SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2021, Vol. 64 112208:1–112208:13
https://doi.org/10.1007/s11432-019-2749-0

A population randomization-based multi-objective genetic algorithm for gesture adaptation in human-robot interaction

Luefeng CHEN[1,2*], Wanjuan SU[1,2], Min LI[1,2], Min WU[1,2],
Witold PEDRYCZ[3] & Kaoru HIROTA[4]

[1] *School of Automation, China University of Geosciences, Wuhan* 430074*, China;*
[2] *Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan* 430074*, China;*
[3] *Department of Electrical and Computer Engineering, University of Alberta, Edmonton* AB T6R 2G7*, Canada;*
[4] *Tokyo Institute of Technology, Yokohama* 226-8502*, Japan*

Received 18 July 2019/Revised 24 September 2019/Accepted 2 December 2019/Published online 24 December 2020

Abstract In recent years, vision-based gesture adaptation has attracted great attention from many experts in the field of human-robot interaction, and many methods have been proposed and successfully applied, such as particle swarm optimization and genetic algorithm. However, the reduction of the error and energy consumption of a robot while paying attention to more subtle attitude changes is very important and challenging. In view of these problems, we propose a population randomization-based multi-objective genetic algorithm. The gesture signal is processed with a slight change by imitating the biological evolution mechanisms. In the proposed algorithm, a random out-of-order matrix is added in the process of population evolution synthesis to prevent the premature grouping convergence of the new population. The weights of the objective function and the elite retention strategy are adopted, and the most adaptable individuals in each generation are inherited directly in the next generation without any recombination or mutation. To verify the effectiveness of the algorithm, preliminary application experiments are performed on the gesture adaptation of a robotic arm. The results are compared with the original signal, and the comparison shows that by using the proposed method, the energy consumption is reduced, and the end error is decreased to less than 3 mm while ensuring the tracking effect of the robotic arm. These obtained results meet the communication requirements for human-robot interactions such as handshakes. Moreover, the proposed method has better performance, uses less energy, and has a smaller tracking error than the particle swarm optimization, the single-objective genetic algorithm, and the traditional multi-objective genetic algorithm. A preliminary application experiment indicates that the robotic arm can adapt to human gestures in real time.

Keywords multi-objective genetic algorithm, gesture adaptation, human-robot interaction

Citation Chen L F, Su W J, Li M, et al. A population randomization-based multi-objective genetic algorithm for gesture adaptation in human-robot interaction. Sci China Inf Sci, 2021, 64(1): 112208, https://doi.org/10.1007/s11432-019-2749-0

1 Introduction

The human-robot interaction (HRI) has become a vibrant research area involving the systems, humans, and robots [1,2]. Currently, many researchers have focused their attention on gesture adaptation, and the related research can be easily applied to HRI, such as the online robot teaching [3], gesture recognition [4], and emotional interaction [5,6]. Because of its excellent performance, vision-based gesture adaptation technology has attracted the attention of many experts in HRI [7,8]. Hence, there is a great demand for a high-performance gesture adaptation method [9–11].

Many studies on gesture adaptation can be found in the literature. For instance , Kılıboz et al. [12] presented a method to detect and recognize the trajectory-based dynamic hand gestures in real time where the gestures were represented as an ordered sequence of directional movements, and a fast learning mechanism on gesture trajectory was used. An adaptive hidden Markov model-based gesture recognition

[*] Corresponding author (email: chenluefeng@cug.edu.cn)

method with user adaptation was presented in [13], where the Kinect sensor was used for obtaining the gesture signals from a user. Then, the gesture action was recognized and used as a control command for driving a humanoid robot to imitate human's actions. In [14], a hybrid approach based on the mutual adaptation for human gesture recognition was proposed, where a neuro-fuzzy system was used to classify human gestures, and an evolution strategy was applied for tuning the parameters and pruning the membership functions. The gesture adaptation has a wide application prospect, and it can be used in the visual tracking of arms [15], interactive games [16], and robot assistance for getting dressed [17]. Moreover, by using the gestures [18,19], human behavior can be analyzed and people's intentions can be deeply understood [20,21].

Recently, objective optimization algorithms [22], e.g., genetic algorithms (GAs) [23] and particle swarm optimization (PSO) algorithm [24], have received a great attention. According to the optimization type, problems can be classified into discrete problems and continuous problems [25]. In [25], a predatory search strategy, the PSO, was proposed to solve continuous variable optimization problems. On the other hand, the GA has the potential to address continuous problems because it has many similarities with the PSO. Thus, the GA can also be used to solve complex optimization models that contain both discrete and continuous variables; for instance, a modified non-inferior stratified genetic algorithm-III (NSGA-III) for large-scale optimization problems was proposed in [26]. In [27], a GA approach for the real-time motion tracking of redundant and non-redundant manipulators was introduced. By applying the genetic operators to a set of suitably generated configurations, the end effector can accurately follow the desired workspace trajectory. Additionally, a multi-objective genetic algorithm (MOGA) for optimally finding the dimensions of links and joint angle of a robot gripper was presented in [28]. In addition, many researchers used the advantage of the GAs to solve non-linear problems, which shows that the GA is effective in engineering applications [29,30]. Besides, the GA has been shown to be capable of simultaneously evaluating many points in the parameter space and converging toward the global solution. Therefore, the GA is chosen to solve the optimization problem in this paper.

However, many real-world applications, such as the gesture adaptation task [31], involve multiple objectives that are conflicting and affecting each other [32]. Consequently, on the basis of the single-objective genetic algorithm (SOGA), the MOGA was developed in [28], which can be defined as finding a decision vector that satisfies the constraints and optimizes a phasor function. The core of the MOGA is to coordinate the relationship between the objective functions and find an optimal solution set of the objective functions to the greatest possible extent. The MOGA allows evolutionary groups to search multiple targets in parallel, and gradually finds the optimal solution to the problem. In addition, the MOGA has played an important role in the field of multiarticular robotic arm tracking for the movement of the human arm [33,34].

