International Journal of Innovative Computing, Information and Control
Volume **16**, Number **1**, February **2020**

ICIC International ©2020 ISSN 1349-4198
pp. 173–187

MULTIOBJECTIVE LÉVY-FLIGHT FIREFLY ALGORITHM FOR OPTIMAL PIDA CONTROLLER DESIGN

Somchai Sumpunsri and Deacha Puangdownreong

Department of Electrical Engineering
Southeast Asia University
19/1 Petchkasem Road, Nongkhangphlu, Nonghkaem, Bangkok 10160, Thailand
deachap@sau.ac.th

Received February 2019; revised August 2019

Abstract. *Firefly algorithm (FA) was proposed as one of the most powerful population-based metaheuristic optimization techniques for solving continuous and combinatorial optimization problems. The FA has been proved and applied to various real-world engineering problems in mostly single objective optimization manner. However, many real-world engineering problems are typically formulated as the multiobjective optimization problems with complex constraints. In this paper, the multiobjective Lévy-flight firefly algorithm (mLFFA) is developed. The proposed mLFFA is validated against four standard multiobjective test functions. Then, the mLFFA is applied to optimally design the proportional-integral-derivative-accelerated (PIDA) controllers for the automatic voltage regulator (AVR) system in order to simultaneously minimize two particular objective functions, i.e., rise time and maximum overshoot. As simulation results, it was found that the mLFFA can provide very satisfactory solutions for all benchmark test functions. Moreover, the optimal PIDA controllers can be successfully obtained by the mLFFA according to the predefined objective and constraint functions to perform the optimal Pareto front containing the set of optimal PIDA controllers for the AVR system in the proposed control application.*

Keywords: Multiobjective Lévy-flight firefly algorithm, PIDA controller, Automatic voltage regulator, Modern optimization

1. **Introduction.** Intelligent control system design has been changed from the conventional paradigm to multiobjective design optimization framework over two decades [1]. Such the multiobjective optimization problems can be effectively solved by efficient metaheuristic optimization searching techniques. Following the literature, many metaheuristics are consecutively developed and launched to perform their effectiveness [2-4]. Among them, the firefly algorithm (FA) was firstly proposed in 2008 by Yang [5,6] based on the flashing behavior of fireflies and uniform distribution for randomly generating the feasible solutions. As one of the most efficient population-based metaheuristic algorithms, the FA was applied to almost every area of sciences and engineering, including power systems, image processing, antenna design, civil engineering, robotics and control engineering [7,8].

In 2010, two years after the former version of the FA was initiated, the later version of FA named the Lévy-flight firefly algorithm (LFFA) was proposed by Yang [9]. The algorithm of LFFA was still based on the flashing behavior of fireflies, but Lévy-flight distribution is employed to randomly generate new solutions. The LFFA was tested against several nonlinear and multimodal standard test functions. Results obtained by the LFFA outperformed those by traditional algorithms including genetic algorithms (GA)

DOI: 10.24507/ijicic.16.01.173

and particle swarm optimization (PSO). The state-of-the-art and its applications of the LFFA have been reviewed and reported [7-9].

Many real-world engineering design problems often consist of many objectives which conflict each other [1-4]. This leads the multiobjective problems much more difficult and complex than single-objective ones. The multiobjective problem possesses multiple optimal solutions forming the so-called Pareto front [1-4]. The challenge is how to perform the smooth Pareto front containing a set of optimal solutions for all objective functions. From literature reviews, the multiobjective problems can be efficiently solved by the modern optimization techniques or metaheuristics, for example, GA [10], PSO [10-12], cuckoo search (CS) [13], flower pollination algorithm (FPA) [14], game algorithm (GaA) [15], whale optimization algorithm (WOA) [16] and hybrid metaheuristics [17].

According to control system design, the proportional-integral-derivative-accelerated (P-IDA) controller was firstly proposed by Jung and Dorf in 1996 [18]. It possesses three arbitrary zeros and one pole at origin. This can provide faster and smoother responses for the higher-order plants than the PID controller. Designing the PIDA controller can be optimized by metaheuristics such as GA [19], PSO [20], current search (CuR) [21], FA [22], bat algorithm (BA) [23,24] and FPA [25,26]. In control system design, the system response needs to meet the design specification that is priori preset. For the time-domain response, rise time (t_r) and maximum percent overshoot (M_p) are usually set as the design specification. However, they often conflict to each other. Therefore, the PIDA controller design scheme can be considered as one of the multiobjective optimization problems.

The multiobjective Lévy-flight firefly algorithm (mLFFA) is proposed in this paper to optimize the PIDA controller based on the multiobjective optimization problems. Due to the Lévy-flight drawn from the Lévy distribution having an infinite mean and infinite variance, the proposed mLFFA is thus very suitable for the PIDA controller design. In this paper, the performance of the mLFFA is evaluated against four standard multiobjective test functions. Then, the proposed mLFFA is applied to optimally design the PIDA controllers for the automatic voltage regulator (AVR) system. This paper is arranged as follows. After an introduction is given in Section 1, the multiobjective optimization problems are described in Section 2. Algorithms of FA, LFFA and the proposed mLFFA are illustrated in Section 3. The performance evaluation of the mLFFA against four standard multiobjective test functions is performed in Section 4. Application of the mLFFA to design optimal PIDA controllers for AVR system based on multiobjective optimization is discussed in Section 5, while conclusions are given in Section 6.

2. **Multiobjective Optimization.** Based on the optimization context [1-4], the multi-objective continuous optimization problems can be expressed in (1), where $\boldsymbol{F}(\boldsymbol{x})$ is the multiobjective function consisting of $f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})$, $n \geq 2$, $g_j(\boldsymbol{x})$, $j = 1, 2, \ldots, m$, is the inequality constraints and $h_k(\boldsymbol{x})$, $k = 1, 2, \ldots, p$, is the equality constraints. The optimal solutions, $\boldsymbol{x}^*$, are the solutions that can make $\boldsymbol{F}(\boldsymbol{x})$ minimum and make both $g_j(\boldsymbol{x})$ and $h_k(\boldsymbol{x})$ satisfied.

$$\left.\begin{array}{ll} \text{Min } \boldsymbol{F}(\boldsymbol{x}) = & \{f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})\} \\ \text{subject to} & g_j(\boldsymbol{x}) \leq 0, \quad j = 1, \ldots, m \\ & h_k(\boldsymbol{x}) = 0, \quad k = 1, \ldots, p \end{array}\right\} \tag{1}$$

Referring to the Pareto optimality [1-4,27-29], a solution $\boldsymbol{x}^*$ is called a non-dominated solution if no solution can be found that dominates it. In other words, a solution $\boldsymbol{x}^*$ is Pareto optimal if $\boldsymbol{F}(\boldsymbol{x}^*) \prec \boldsymbol{F}(\boldsymbol{x})$. For a given multiobjective optimization problem, the Pareto optimal set is defined as P^* stated in (2). The Pareto front PF^* of a given multiobjective optimization problem can be defined as the image of the Pareto optimal set P^* expressed in (3). In case of bi-objective optimization problem, its Pareto front can be performed as shown in Figure 1 to demonstrate the trade-off characteristics between two objective functions.

$$P^* = \{\boldsymbol{x} \in \boldsymbol{F} | \exists \boldsymbol{x}^* \in \boldsymbol{F} : \boldsymbol{F}(\boldsymbol{x}^*) \prec \boldsymbol{F}(\boldsymbol{x})\} \tag{2}$$

$$PF^* = \{\boldsymbol{s} \in S | \exists \boldsymbol{s}^* \in S : \boldsymbol{s}^* \prec \boldsymbol{s}\} \tag{3}$$

