlin, Germany, 1998.

[22] A. Afshar, A. Kaveh, and O. R. Shoghli, "Multi-objective optimization of time-cost-quality using multi-colony ant algorithm," *Asian Journal Of Civil Engineering*, vol. 8, no. 2, pp. 113–124, 2007.

[23] J. G. Ziegler and N. B. Nichols, "Optimum settlings for automatic controllers," *Transactions of the ASME*, vol. 64, pp. 759–768, 1942.

[24] A. Bagis, "Determination of the PID controller parameters by modified genetic algorithm for improved performance," *Journal of Information Science and Engineering*, vol. 23, no. 5, pp. 1469–1480, 2007.

Advances in
Multimedia

The Scientific World Journal

International Journal of
Distributed Sensor Networks

Journal of
Industrial Engineering

Applied
Computational Intelligence and Soft Computing

Advances in
Fuzzy Systems

Modelling & Simulation in Engineering

Journal of
Computer Networks and Communications

Advances in
Artificial Intelligence
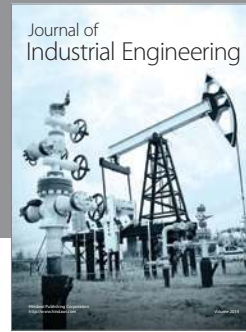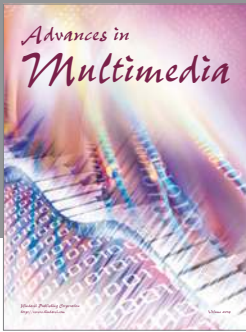
Hindawi

Submit your manuscripts at
http://www.hindawi.com

Advances in
Computer Engineering

International Journal of
Computer Games Technology

International Journal of
Biomedical Imaging

Advances in
Artificial Neural Systems

Advances in
Software Engineering

Journal of
Robotics

Advances in
Human-Computer Interaction

Computational Intelligence and Neuroscience

International Journal of
Reconfigurable Computing

Journal of
Electrical and Computer Engineering