The MOGA starts from a randomly generated population and then progresses by performing the operations of selection, crossover, and mutation through several generations of evolution during which the adaptation of individuals in a population continues to increase and approaches the Pareto optimal solution set gradually [35,36]. In [37], we presented the results of a pilot study conducted with the aim to optimize building energy performance using the MOGA, i.e., the evolutionary adaptive approach. This study concluded that the MOGA is an appropriate approach that can ensure a better global optimal solution for the design of energy-efficient buildings. The position of the robotic arm end is different from the target position. Thus, we not only need to reduce the energy consumption of a general robotic arm system but also to increase its precision. In order to solve these problems, the population randomization multi-object genetic algorithm (PRMOGA) is proposed. The proposed algorithm merges all the generated subpopulations into a complete population the same as the traditional multi-objective genetic algorithm (TMOGA), but the PRMOGA adopts a stochastic matrix to make the new population out-of-order randomly. Meanwhile, we select the elitist individual in each generation and put it directly into the next generation. It has been proven by experiments that an elite reservation strategy and random population sorting can improve the performance of our model. Moreover, the objective functions are designed to consider not only the output error but also the weight of a robotic arm and joint angles of human arms. These considerations are beneficial for adapting to the human arm gesture and reducing the robot's energy consumption.

Vision-based gesture adaptation has attracted great attention in the field of human-robot interaction, and many methods have been introduced, such as PSO and GA. However, the gesture adaptation task represents a multi-objective problem. Among the multi-objective optimization algorithms, PSO has a limited ability to explore new space and can easily converge to a local optimal solution. Although the GA

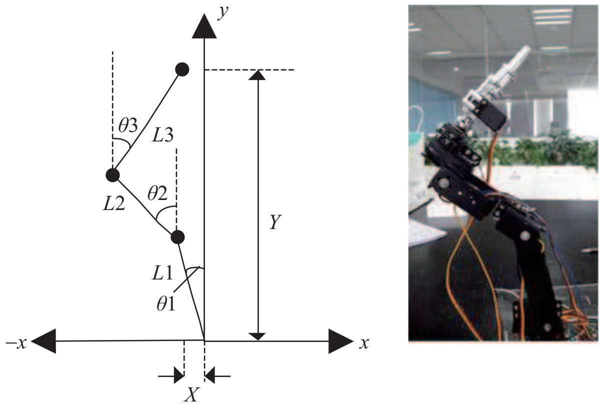

Figure 1 (Color online) Kinematic model of a human arm gesture.

can calculate multiple points in the parameter space simultaneously, it can also converge prematurely. In order to coordinate the relationship between the objective functions, maximize the search for an optimal solution set of the objective functions, and avoid premature convergence of the GA, the MOGA algorithm is adopted. However, the traditional MOGA algorithm combines all generated subpopulations into a complete population, which results in a large computing overhead and a large end error. Therefore, it is needed to reduce the error and energy consumption, and shorten the calculation time.

The PRMOGA is proposed to quickly search for an optimal solution for the three angles of robotic arm joints as shown in Figure 1, in which the objective functions and elitist generation are developed by considering the human gesture information, and in such a way that the solution conforms to the object end position and adapts to the movement of a human arm. In combination with the closed-loop feedback control from the angle sensor, this method greatly improves the control precision of a robotic arm, and the system response is also ameliorated. Finally, the precise control of the robotic arm is realized. The PRMOGA is verified by a preliminary application experiment of a robotic arm adapting to a human arm gesture. A scene is executed by a user using a Kinect sensor, a robotic arm, and PCs (personal computers). The right arm of the user is used as the original model for the Kinect to acquire the gesture data, and the robotic arm is used to adapt to the right arm of the user. By considering the three objective functions, the end error, tracking error, and energy consumption of the manipulator are optimized globally.

The rest of the paper is organized as follows. In Section 2, a multi-objective genetic algorithm and robotic arm-based gesture adaptation are presented. The experiments are introduced in Section 3. The conclusion and prospects are given in Section 4.

2 Multi-objective genetic algorithm and robotic arm-based gesture adaptation

The development of HRI is gradually characterized by multi-modal perception and smooth interaction. The existing posture recognition optimization methods in the HRI seldom consider the end error, tracking error, and energy consumption simultaneously. However, the proposed method can find the global optimal solution by considering these three objective functions.

This paper proposes a population randomization-based multi-objective genetic algorithm for robots adapted to human arm gestures that can realize flexible adaptation and certain positioning behavior. First, the Kinect is employed to recognize human gestures and obtain feature points of the recognized human gesture. Then, the kinematic model of the human arm gesture is built, as shown in Figure 1. In the optimization of joint angles, a method of group randomization using the parallel selection method is adopted to design the adaptive PRMOGA of robot posture better. Finally, by delivering the joint information, the robot can smoothly adapt to the human arm gesture.

Table 1 Parameters of the human arm gesture kinematic model

Joint	Length (cm)	Angle	Weight (kg)
1	17	$-\pi/2 - \pi/2$	0.8
2	10	$-\pi/2 - \pi/2$	0.6
3	17	$-\pi/2 - \pi/2$	0.4

2.1 Population randomization-based multi-objective genetic algorithm

The population randomization method utilizing the parallel selection method called the PRMOGA is proposed to design the robot gesture adaptation. The implementation steps of the PRMOGA are shown as follows:

Step 1: Determine the rotation angle of the three joints deviating from the vertical axis: $\theta_i = (\theta_1, \theta_2, \theta_3)$, and the end position of the robotic arm: $X = \sum_{i=1}^{3} (L_i \times \sin\theta_i)$, $Y = \sum_{i=1}^{3} (L_i \times \cos\theta_i)$, where L_i denotes the joint length of a robotic arm.