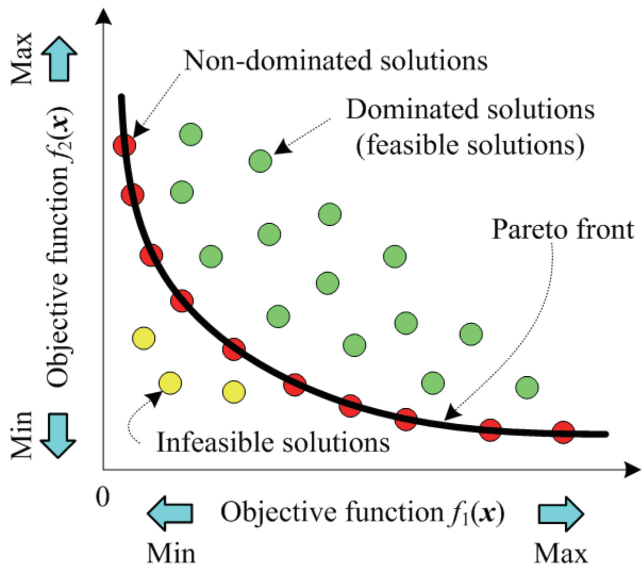

FIGURE 1. Pareto front of bi-objective problem

3. **The mLFFA Algorithm.** To understand the proposed mLFFA, the original FA and LFFA are briefly reviewed. Then, the algorithm of the proposed mLFFA is elaborately described.

3.1. **FA algorithm.** The original firefly algorithm (or FA) was firstly developed by Yang in 2008 by [5,6] based on the flashing behavior of fireflies. The flashing light of fireflies is produced by a process of bioluminescence to attract mating partners (communication) and to attract potential prey. The FA's algorithm is developed from three idealized rules [5,6]:

Rule (1): fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex;

Rule (2): the attractiveness is proportional to the brightness, and they both decrease as their distance increases. Thus for any two flashing fireflies, the less brighter one will move towards the brighter one. If there is no brighter one than a particular firefly, it will move randomly; and

Rule (3): the brightness of a firefly is determined by the landscape of the objective function.

In FA, there are two important issues: the variation of light intensity and formulation of the attractiveness. The attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function. Along the distance r, the light intensity I varies according to the inverse square law $I(r) = I_s/r^2$, where I_s is the intensity at the source. For a given medium with a fixed light absorption coefficient, the light intensity I varies with the distance r as stated in (4), where I_0 is the original light intensity.

$$I = I_0 e^{-\gamma r} \tag{4}$$

$$\beta = \beta_0 e^{-\gamma r^2} \tag{5}$$

$$r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\| = \sqrt{\sum_{k=1}^{d} (\boldsymbol{x}_{i,k} - \boldsymbol{x}_{j,k})^2} \tag{6}$$

The attractiveness of a firefly observed by adjacent fireflies is proportional to the light intensity. This can define the variation of attractiveness β with the distance r as expressed in (5), where β_0 is the attractiveness at $r = 0$. From parametric studies, $\beta_0 = 1$ is suggested for most applications [5,6]. The scaling factor γ in (4) and (5) is defined as the light absorption coefficient. In addition in (4) and (5), the distance r_{ij} between any two fireflies i and j at their locations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be calculated by the Cartesian distance as expressed in (6), where $\boldsymbol{x}_{i,k}$ is the kth component of the spatial coordinate $\boldsymbol{x}_i$ of ith firefly.

For FA, the new solution $\boldsymbol{x}^{t+1}$ can be obtained by the old solution $\boldsymbol{x}^t$ as formulated in (7). Referring to (7), the movement of a firefly i is attracted to another more attractive (brighter) firefly j, where α_t is the randomization parameter, and $\boldsymbol{\varepsilon}_i$ is a vector of random numbers drawn from a Gaussian distribution or uniform distribution at time t [5,6]. In addition, α_t can be controlled during iterations as stated in (8), where α_0 is the initial randomness scaling factor, and δ is a cooling factor.

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \beta_0 e^{-\gamma r_{ij}^2} \left(\boldsymbol{x}_j^t - \boldsymbol{x}_i^t\right) + \alpha_t \boldsymbol{\varepsilon}_i^t \tag{7}$$

$$\alpha_t = \alpha_0 \delta^t, \quad (0 < \delta < 1) \tag{8}$$

3.2. **LFFA algorithm.** The Lévy-flight firefly algorithm (or LFFA), the modified version of the FA, was proposed by Yang in 2010 [9]. Movement of a firefly i is attracted to another more attractive (brighter) firefly j as determined by (9), where the second term is due to the attraction while the third term is randomization via Lévy flights with α being the randomization parameter. Referring to (9), the product $\oplus$ means entrywise multiplications. The sign [rand $- 1/2$] where rand $\in [0, 1]$ essentially provides a random sign or direction while the random step length is drawn from a Lévy distribution having an infinite variance with an infinite mean. From (9), a symbol Lévy (λ) represents the Lévy distribution as expressed in (10). The step length s can be calculated by (11), where u and v stand for normal distribution as stated in (12). Standard deviations of u and v are also expressed in (13).

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \beta_0 e^{-\gamma r_{ij}^2} \left(\boldsymbol{x}_j^t - \boldsymbol{x}_i^t\right) + \alpha \text{sign}\left[\text{rand} - \frac{1}{2}\right] \oplus \text{Lévy}(\lambda) \tag{9}$$

$$\text{Lévy} \approx u = t^{-\lambda}, \quad (1 < \lambda \leq 3) \tag{10}$$

$$s = \frac{u}{|v|^{1/\beta}} \tag{11}$$

$$u \approx N\left(0, \sigma_u^2\right), \quad v \approx N\left(0, \sigma_v^2\right) \tag{12}$$

$$\sigma_u = \left\{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}\right\}^{1/\beta}, \quad \sigma_v = 1 \tag{13}$$

3.3. **Proposed mLFFA algorithm.** The multiobjective Lévy-flight firefly algorithm (or mLFFA) is thus proposed. The LFFA can be modified to minimize the $\boldsymbol{F}(\boldsymbol{x})$ in (1). The algorithm of the proposed mLFFA can be represented by the pseudo code as shown in Figure 2. Multiobjective function $\boldsymbol{F}(\boldsymbol{x})$ as stated in (1) will be simultaneously minimized according to the equality and inequality constraints. The best solution will be checked in each iteration. If it is a non-dominated solution, it will be sorted and stored into the Pareto optimal set P^*. After the search terminated, the solutions stored in P^* will be used to perform the Pareto front PF^*. Solutions that appeared on the PF^* are the optimal solutions of the problem of interest.

```
Multiobjective function:
F(x) = {f1(x), f2(x),…,fn(x)}, x = (x1,…,xd)^T
Initialize LFFA1,…,LFFAk, and Pareto optimal set P*
Generate initial population of fireflies xi = (i = 1, 2,…,n)
Light intensity Ii at xi is determined by F(xi)
Define light absorption coefficient γ
while (Gen ≤ Max_Generation)
for z =1: k all k LFFA
    for i =1 : n all n fireflies
        for j =1 : i all n fireflies
            if (Ij > Ii)
                - Move firefly i towards j in d-dimension via
                  Lévy-flight distributed random
            end if
            - Attractiveness varies with distance r via exp[-γr]
            - Evaluate new solutions and update light intensity
        end for j
    end for i
end for z
- Rank the fireflies and find the current best x*
- Sort and find the current Pareto optimal solutions
end while
- Pareto optimal solutions are found and
- Pareto front PF* is performed.
```