Step 2: Determine the three sub-objective functions f_1, f_2, and f_3.

Step 3: Encode the joint rotation angle θ_i using the traditional genetic algorithm, and obtain an individual S, $S = (\theta_1, \theta_2, \theta_3)$.

Step 4: Using the three sub-objective functions, calculate the fitness of each individual in each group, and retain the highest fitness individual in the next generation.

Step 5: Choose the two individuals S_1 and S_2, and generate the progeny populations using the genetic cross mutations.

Step 6: Judge whether the fitness of individuals in the population converges. If so, the final Pareto optimal solution is obtained; otherwise, go to Step 4.

In the parallel selection method, all individuals in a population are divided into several subpopulations according to the number of sub-objective functions. Each subpopulation is independently assigned to a sub-objective function, and each sub-objective function is used to select good individuals in its corresponding subpopulation. In order to maintain the population diversity, when a new subpopulation is formed, all the generated subpopulations are merged into a complete population, and then the random matrix is used to order the new population randomly. Finally, the crossover and mutation operations are applied to determine the final Pareto optimal solution $(\theta_1, \theta_2, \theta_3)$ of the problem by iterating until the preset number of generation is reached.

Meanwhile, we select the best solution called the elitist individual during each generation, and use it in the next generation without conducting the crossover and mutation operations. The experimental results show that the elite reservation strategy and random population sorting improve system performance.

2.2 Robot gesture adaptation using population randomization-based multi-objective genetic algorithm

The kinematic model of a human arm gesture including six degrees of freedom is established, and the parameters are determined, as shown in Figure 1 and given in Table 1. The human arm gesture is simplified to three joints, so the number of chromosomes in the gene is also three. The range of the joint angle is $[-\pi/2, \pi/2]$, and each chromosome consists of 18 bits. The total number of entries of the chromosomes of each individual is 54. The coordinates of the end position of the robotic arm are (X, Y), as shown in Figure 1.

The three chromosomes of an individual can be expressed as

$$\theta_i = \theta_i(\text{low}) + \frac{\text{Rind}_i}{2^{18} - 1}\theta_i(\text{up}) - \theta_i(\text{low}), \tag{1}$$

where the resolution of a joint angle (θ_i) is expressed as $(\theta_i(\text{up}) - \theta_i(\text{low}))/(2^{18} - 1)$. The mechanical arm is powered by batteries, and energy consumption is a major constraint. In practical applications, due to different connections between the joints, the energy consumed by the joints is different, even when they turn by the same angle. On the premise of ensuring that the desired joint angle value can make a robotic arm achieve the desired position (x_0, y_0), the mechanical arm should consume the lowest energy. In the process of a mechanical arm movement, the potential energy changes with $L1$, $L2$ and $L3$.

The objective function f_1 indicates the change in joint potential energy, which is expressed as

$$\begin{aligned} f_1 = {} & \frac{L_1\cos\theta_1}{2} \times G_1 + \left(L_1\cos\theta_1 + \frac{L_2\cos\theta_2}{2}\right) \times G_2 \\ & + \left(L_1\cos\theta_1 + L_2\cos\theta_2 + \frac{L_3\cos\theta_3}{2}\right) \times G_3, \end{aligned} \tag{2}$$

where G_1, G_2, and G_3 denote the weights of the three section joints, respectively; L_1, L_2, and L_3 are three different distances, as shown in Figure 1.

The objective function f_2 describes the error between the expected position and theoretical position of the mechanical arm's end, and this error is given by

$$e = \sqrt{(X - x_0)^2 + (Y - y_0)^2}, \tag{3}$$

where (x_0, y_0) denote the actual position coordinates of the end of a human arm gathered by the genetic training PC; (X, Y) denote the position of the mechanical arm's end, and they are expressed as

$$X = \sum_{i=1}^{3} (L_i \times \sin\theta_i), \quad Y = \sum_{i=1}^{3} (L_i \times \cos\theta_i). \tag{4}$$

To make the method converge faster, a penalty function f_2 is used, e is the threshold value for the current joint end of a robotic arm and the expected end of the robotic arm. In the experiment, we set $e_{\max} = 100$.

$$f_2 = e(0 < e < e_{\max}) + 100 \times e(e > e_{\max}). \tag{5}$$

Furthermore, the arm angle deviation is employed as an objective function f_3 so that the robotic arm can imitate the human arm better and can reach the end position accurately while consuming the least possible energy. The three angles for which the system consumes the least energy are θ_1^o, θ_2^o, and θ_3^o. The energy consuming function f_3 is defined as

$$f_3 = \sqrt{(\theta_1 - \theta_1^o)^2 + (\theta_2 - \theta_2^o)^2 + (\theta_3 - \theta_3^o)^2}. \tag{6}$$

In order to observe the average level of the population, and not just the high-fitness individuals, each sub-objective function is independently applied to select a good individual [21]. All the generated subpopulations are merged into a complete population. Also, the newly formed population is a matrix arranged in rows. We select the first element of each row in the newly formed population matrix to form a new matrix. By randomly arranging the new matrix without repetition, the required stochastic matrix can be obtained. Then, the rows of the new generation are taken out and rearranged row by row according to the rows specified in the stochastic matrix. Finally, the new population that is randomly sorted is obtained.

3 Experiments using population randomization-based multi-objective genetic algorithm

3.1 Experimental settings

The gesture adaptation system includs a user, a Kinect for obtaining the gesture data, a data-processing PC used for processing the gesture information, a PC for training the PRMOGA, and a robotic arm with six degrees of freedom. The gesture adaptation system is presented in Figure 2.

The Kinect is applied to acquire the 3D coordinates of the joints of the human arm. In the data-

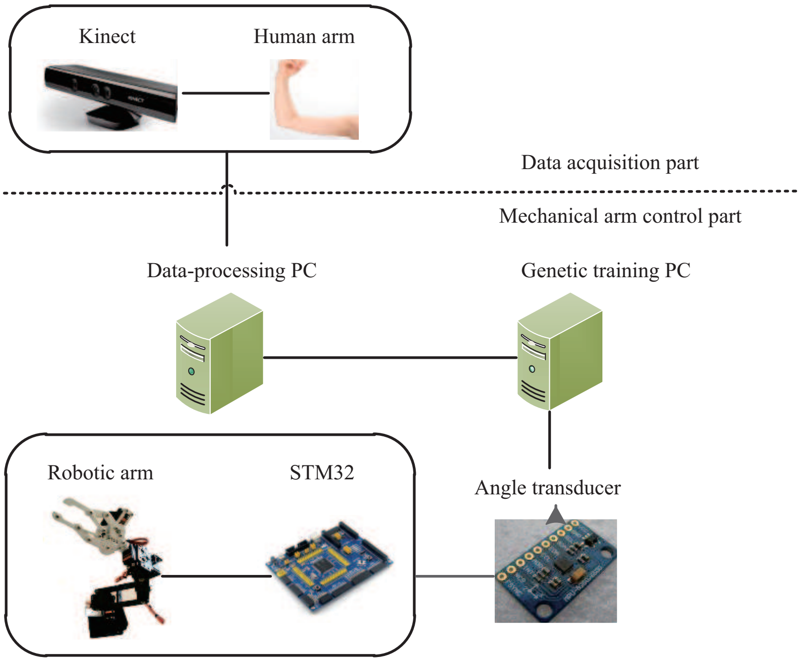

Figure 2 (Color online) The gesture adaptation system.

processing PC, the 3D coordinates are converted into the rotation angles of the joints, given by

$$\begin{aligned}
a &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2},\\
b &= \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2},\\
c &= \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2},\\
\theta' &= \arccos\left(\frac{a^2 + b^2 + c^2}{2ab}\right),\\
\theta &= 180 - \theta'.
\end{aligned} \tag{7}$$

The data of the rotation angle of the robotic arm's joints are converted into angle commands in the data-processing PC. In order to facilitate and speed up the transmission through an 8-bit serial port, the range of the angle command is set to 0–255. The angle command is configured directly in the timer, and the control signal is generated by controlling the interrupt of the timer. The rotation angle is directly proportional to the duty cycle of the pulse-width modulation (PWM) wave. Thus, the required angle command is obtained by

$$\text{Angle\_com} = \frac{\theta \times 200}{\pi} + 150. \tag{8}$$

In what follows, it is explained how the angle commands are obtained by the data-processing PC and how the results are transmitted to the genetic training PC for the PRMOGA. The STM32 is employed to receive the angle commands from the genetic training PC. The genetic training PC obtains the angle command from the data-processing PC through a serial port and sets it as the timer interrupt time. According to the angle command, the control signal is generated, and the corresponding steering mechanism of the robotic arm's joint is used to rotate the arm. The steering gear is powered by a 7.4 V-3000 mA lithium battery with a rotation range of 0°–180°. The degree of rotation is determined by the control signal input from the control port to rotate it to the position required by the command from the genetic training PC.

After that, the three-axis acceleration data x, y, z of the ADXL345 sensors installed in the joint robotic arm are obtained, and the change in the triaxial angle is calculated. After initial filtering by the median algorithm, the data are fed back to the data-processing PC. The error value $e(t)$, which denotes the difference between the angle command and the rotating position of the steering gear, is calculated, and

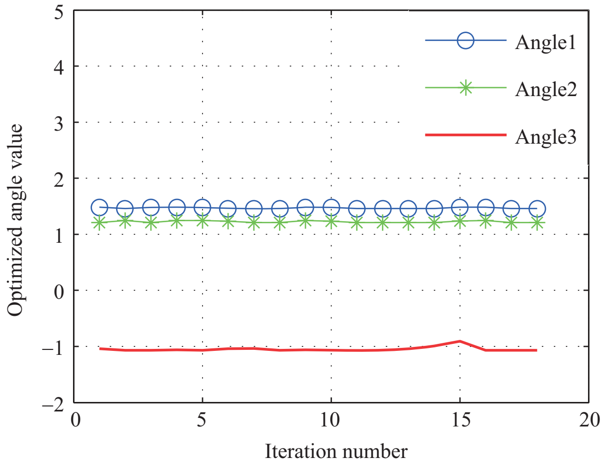

Figure 3 (Color online) Optimal angle values of the three joints.

it then is used as the proportion-integration-differentiation (PID) input. After the feedback value of the sensor became stable, the ratio, integration, differential, and the other three parameters are adjusted, and the corresponding compensation value $u(t)$ is obtained by the PID algorithm. The compensation value is added to the original angle instruction as the next angle command and is then sent to the genetic training PC until the compensation value become zero. The PID algorithm control law can be expressed as

$$u(t) = K_{\mathrm{p}} \left[e(t) + \frac{1}{T} \int_0^t e(t)\mathrm{d}t + T_d \frac{\mathrm{d}e(t)}{\mathrm{d}t} \right], \tag{9}$$

where K_{p} denotes the proportional coefficient of a PID system, T denotes the integration time, T_d denotes the differential time, $e(t)$ denotes the deviation signal of the input system, and $u(t)$ denotes the output signal. $e(t)$ and $u(t)$ stand for the difference values between the angle command and the rotating position of a steering gear, respectively.

3.2 Simulation experiments of population randomization-based multi-objective genetic algorithm

The basic parameters of the PRMOGA model are as follows: population size NIND = 75, the maximum number of generations MAXGEN = 50, number of variables NVAR = 3, binary digit PRECI = 18, generation gap (crossover probability) GGAP = 0.9, and mutation probability Pm = 0.013.