Figure 2. Pseudo code of mLFFA algorithm

Referring to the pseudo code in Figure 2 representing the mLFFA algorithm, the multiobjective function $\boldsymbol{F}(\boldsymbol{x})$ as stated in (1) is firstly initialized. $\text{LFFA}_1, \ldots, \text{LFFA}_k$, and Pareto optimal set P^* are also initially performed. The initial population of fireflies $\boldsymbol{x}_i = (i = 1, 2, \ldots, n)$ is randomly generated. The light intensity I_i at $\boldsymbol{x}_i$ is determined by $\boldsymbol{F}(\boldsymbol{x}_i)$. The light absorption coefficient γ, the initial attractiveness β_0 and the initial randomness scaling factor α_0 are also defined. Gen = 1 as a counter and Max_Gen as the termination criteria (TC) are set for terminating the search process. In the iteration process, the mLFFA algorithm will check the TC. If Gen $\leq$ Max_Gen, the search process will continue. Otherwise the search process will be stopped and the best solution will be reported. In each iteration, n fireflies will check the light intensity I to each other. If $I_j > I_i$, a firefly i will move firefly toward a firefly j in d-dimension via Lévy-flight random distribution. Otherwise, a firefly i will move randomly. After that, the attractiveness β, the light intensity I and the randomness scaling factor α will be updated. The new positions of all fireflies (new solutions) are ranked. The current best solution $\boldsymbol{x}^*$ is then updated. The mLFFA algorithm will be iteratively processed until the TC is met. For post process of iteration, all best solution $\boldsymbol{x}^*$ found by all LFFA will be sorted to generate the Pareto optimal solutions into the Pareto optimal set P^*. Finally, the Pareto front PF^* containing all Pareto optimal solutions will be performed.

4. **Performance Evaluation.** In order to perform its effectiveness, the mLFFA is then evaluated against several multiobjective test functions. In this work, four widely used standard multiobjective functions, ZDT1-ZDT4, providing a wide range of diverse properties in terms of Pareto front and Pareto optimal set are conducted [30,31]. ZDT1 is with convex front as stated in (14) where d is the number of dimensions. ZDT2 as stated in (15) is with non-convex front, while ZDT3 with discontinuous front is expressed in (16), where g and x_i in functions ZDT2 and ZDT3 are the same as in function ZDT1. For ZDT4, it is stated in (17) with convex front but more specific.

$$\left.\begin{aligned} f_1(\boldsymbol{x}) = \boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = g\left(1 - \sqrt{f_1/g}\right), \\ g = 1 + \left(9\sum_{i=2}^{d} \boldsymbol{x}_i\right) \Big/ (d-1), \quad \boldsymbol{x}_i \in [0,1], \quad i = 1, \ldots, 30. \end{aligned}\right\} \tag{14}$$

$$f_1(\boldsymbol{x}) = \boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = g\left(1 - \frac{f_1}{g}\right)^2 \tag{15}$$

$$f_1(\boldsymbol{x}) = \boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = g\left(1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\sin(10\pi f_1)\right) \tag{16}$$

$$\left.\begin{aligned} f_1(\boldsymbol{x}) = \boldsymbol{x}_1, \quad f_2(\boldsymbol{x}) = g\left(1 - \sqrt{f_1/g}\right), \\ g = 1 + 10(d-1) + \sum_{i=2}^{d} \left[\boldsymbol{x}_i^2 - 10\cos(4\pi f_1)\right], \quad \boldsymbol{x}_i \in [0,1], \quad i = 1, \ldots, 30. \end{aligned}\right\} \tag{17}$$

$$E_f = \|PF_e - PF_t\| = \sum_{j=1}^{N} \left(PF_e^j - PF_t\right)^2 \tag{18}$$

In evaluation process, the error E_f between the estimated Pareto front PF_e and its corresponding true front PF_t is defined in (18), where N is the number of solution points. The proposed mLFFA algorithms were coded by MATLAB version 2018b (License No.#40637337) run on Intel(R) Core(TM) i5-3470 CPU@3.60 GHz, 4.0 GB-RAM. Search parameters of each LFFA in the mLFFA are set according to Yang's recommendations [6,9], i.e., the numbers of fireflies $n = 30$, $\alpha_0 = 0.25$, $\beta_0 = 1$, $\lambda = 1.50$ and $\gamma = 1$. These searching parameters of LFFA are sufficient for most optimization problems because the LFFA algorithm is very robust (not very sensitive to the parameter adjustment) [6,9]. In this work, the TC either uses a given tolerance or a fixed number of generations. As implementation results, it was found that a fixed number of generations is not only easy to implement, but also suitable to compare the closeness of Pareto front of test functions. Therefore, for all test functions, Max_Gen = 2,000 is set as the TC.

For comparison, the results obtained by the proposed mLFFA over all test functions are compared with those obtained by the well-known algorithms, i.e., vector evaluated genetic algorithm (VEGA) [32], non-dominated sorting genetic algorithm II (NSGA-II) [33], differential evolution for multiobjective optimization (DEMO) [34] and multiobjective multipath adaptive tabu search (mMATS) [35]. The performance of all algorithms is measured via the error E_f stated in (18) and for all algorithms, a fixed number of generations/iterations of 2,000 (Max_Gen) is set as the TC. The results obtained from all test functions are summarized in Tables 1-2, and the estimated Pareto fronts and the true front of functions ZDT1-ZDT4 are depicted in Figures 3-6, respectively. It was found from all figures that the mLFFA can satisfactorily provide the Pareto front containing all Pareto optimal solutions of each function very close to the true front of each test function. Referring to Tables 1-2, the mLFFA shows superior results in term of error E_f to other algorithms with lesser search time consumed.

TABLE 1. Error E_f between PF_e and PF_t

Methods	Error E_f			
	ZDT1	ZDT2	ZDT3	ZDT4
VEGA	2.79e-02	2.37e-03	3.29e-01	4.87e-01
NSGA-II	3.33e-02	7.24e-02	1.14e-01	3.38e-01
DEMO	2.08e-03	7.55e-04	2.18e-03	2.96e-01
mMATS	1.24e-03	2.52e-04	1.07e-03	1.02e-01
mLFFA	***1.20e-03***	***2.48e-04***	***1.01e-03***	***1.01e-01***

TABLE 2. Search time consumed

Methods	Search time (sec.)			
	ZDT1	ZDT2	ZDT3	ZDT4
VEGA	125.45	132.18	121.40	122.24
NSGA-II	126.82	145.63	158.27	165.51
DEMO	89.31	98.44	102.32	120.86
mMATS	65.54	72.33	82.47	78.52
mLFFA	***52.42***	***65.18***	***71.53***	***64.78***

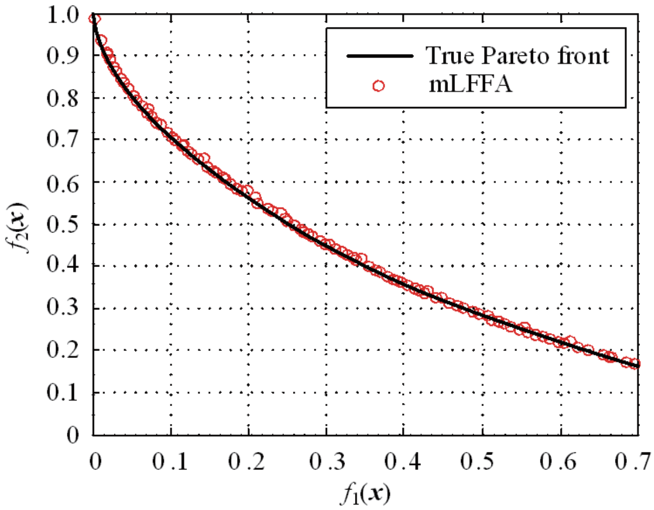

FIGURE 3. Pareto front of ZDT1

5. **The mLFFA-Based PIDA Controller Design.** The application of the mLFFA to design optimal PIDA controllers for AVR system based on multiobjective optimization is discussed in this section. The mLFFA-based PIDA controller design framework for the AVR system is represented in Figure 7. The AVR is commonly used in the generator