The desired position of the robotic arm is a random point on a plane; (x_0, y_0) is calculated by a given angle $(\theta_1^o, \theta_2^o, \theta_3^o)$ using Eq. (4) and the unit is cm. In the experiment, we set $(\theta_1^o, \theta_2^o, \theta_3^o)$ as (1.4516, 1.1898, −1.0741).

As shown in Figure 3, after 50 computing generations, the inverse solution of the robotic arm's joint angle includes 18 combination groups. Considering the shortcomings of the genetic algorithm after the population merging process, which requires numerous computations and high energy consumption, we adopt the elite reservation strategy. We select the best individuals in each generation and directly put them into the next generation. We calculate the fitness values of each individual and transfer the best one to the data-processing PC. The final results shown in radians are obtained as $(\theta_1^o, \theta_2^o, \theta_3^o) = (1.4525, 1.1919, -1.0632)$.

To demonstrate the effectiveness of the proposed algorithm, several sets of angle data are tested, and the results are shown in Table 2. In Table 2, $R1$ denotes the optimized energy-to-original energy ratio, $R2$ is the end position error expressed in cm, and $R3$ is the tracking error.

As shown in Table 2, the results produced by the multi-objective genetic algorithm have a low tracking error, and energy consumption has decreased. And the end error of the robotic arm is only 3 mm. Thus, the proposed algorithm plays a significant role in reducing energy consumption and the position error of the terminal while tracking the motion trajectory of the arm. Moreover, the position error of 3 mm is satisfactory for smooth communications in HRI, such as handshakes, embraces, and waving. The position error is also satisfactory for performing more complicated actions, e.g., carrying a plate.

Table 2 Verification results of angle data produced by the multi-objective genetic algorithm

$(\theta_1^o, \theta_2^o, \theta_3^o)$	$(\theta_1, \theta_2, \theta_3)$	$R1$	$R2$	$R3$
$(1.4516, 1.1898, -1.0741)$	$(1.4525, 1.1919, -1.0632)$	0.9997	0.3106	0.0896
$(1.4686, 1.0338, -1.3264)$	$(1.4853, 1.0282, -1.3178)$	0.9511	0.0990	0.0196
$(0.7142, -0.1139, -0.6549)$	$(0.7375, -0.1496, -0.6615)$	0.9850	0.4062	0.0432
$(-0.2800, -0.4277, -0.3330)$	$(-0.2987, -0.4233, -0.3254)$	0.9967	0.1459	0.0207
$(1.5347, -0.5657, -1.2554)$	$(1.5677, -0.6260, -1.1946)$	0.8941	0.1555	0.0918
$(-0.9016, 1.0417, -0.8502)$	$(-1.0350, 0.9896, -0.7193)$	0.9032	0.1482	0.1940
$(1.0270, 0.3269, 0.5976)$	$(1.0589, 0.4945, 0.4887)$	0.9571	0.2867	0.2024
$(0.6138, -0.2058, -0.7063)$	$(0.6396, -0.3152, -0.6449)$	0.9852	0.1600	0.1280
$(1.5058, -0.7557, -0.2540)$	$(1.5230, -0.7449, -0.2590)$	0.9637	0.2399	0.0209
$(0.7108, 0.2231, -1.2131)$	$(0.7408, 0.1680, -1.2036)$	0.9859	0.1289	0.0635
$(-0.4804, -0.8168, -0.7634)$	$(-0.5372, -0.7622, -0.7419)$	0.9881	0.2545	0.0817
$(1.0961, -1.1386, -1.2963)$	$(1.116, -1.1493, -1.2733)$	0.9705	0.2266	0.0328
$(-1.1527, 1.4562, -0.3654)$	$(-1.1864, 1.4285, -0.3434)$	0.9619	0.1530	0.0488
$(-1.1481, 0.3165, 1.1845)$	$(-1.1871, 0.3711, 1.1546)$	0.9498	0.3303	0.0735

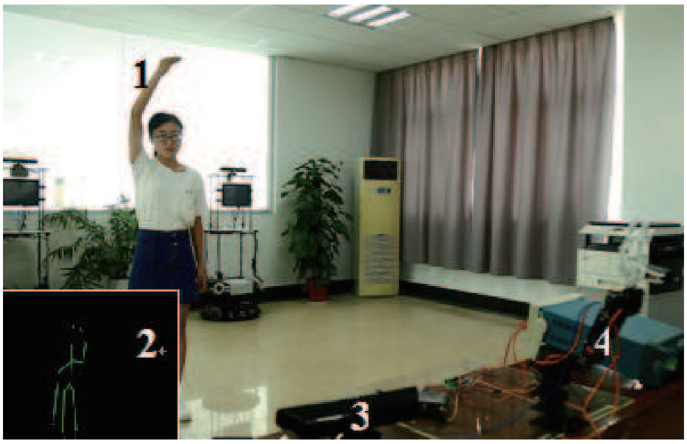

Figure 4 (Color online) Experimental environment. 1: the human arm, 2: the data-processing PC used to obtain the gesture data, 3: Kinect, 4: robotic arm.

In addition, the design of the objective function in PRMOGA considers not only the output error but also the weight of the manipulator and the joint angle of the human arm. In the proposed method, the solution (minimum) and average value of the objective functions f_1, f_2, and f_3 are changed. The experimental results are shown in Figure 3, where it can be seen that with the increase in the number of iterations, the minimum value and average value of the objective function decrease, indicating that the population is evolving continuously, which greatly improves the control accuracy of the manipulator and system response. Meanwhile, during the experiment, we constantly adjust the angles of the three joints to minimize the energy consumption of the robotic arm. In the practical applications of the human-computer interaction system, this method has the advantages of good real-time performance and low energy consumption.

3.3 Preliminary application experiment

The gesture data is collected from a 20-year-old Chinese girl by moving her right arm vertically, flexibly, and horizontally. The experimental environment is shown in Figure 4.

During the gesture change, the data-processing PC obtains the gesture data from the Kinect sensor and transmits the angle data of all the joints to the genetic training PC. At the same time, the angular sensor feeds the actual angle of the robotic arm to the genetic training PC. The feedback angle data and the individual data in the PRMOGA process are used to compensate for the next action. Then, the best individual is selected so that the robotic arm perfectly follows the human arm's movement. Thus, the gesture adaptation is improved.

To prove the advantages of the proposed algorithm, the PSO, the SOGA, and the TMOGA are selected as the compared algorithms. First, for the PSO and SOGA, the target function is set as

$$f = \omega_1 \times f_1 + \omega_2 \times K_1 \times f_2 + \omega_3 \times K_2 \times f_3, \tag{10}$$

where ω_1, ω_2, and ω_3 denote the weights used to combine the objective functions. According to the HRI environment, the energy consumption (for saving energy), terminal error (ensuring touching), and hand signal tracking function (for comfort) are all important for the communication, so the weights are all set to 0.33. Also, K_1 and K_2 are set to 10 and 100 to ensure that the results of f_1, f_2, and f_3 are of the same order of magnitude. In that way, we can draw the conclusion easily.

Simultaneously, we use the TMOGA in another comparison. In the TMOGA, the sub-objective functions are F_1 and F_2. First, in the simple genetic algorithm, we set the target function as follows:

$$\begin{aligned} &F_1 = f_1 + K \times f_3, \\ &F_2 = f_2, \end{aligned} \tag{11}$$

where F_1 represents the sum of the changes in joint potential energy and angle deviation or arm imitation function. The function F_2 is the same as f_2 which we used in the proposed PRMOGA.

However, there are certain differences in the processing procedure between the two algorithms. The TMOGA uses the parallel selection method, and all the individuals in the population are divided equally into several subpopulations according to the number of sub-objective functions. Each subpopulation is assigned to a sub-objective function, and each sub-objective function is used to select good individuals in the corresponding subpopulation independently. Then, a new subpopulation is formed, and all the generated subpopulations are merged into a complete population directly. However, the population sequence remains unchanged because the angle solutions only adapt to the function which they belong to, and they may not be well suited for another function. For the PRMOGA, two strategies are added. They are the new population random disorder strategy in each generation and the elitist reservation strategy. By selecting the best solution in each generation as the elitist individual and using it into the next generation directly, the probability of generating the best offspring individual is increased.

In each generation, we select the best solutions of all the four different algorithms and then compare them. The results are shown in Figure 5. According to the obtained results, the angle obtained by the PRMOGA is the closest to its initial value.

Additionally, we also compare the variance of each angle selected by the four algorithms to further analyze our results. As shown in Figure 6, the PRMOGA can achieve a smallest variance value.

Finally, we put the best angles of each generation calculated by the four algorithms separately into the same energy consumption function, end position function, and imitation function. The obtained results are shown in Figure 7. The presented results prove that the PRMOGA yields the smallest energy consumption, the lowest end position error, and the best imitation effect.

Meanwhile, we compare the results calculated by the four different algorithms, and they are shown in Table 3. In Table 3, it can be seen that the PRMOGA achieves better results than the other methods. The energy consumption, end error, and tracking error of the PRMOGA are the smallest. In Table 3, the computation time of each algorithm is provided, and it can be seen that the computation times of the PSO, the SOGA, the TMOGA, and the PRMOGA are 0.043449, 0.066131, 0.069721, and 0.073386 s, respectively. Although the proposed algorithm has the longest computation time, the computation time is still at the sub-second level, so the accuracy of the real-time tracking is in the acceptable range. According to the HRI environment, since the energy consumption (for saving energy), end error (ensuring touching), and gesture tracking (for comfort) function are all important for communication performance, the PRMOGA can be considered as the most suitable method for practical application in the HRI systems among all the several tested methods.

To further verify the performance of our proposed method, we complete the replication of the two algorithms proposed recently and compare the experimental results; the results are provided in Table 4. In [38], an end point positioning method was used for continuous gestures recognition (CHG-EP), and in [39] an adaptive gradient multi-objective PSO method (AGMOPSO) was proposed. Compared with the method proposed in [38], the PRMOGA consumes significantly less energy. Compared with the method proposed in [39], the proposed method has a smaller end error and tracking error. Although the PRMOGA's advantage in computing time is not very prominent, compared with the other two methods, the PRMOGA performs well regarding the tracking error, end error, and energy consumption. Moreover,

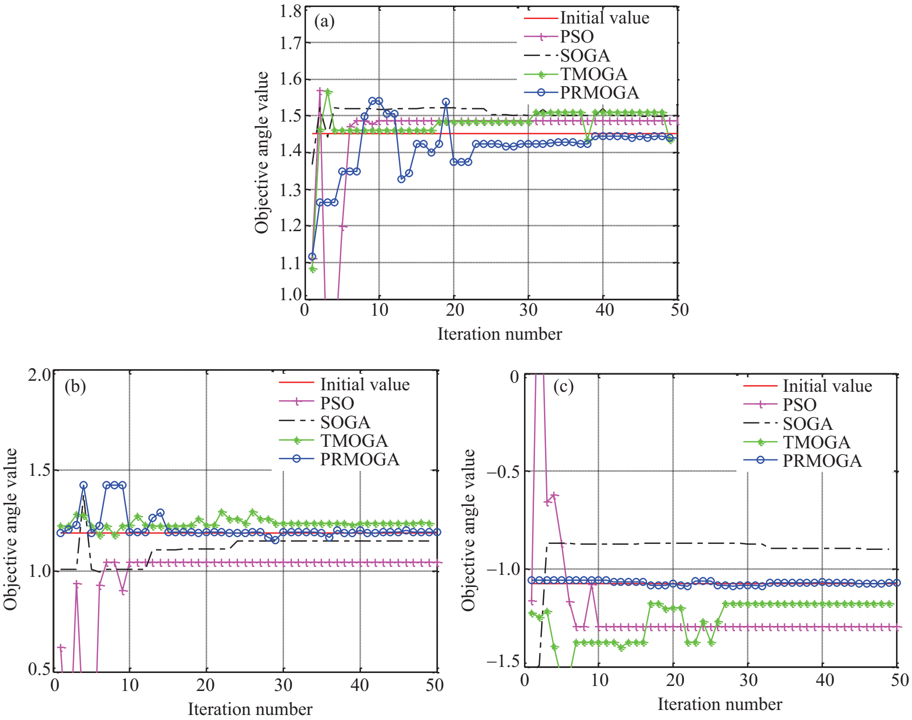

Figure 5 (Color online) Experimental comparison of objective angle values at (a) angle1, (b) angle2, and (c) angle3.