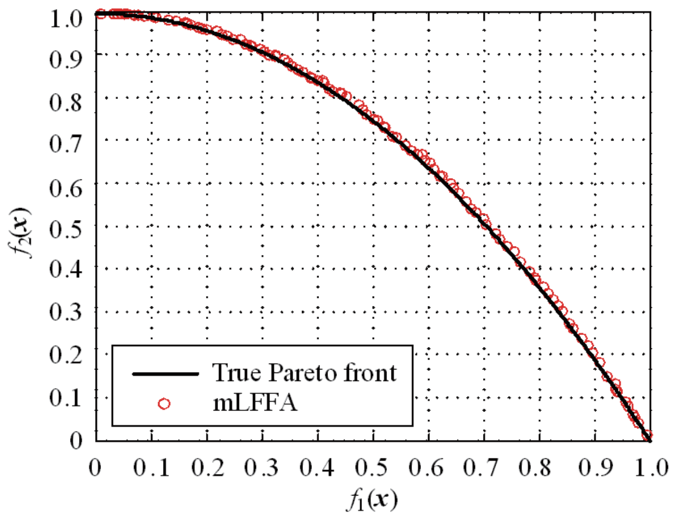

FIGURE 4. Pareto front of ZDT2

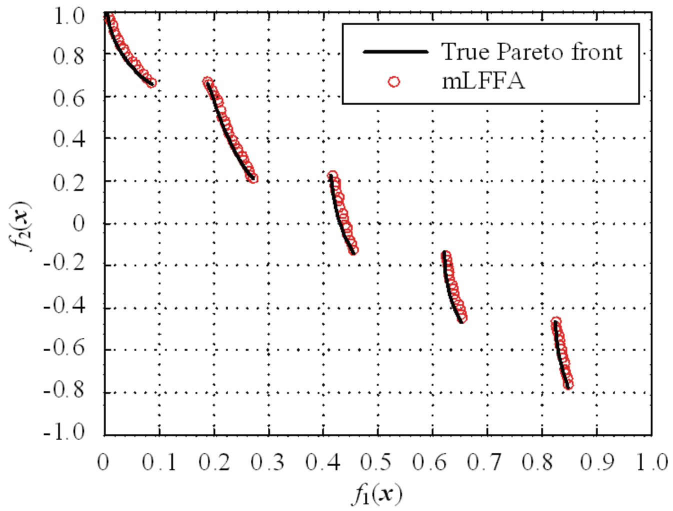

FIGURE 5. Pareto front of ZDT3

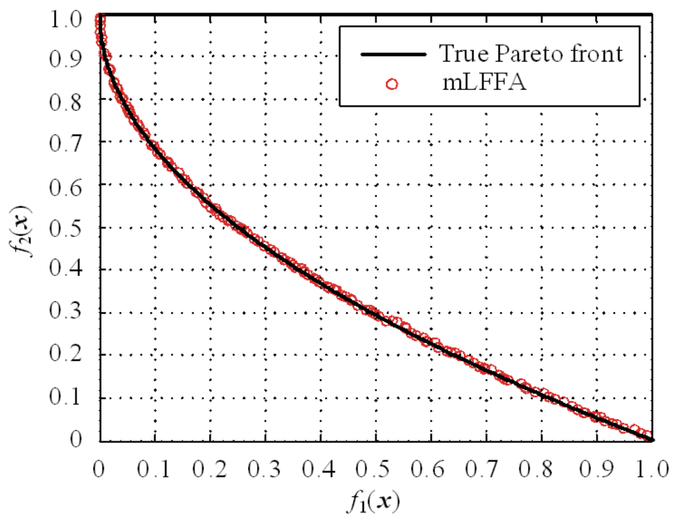

FIGURE 6. Pareto front of ZDT4

excitation system of hydro and thermal power plants. The main role of the AVR is to regulate generator voltage and control the reactive power flow at a specified level. In this work, a simple AVR consists of four main components, i.e., amplifier, exciter, generator, and sensor, respectively as shown in Figure 7, where E is the error voltage between the referent input voltage $V_{ref}(s)$ and sensor voltage V_B, while U, V_R and V_F are the controlled, amplified, and excited voltage signals, and $V_o(s)$ is the output voltage. Four main components of the AVR are linearized and modeled by transfer functions [36,37] as visualized in Figure 7. From [36,37], the amplifier gain model K_A is in the range of 10 to 400, while the amplifier time constant τ_A is very small ranging from 0.02 to 0.1 sec. For an exciter, a gain K_E is in the range of 1 to 400 and a time constant τ_E is from 0.25 to 1.0 sec. For a generator, a gain K_G may vary from 0.7 to 1.0, while a time constant τ_G is varied from 1.0 to 2.0 sec. Finally, a sensor gain K_R is very small ranging from 0.1 to 1.0, and its time constant τ_R is varied from 0.001 to 0.06 sec. Models of four main components will be used as a system plant in the control loop.

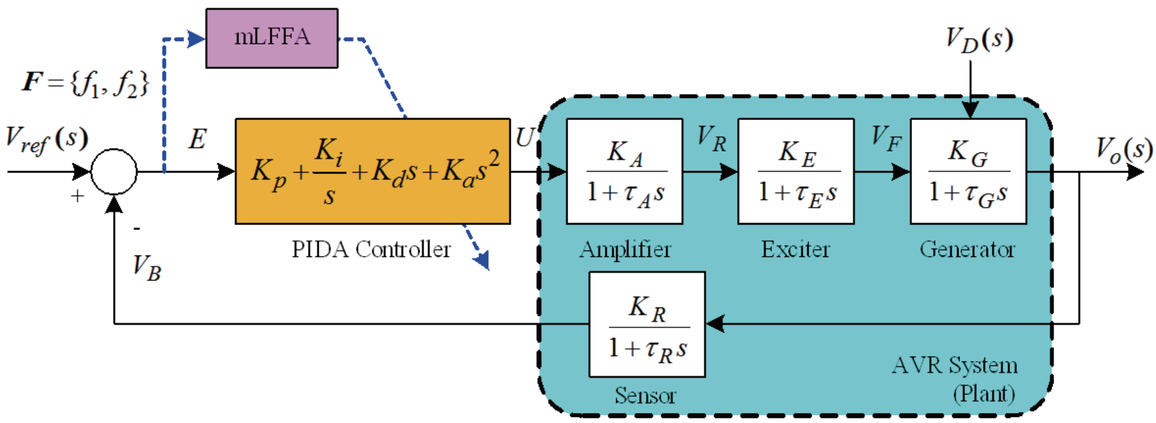

Figure 7. The mLFFA-based PIDA controller design for AVR system

Referring to the control loop in Figure 7, the PIDA controller receives the error signal, $E(s)$, and produces the control signal, $U(s)$, to control the voltage output response, $V_o(s)$, referring to the voltage referent input, $V_{ref}(s)$, and regulate the voltage output response, $V_o(s)$, from the external disturbance signal, $D(s)$. The s-domain transfer function of the PIDA controller $G_c(s)$ is stated in (19) [18], where K_p, K_i, K_d and K_a are proportional, integral, derivative and accelerated gains, respectively.