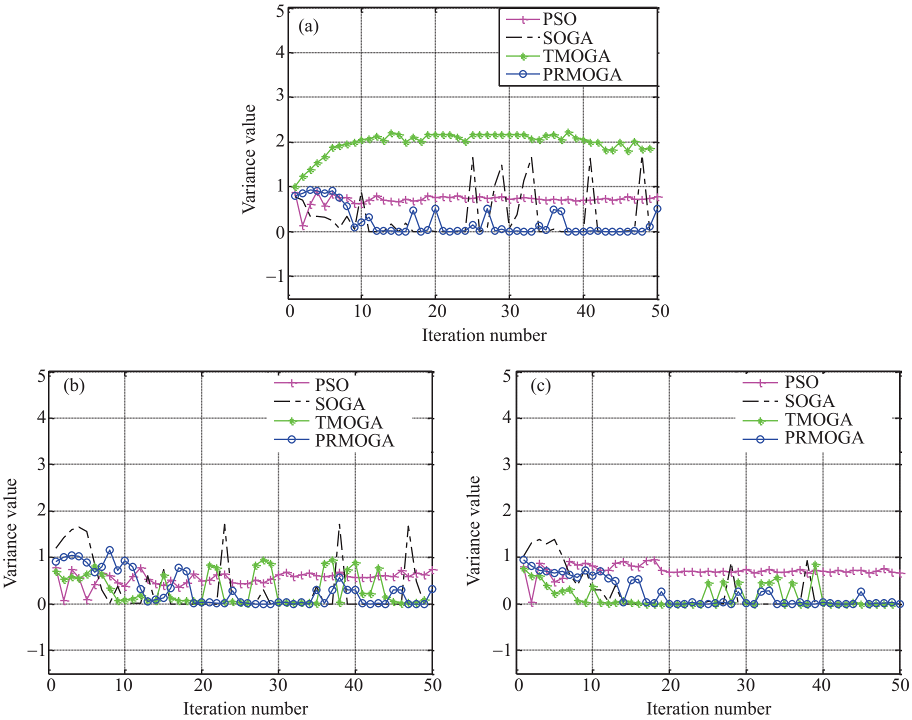

Figure 6 (Color online) Experimental comparison of the variance at (a) angle1, (b) angle2, and (c) angle3.

in the existing gesture adaptive researches, there are only a few studies that took into account tracking error, end error, and energy consumption. In contrast, our method is well coordinated by considering all the three aspects, which has advantages in the application.

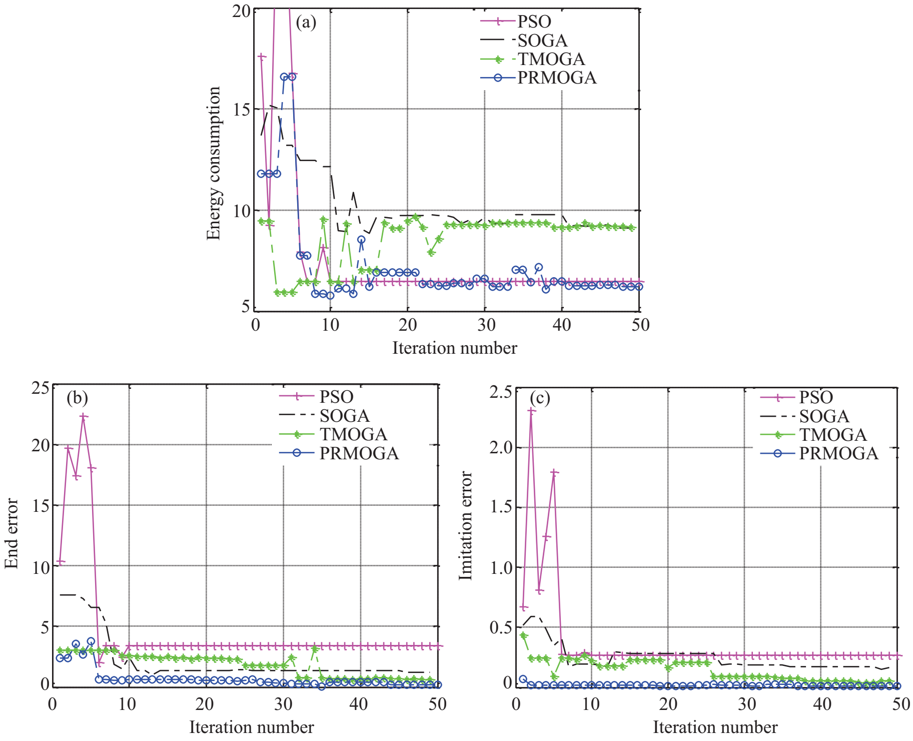

Figure 7 (Color online) Comparison of (a) energy consumption, (b) end errors, and (c) imitation errors.