$$G_c(s)|_{PIDA} = K_p + \frac{K_i}{s} + K_d s + K_a s^2 \tag{19}$$

In the time-domain response of a controlled system, t_r and M_p usually conflict to each other. Therefore, two particular objective functions, i.e., $f_1(\boldsymbol{x}) = t_r$ and $f_2(\boldsymbol{x}) = M_p$ are then set as stated in (20) to be minimized by the mLFFA in order to obtain the optimal PIDA parameters, i.e., K_p, K_i, K_d and K_a, for the AVR system, corresponding to their constraints and search spaces as given in (21), where t_s is settling time and E_{ss} is steady-state error.

$$\left.\begin{aligned} \text{Min } \boldsymbol{F}(\boldsymbol{x}) &= \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x})\}, \\ f_1(\boldsymbol{x}) = t_r, &\quad f_2(\boldsymbol{x}) = M_p, \\ \boldsymbol{x} &= (K_p,\ K_i,\ K_d,\ K_a)^T \end{aligned}\right\} \tag{20}$$

$$\left.\begin{array}{ll}\text{subject to} & t_r \leq 0.3 \text{ sec.}, \quad M_p \leq 10\%, \\ & t_s \leq 1.5 \text{ sec.}, \quad E_{ss} \leq 0.01\%, \\ & 0 \leq K_p \leq 1, \quad 0 \leq K_i \leq 1, \\ & 0 \leq K_d \leq 0.3, \quad 0 \leq K_a \leq 0.01 \end{array}\right\} \tag{21}$$

Referring to Figure 2, the application of the mLFFA to PIDA controller design is described as follows. Firstly, the multiobjective function $\boldsymbol{F}(\boldsymbol{x})$ in (20) and the inequality constraint functions in (21) are initialized, where the solutions $\boldsymbol{x} = \{K_p, K_i, K_d \text{ and } K_a\}$ of the PIDA controller. $\text{LFFA}_1, \ldots, \text{LFFA}_k$, and Pareto optimal set P^* are also initially performed. The initial population of fireflies $\boldsymbol{x}_i = (i = 1, 2, \ldots, n)$ is randomly generated. The light intensity I_i at $\boldsymbol{x}_i$ is determined by $\boldsymbol{F}(\boldsymbol{x}_i)$. The light absorption coefficient γ, the initial attractiveness β_0 and the initial randomness scaling factor α_0 are also defined. Gen = 1 as a counter and Max_Gen as the TC are set for terminating the search process. In the iteration process, the mLFFA algorithm will check the TC. If Gen $\leq$ Max_Gen, the search process will continue. Otherwise the search process will be stopped and the best solution $\boldsymbol{x}^* = \{K_p, K_i, K_d \text{ and } K_a\}$ will be reported. In each iteration, n fireflies will check the light intensity I to each other. If $I_j > I_i$, a firefly i moves firefly toward a firefly j in d-dimension via Lévy-flight random distribution. Otherwise, a firefly i moves randomly. After that, the attractiveness β, the light intensity I and the randomness scaling factor α will be updated. The new positions of all fireflies (new solutions $\boldsymbol{x} = \{K_p, K_i, K_d \text{ and } K_a\}$) are ranked. The current best solution $\boldsymbol{x}^* = \{K_p, K_i, K_d \text{ and } K_a\}$ is then updated. The mLFFA algorithm will be iteratively processed until the TC is met. For post process of iteration, all best solution $\boldsymbol{x}^* = \{K_p, K_i, K_d \text{ and } K_a\}$ found by all LFFA will be sorted to generate the Pareto optimal solutions into the Pareto optimal set P^*. Finally, the Pareto front PF^* containing all Pareto optimal solutions $\boldsymbol{x}^* = \{K_p, K_i, K_d \text{ and } K_a\}$ will be performed.

To design optimal PIDA controllers for the AVR system based on multiobjective optimization context, the proposed mLFFA algorithms were coded by MATLAB version 2018b (License No.#40637337) run on Intel(R) Core(TM) i5-3470 CPU@3.60 GHz, 4.0 GB-RAM. Search parameters of the mLFFA are set according to Yang's recommendations [6,9], i.e., the numbers of fireflies $n = 30$, $\alpha_0 = 0.25$, $\beta_0 = 1$, $\lambda = 1.50$ and $\gamma = 1$. These searching parameters of LFFA are sufficient for most optimization problems because the LFFA algorithm is very robust to parameter variation-dependent [6,9]. The maximum generation Max_Gen = 100 is then set as the TC in each trial. 50 trials are conducted to find a set of the optimal PIDA controllers for the AVR system. In this work, the parameters of the AVR system are set according to [36,37] as follows: $K_A = 10$, $\tau_A = 0.1$ sec., $K_E = 1.0$, $\tau_E = 0.4$ sec., $K_G = 1.0$, $\tau_G = 1.0$ sec., $K_R = 1.0$ and $\tau_R = 0.01$ sec.

After the searching process of the mLFFA over 50 trials stopped, 50 optimal PIDA controllers are successfully obtained and summarized in Table 3 with their corresponding responses. As non-dominated solutions, 50 sets of obtained PIDA controllers are plotted in Figure 8 to formulate the Pareto front and perform trade-off characteristics between $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$.

Tracking (or command following) responses of the AVR system with PIDA controllers are depicted in Figure 9, while regulating (or disturbance rejection) responses of the AVR system with PIDA controllers are plotted in Figure 10. From obtained results, it was found that the optimal PIDA controller's parameters obtained by the proposed mLFFA for the AVR system and their corresponding responses are very satisfactory according to the design constraints defined in (21). For entire results, the mLFFA can successfully provide the optimal PIDA controllers for the AVR system based on the multiobjective optimization. It was found that the proposed mLFFA performs very high performance to

TABLE 3. 50 PIDA controllers designed by mLFFA for AVR system and responses

PIDA No#	**PIDA controller's parameters**				**Responses**			
	K_p	K_i	K_d	K_a	$f_1(\boldsymbol{x})$ t_r (sec.)	$f_2(\boldsymbol{x})$ M_p (%)	t_s (sec.)	E_{ss} (%)
1. → (min f_2)	***0.846024***	***0.539313***	***0.299987***	***0.009998***	***0.290021***	***0.000000***	***0.410000***	***0.000001***
2.	0.838297	0.602389	0.299990	0.009999	0.289876	0.300204	0.400000	0.000000
3.	0.806808	0.613666	0.299988	0.008085	0.288747	0.636751	0.390000	0.000000
4.	0.844065	0.533188	0.300000	0.006392	0.286984	1.307865	1.020000	0.000002
5.	0.779164	0.658077	0.299988	0.009934	0.274957	1.409115	0.450000	0.000001
6.	0.877813	0.660406	0.299984	0.010000	0.271248	1.668555	0.380000	0.000000
7.	0.719457	0.700331	0.299986	0.008362	0.269240	2.020076	2.190000	0.000002
8.	0.964169	0.528245	0.299995	0.009999	0.268210	2.194901	0.580000	0.000052
9.	0.940636	0.653474	0.299996	0.010000	0.266028	3.204826	0.650000	0.000000
10.	0.897606	0.780465	0.299991	0.009999	0.263014	3.253317	0.770000	0.000000
11.	0.891220	0.820574	0.299986	0.009999	0.261005	3.475652	1.440000	0.000000
12.	0.989782	0.541024	0.299983	0.010000	0.259917	3.555930	0.620000	0.000054
13.	0.953455	0.664484	0.300000	0.009999	0.258840	3.618364	0.670000	0.000000
14.	0.966625	0.510907	0.299981	0.008149	0.257701	3.644928	1.200000	0.000078
15. → (min f_1 & f_2)	***0.943977***	***0.723377***	***0.299990***	***0.009999***	***0.256693***	***3.880744***	***0.720000***	***0.000000***
16.	0.926938	0.648208	0.299990	0.007540	0.255581	3.939943	0.630000	0.000000
17.	0.999988	0.562378	0.299988	0.010000	0.254496	3.981911	0.640000	0.000039
18.	0.969642	0.648106	0.299997	0.009776	0.253932	3.988423	0.670000	0.000001
19.	0.813201	0.907979	0.299997	0.009997	0.252140	4.017212	2.030000	0.000001
20.	0.948408	0.638904	0.299986	0.008184	0.251093	4.114728	0.640000	0.000001
21.	0.969620	0.671554	0.299984	0.009725	0.250024	4.202277	0.690000	0.000000
22.	0.999965	0.602366	0.299991	0.009999	0.249011	4.293520	0.670000	0.000015
23.	0.938238	0.815793	0.299991	0.009999	0.249005	4.544896	0.920000	0.000000
24.	0.999975	0.640605	0.299991	0.010000	0.248896	4.593256	0.690000	0.000005
25.	0.999997	0.641186	0.299990	0.010000	0.248546	4.598262	0.690000	0.000005
26.	0.999999	0.641186	0.299990	0.010000	0.248310	4.598427	0.690000	0.000005
27.	0.999964	0.642427	0.299989	0.010000	0.248294	4.607199	0.690000	0.000005
28.	0.958590	0.783677	0.299994	0.010000	0.248015	4.752795	0.820000	0.000000
29.	0.935873	0.861061	0.299987	0.010000	0.247887	4.893529	1.340000	0.000000
30.	0.998341	0.687555	0.299985	0.010000	0.247624	4.928583	0.730000	0.000001
31.	1.000000	0.683109	0.299986	0.010000	0.247420	4.933428	0.720000	0.000001
32.	0.999979	0.696564	0.299994	0.010000	0.247228	5.041405	0.730000	0.000000
33.	0.999949	0.704418	0.299994	0.009999	0.247085	5.104554	0.740000	0.000000
34.	0.957756	0.829153	0.300000	0.009999	0.246602	5.126159	0.950000	0.000000
35.	0.929096	0.904153	0.299986	0.010000	0.246469	5.130600	1.540000	0.000000
36.	0.941326	0.877485	0.299987	0.010000	0.246203	5.167980	1.380000	0.000000
37.	0.999983	0.715371	0.299987	0.010000	0.245872	5.194031	0.750000	0.000000
38.	0.999990	0.739741	0.299987	0.010000	0.245536	5.391459	0.780000	0.000000
39.	0.981187	0.819827	0.299998	0.009978	0.245201	5.609652	0.890000	0.000000
40.	0.999988	0.789630	0.299985	0.010000	0.244854	5.798575	0.830000	0.000000
41.	0.999997	0.801844	0.299987	0.010000	0.244544	5.900466	0.850000	0.000000
42.	0.996214	0.815113	0.299985	0.010000	0.244020	5.921130	0.880000	0.000000
43.	0.999457	0.806589	0.299991	0.009999	0.243696	5.927196	0.860000	0.000000
44.	0.992112	0.844931	0.299989	0.009998	0.243207	6.072431	0.950000	0.000000
45.	0.999966	0.823675	0.299990	0.009999	0.243034	6.081243	0.890000	0.000000
46.	0.999902	0.842957	0.299983	0.010000	0.242669	6.240252	0.930000	0.000000
47.	0.999957	0.844477	0.299997	0.009998	0.242207	6.254190	0.930000	0.000000
48.	0.999971	0.832714	0.299992	0.008906	0.241943	6.629873	0.870000	0.000000
49.	0.999989	0.908276	0.299994	0.009999	0.241079	6.790362	1.160000	0.000000
50. → (min f_1)	***0.999474***	***0.947491***	***0.299989***	***0.010000***	***0.240112***	***7.113321***	***1.310000***	***0.000000***

optimize the PIDA controllers for the AVR system. 50 optimal PIDA controllers obtained by the mLFFA are smoothly distributed though the Pareto front. Referring to Figure 9, three specific results, i.e., $\min f_1(\boldsymbol{x})$, $\min f_1(\boldsymbol{x})$ & $f_2(\boldsymbol{x})$, and $\min f_2(\boldsymbol{x})$, are selected to plot the tracking and regulating responses as visualized in Figure 11 and Figure 12,

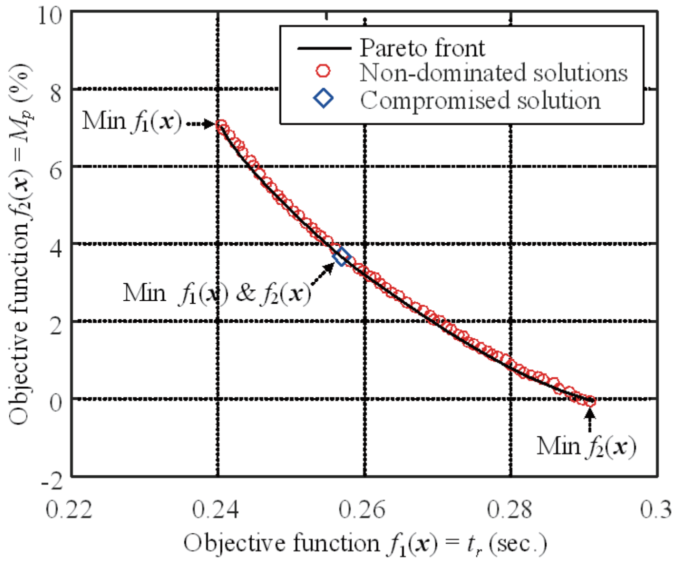

FIGURE 8. Pareto front of 50 PIDA controllers designed by mLFFA for AVR system

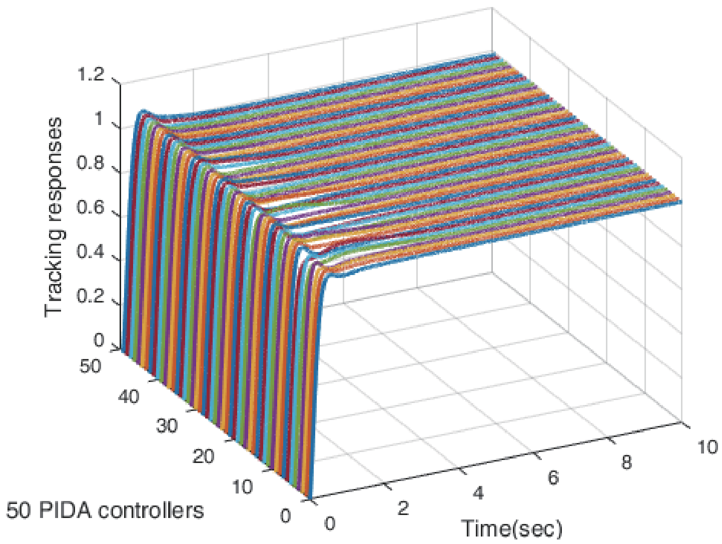

FIGURE 9. Tracking responses of AVR system with PIDA controllers designed by mLFFA

respectively. Results obtained assure the perfectly tread-off characteristics between t_r and M_p that were set as the particular objective functions.

6. **Conclusions.** The multiobjective Lévy-flight firefly algorithm (mLFFA) has been developed and proposed in this paper for solving multiobjective optimization problems. The proposed mLFFA is the one of the modified versions of the Lévy-flight firefly algorithm (LFFA) extended from the original firefly algorithm (FA). To perform its effectiveness, the proposed mLFFA has been validated against four standard multiobjective test functions. As results, the mLFFA could provide very satisfactory solutions for all benchmark test functions. The proposed mLFFA has been applied to design the optimal PIDA controller for the AVR system based on the modern optimization. As simulation results, it was found that optimal PIDA controllers could be successfully obtained by the mLFFA

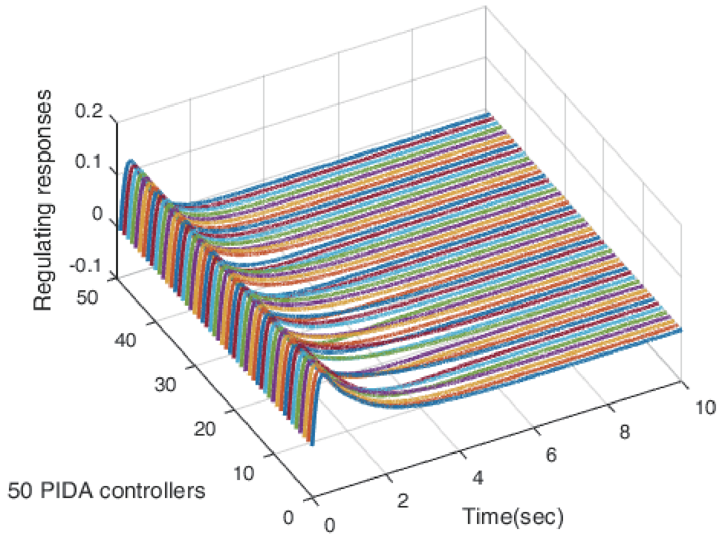

FIGURE 10. Regulating responses of AVR system with PIDA controllers designed by mLFFA

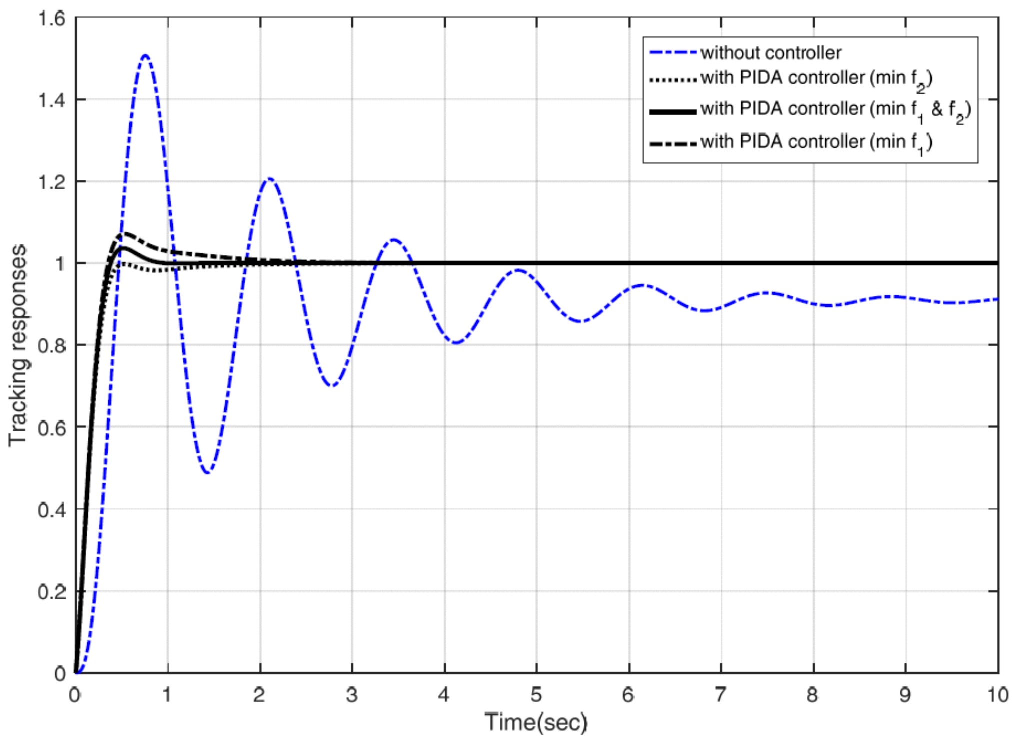

FIGURE 11. Tracking responses of selected min $f_1(\boldsymbol{x})$, min $f_1(\boldsymbol{x})$ & $f_2(\boldsymbol{x})$ and min $f_2(\boldsymbol{x})$ of AVR system with PIDA controllers designed by mLFFA

according to the predefined objective functions associated with predefined constraint functions. As non-dominated solutions of multiobjective optimization problems, 50 optimal PIDA controllers obtained could completely perform the optimal Pareto front assuring the tread-off characteristics between two particular objective functions. Tracking and regulating responses of the AVR controlled system have been successfully produced by the PIDA controllers designed by the proposed mLFFA. For the future research of interests, other real-world multiobjective optimization problems will be investigated based on

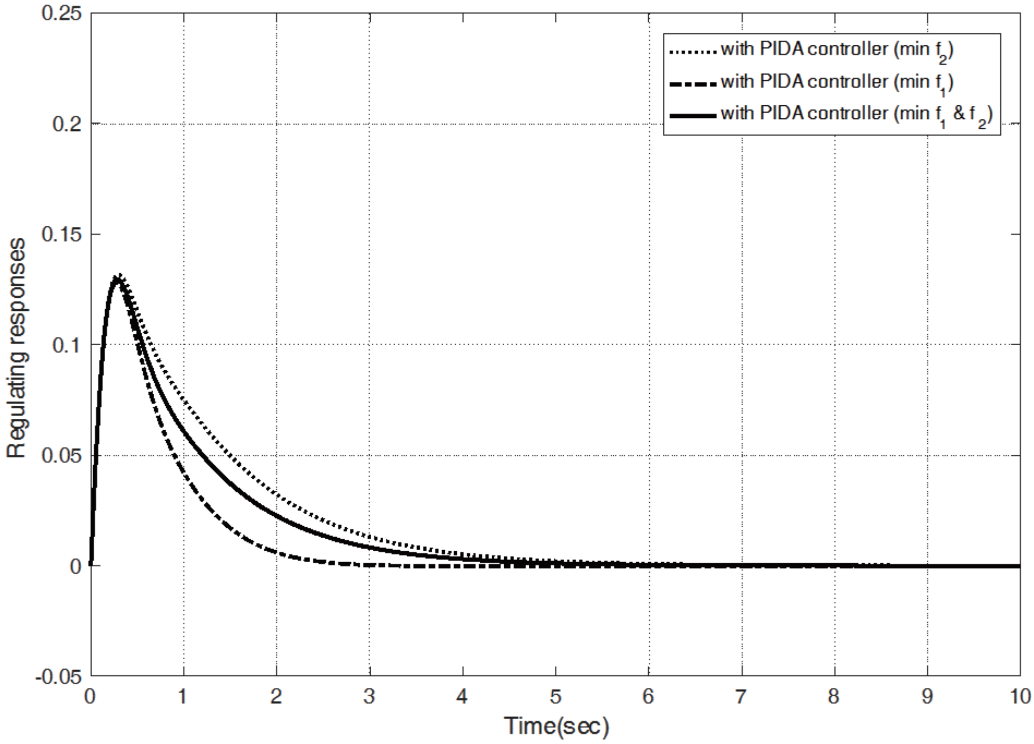

FIGURE 12. Regulating responses of selected min $f_1(\boldsymbol{x})$, min $f_1(\boldsymbol{x})$ & $f_2(\boldsymbol{x})$ and min $f_2(\boldsymbol{x})$ of AVR system with PIDA controllers designed by mLFFA

the mLFFA-based approach. Multiobjective fractional-order PID/PIDA controller design optimization by the mLFFA or other novel and hybrid metaheuristics will be carried on.

REFERENCES

[1] V. Zakian, *Control Systems Design: A New Framework*, Springer-Verlag, 2005.

[2] E. G. Talbi, *Metaheuristics: Form Design to Implementation*, John Wiley & Sons, 2009.

[3] F. Glover and G. A. Kochenberger, *Handbook of Metaheuristics*, Kluwer Academic Publishers, 2003.

[4] D. T. Pham and D. Karaboga, *Intelligent Optimisation Techniques*, Springer, London, 2000.

[5] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, 2008.

[6] X. S. Yang, Firefly algorithms for multimodal optimization, stochastic algorithms, *Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences*, vol.5792, pp.169-178, 2009.

[7] I. Fister, I. Fister, Jr., X. S. Yang and J. Brest, A comprehensive review of firefly algorithms, *Swarm and Evolutionary Computation*, vol.13, pp.34-46, 2013.

[8] I. Fister, X. S. Yang, D. Fister and I. Fister, Jr., Firefly algorithm: A brief review of the expanding literature, *Cuckoo Search and Firefly Algorithm*, vol.347, pp.347-360, 2014.

[9] X.-S. Yang, Firefly algorithm, Lévy flights and global optimization, *Research and Development in Intelligent Systems*, vol.XXVI, pp.209-218, 2010.

[10] M. Song and D. Chen, A comparison of three heuristic optimization algorithms for solving the multi-objective land allocation (MOLA) problem, *Annals of GIS*, pp.1-13, 2018.

[11] Y. Sun, Y. Gao and X. Shi, Chaotic multi-objective particle swarm optimization algorithm incorporating clone immunity, *Mathematics*, vol.7, no.146, pp.1-16, 2019.

[12] H. Jia, Y. Lin, Q. Luo, Y. Li and H. Miao, Multi-objective optimization of urban road intersection signal timing based on particle swarm optimization algorithm, *Advances in Mechanical Engineering*, vol.11, no.4, pp.1-9, 2019.

[13] W. Yamany, N. El-Bendary, A. E. Hassanien and E. Emary, Multi-objective cuckoo search optimization for dimensionality reduction, *Procedia Computer Science*, vol.96, pp.207-215, 2016.

[14] X. S. Yang, M. Karamanoglu and X. Heb, Flower pollination algorithm: A novel approach for multiobjective optimization, *Engineering Optimization*, pp.1-16, 2013.
[15] C. Yang, Q. Li and Q. Chen, Multi-objective optimization of parallel manipulators using a game algorithm, *Applied Mathematical Modelling*, vol.74, pp.217-243, 2019.
[16] W. L. Wang, W. K. Li, Z. Wang and L. Li, Opposition-based multi-objective whale optimization algorithm with global grid ranking, *Neurocomputing*, vol.341, pp.41-59, 2019.
[17] E. G. Talbi, Hybrid metaheuristics for multi-objective optimization, *Journal of Algorithms & Computational Technology*, vol.9, no.1, pp.41-63, 2014.
[18] S. Jung and R. C. Dorf, Analytic PIDA controller design technique for a third order system, *Proc. of the 35th IEEE Conference on Decision and Control*, Kobe, pp.2513-2518, 1996.
[19] S. Sornmuang and S. Sujitjorn, GA-based PIDA control design optimization with an application to AC motor speed control, *International Journal of Mathematics and Computers in Simulation*, vol.4, no.3, pp.67-80, 2010.
[20] D. Puangdownreong and S. Suwannarongsri, Torsional resonance suppression via PIDA controller designed by the particle swarm optimization, *Proc. of the ECTI-CON International Conference*, pp.673-676, 2008.
[21] D. Puangdownreong, Application of current search to optimum PIDA controller design, *Intelligent Control and Automation*, vol.3, no.4, pp.303-312, 2012.
[22] D. Puangdownreong, S. Sumpunsri, M. Sukchum, C. Thammarat, S. Hlangnamthip and A. Nawikavatan, FA-based optimal PIDA controller design for AVR system, *Proc. of the iEECON2018 International Conference*, pp.548-551, 2018.
[23] C. Thammarat, K. Lurang, D. Puangdownreong, S. Suwammarongsri, S. Hlangnamthip and A. Nawikavatan, Application of bat-inspired algorithm to optimal PIDA controller design for liquid-level system, *Proc. of the iEECON2018 International Conference*, pp.556-559, 2018.
[24] K. Lurang, C. Thammarat, S. Hlangnamthip and D. Puangdownreong, Optimal design of two-degree-of-freedom PIDA controllers for liquid-level system by bat-inspired algorithm, *International Journal of Circuits, Systems and Signal Processing*, vol.13, pp.34-39, 2019.
[25] T. Niyomsat and D. Puangdownreong, Optimal PIDA load frequency controller design for power systems via flower pollination algorithm, *WSEAS Trans. Power Systems*, vol.14, pp.1-7, 2019.
[26] N. Pringsakul, D. Puangdownreong, C. Thammarat and S. Hlangnamthip, Obtaining optimal PIDA controller for temperature control of electric furnace system via flower pollination algorithm, *WSEAS Trans. Systems and Control*, vol.14, pp.1-7, 2019.
[27] H. Neisser, A note on Pareto's theory of production, *Econometrica*, vol.8, pp.253-262, 1940.
[28] F. Y. Edgeworth, *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*, C. Kegan Paul and Co., London, 1881.
[29] C. Yunfang, A general framework for multi-objective optimization immune algorithms, *International Journal of Intelligent Systems and Applications (IJISA)*, vol.4, no.6, pp.1-13, 2012.
[30] E. Zitzler and L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, *IEEE Trans. Evolutionary Computation*, vol.3, pp.257-271, 1999.
[31] E. Zitzler, K. Deb and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolution Computing*, vol.8, pp.173-195, 2000.
[32] J. D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, *Proc. of the 1st International Conference on Genetic Algorithms*, pp.93-100, 1985.
[33] K. Deb, A. Pratap, S. Agarwal and T. Mayarivan, A fast and elitist multiobjective algorithm: NSGA-II, *IEEE Trans. Evolutionary Computation*, vol.6, pp.182-197, 2002.
[34] T. Robič and B. Filipič, DEMO: Differential evolution for multiobjective optimization, *Lecture Notes in Computer Sciences*, vol.3410, pp.520-533, 2005.
[35] D. Puangdownreong, Multiobjective multipath adaptive tabu search for optimal PID controller design, *International Journal of Intelligent Systems and Applications*, vol.7, no.8, pp.51-58, 2015.
[36] Z. L. Gaing, A particle swarm optimization approach for optimum design of PID controller in AVR system, *IEEE Trans. Energy Conversion*, vol.19, no.2, pp.384-391, 2004.
[37] A. Nawikavatan, S. Tunyasrirut and D. Puangdownreong, Application of intensified current search to optimum PID controller design in AVR system, *Lecture Notes in Computer Science*, pp.255-266, 2014.