Table 3 Comparison results calculated by four different algorithms

Algorithm	Energy	End error	Tracking error	Computation time (s)
PSO	0.9154	3.3924	0.2682	0.043449
SOGA	1.6277	1.9666	0.2933	0.066131
TMOGA	1.0336	−0.5312	0.1037	0.069721
PRMOGA	0.8311	0.1497	0.0550	0.073386

4 Conclusion

With the aim to make a robotic arm adapt to human arm gestures, the PRMOGA was proposed to optimize the movement of the three steering gears of a robotic arm. The simulation experiment showed that the proposed PRMOGA could improve the performance of the robotic arm to adapt to human arm gestures. Moreover, the results of the preliminary application experiment indicated that the robotic arm could adapt to the human arm's motion in real time.

The energy consumption (for saving energy), end error (ensuring touching), and posture tracking (for comfort) function are all important parameters in human-computer interaction [40]. However, in the related literature, there are only a few studies that took into account all the three aspects. By setting the objective function to include these three aspects, the proposed method solved not only the problem of time and error but also the problem of premature convergence of the traditional GA algorithms. The proposed PRMOGA can rapidly search the solution space of the three angles of the robotic arm's joints. By randomly shuffling each generation, the diversity of the population was ensured. In each generation, the elite strategy was adopted to reduce the energy consumption of a robotic arm. In robotic arm control, the tracking error function was used to accelerate the convergence of the objective function. Then, the output was transmitted to the lower machine through command conversion to control the simulated robotic arm and accurately simulate the human arm movement to reach the pre-defined end position. The experimental results showed that PRMOGA can effectively improve the accuracy of the robotic arm's attitude adaptation.

Although the proposed algorithm can solve the problems of energy consumption and errors in an HRI system, it still needs to be improved in terms of computing time, while the traditional algorithms have their own advantages in setting up the mathematical model having sufficient theory to support the

Table 4 Comparative analysis of the existing methods

Algorithm	Energy	End error	Tracking error	Computation time (s)
CHG-EP [38]	1.8231	0.4106	0.0932	0.076231
AGMOPSO [39]	0.9062	1.8741	0.1013	0.062146
PRMOGA	0.8311	0.1497	0.0550	0.073386

optimization problem [41]. Compared with the GA, the traditional algorithm still has many advantages in solving some problems, such as high computational speed and easy implementation for solving multi-objective problems. Therefore, in our future research, we will combine the advantages of the GAs and the traditional methods to further explore the robotic arm tracking problem. Further, we will extend the proposed method to robots for developing an emotional, social robot system [42], so that it is able to achieve smoother HRIs.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61973286, 61603356, 61733016, 61773353), 111 Project (Grant No. B17040), Hubei Provincial Natural Science Foundation (Grant No. 2015CFA010), Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) (Grant No. 2018039), and Wuhan Science and Technology Project (Grant No. 2017010201010133).

References

1 Zhang J W, Wang L, Xing L N. Large-scale medical examination scheduling technology based on intelligent optimization. J Comb Optim, 2019, 37: 385–404

2 Yue M, Ning Y G, Yu S Z, et al. Composite following control for wheeled inverted pendulum vehicles based on human-robot interaction. Sci China Inf Sci, 2019, 62: 050206

3 Du G L, Chen M X, Liu C B, et al. Online robot teaching with natural human-robot interaction. IEEE Trans Ind Electron, 2018, 65: 9571–9581

4 Hwang J, Kim J, Ahmadi A, et al. Dealing with large-scale spatio-temporal patterns in imitative interaction between a robot and a human by using the predictive coding framework. IEEE Trans Syst Man Cybern Syst, 2018, 50: 1918–1931

5 Chen L F, Zhou M T, Wu M, et al. Three-layer weighted fuzzy support vector regression for emotional intention understanding in human-robot interaction. IEEE Trans Fuzzy Syst, 2018, 26: 2524–2538

6 Chen L F, Wu M, Zhou M T, et al. Information-driven multirobot behavior adaptation to emotional intention in human-robot interaction. IEEE Trans Cogn Dev Syst, 2018, 10: 647–658

7 Jang Y, Jeon I, Kim T K, et al. Metaphoric hand gestures for orientation-aware VR object manipulation with an egocentric viewpoint. IEEE Trans Human-Mach Syst, 2017, 47: 113–127

8 Ding I J, Chang C W. An eigenspace-based method with a user adaptation scheme for human gesture recognition by using Kinect 3D data. Appl Math Model, 2015, 39: 5769–5777

9 Yang H, Li X, Wang Y, et al. Control of single-cell migration using a robot-aided stimulus-induced manipulation system. IEEE/ASME Trans Mechatron, 2017, 22: 815–825

10 Zhang Z, Beck A, Magnenat-Thalmann N. Human-like behavior generation based on head-arms model for robot tracking external targets and body parts. IEEE Trans Cybern, 2015, 45: 1390–1400

11 Lou Y H, Wu W J, Vatavu R D, et al. Personalized gesture interactions for cyber-physical smart-home environments. Sci China Inf Sci, 2017, 60: 072104

12 Kılıboz N, Güdükbay U. A hand gesture recognition technique for human—computer interaction. J Visual Commun Image Represent, 2015, 28: 97–104

13 Ding I J, Chang C W. An adaptive hidden Markov model-based gesture recognition approach using Kinect to simplify large-scale video data processing for humanoid robot imitation. Multimed Tools Appl, 2016, 75: 15537–15551

14 Obo T, Loo C K, Seera M, et al. Hybrid evolutionary neuro-fuzzy approach based on mutual adaptation for human gesture recognition. Appl Soft Comput, 2016, 42: 377–389

15 Huang C M, Chen Y R, Fu L C. Visual tracking of human head and arms using adaptive multiple importance sampling on a single camera in cluttered environments. IEEE Sens J, 2014, 14: 2267–2275

16 Emgin S E, Aghakhani A, Sezgin T M, et al. HapTable: an interactive tabletop providing online haptic feedback for touch gestures. IEEE Trans Visual Comput Graph, 2019, 25: 2749–2762

17 Jevtic A, Valle A F, Alenya G, et al. Personalized robot assistant for support in dressing. IEEE Trans Cogn Dev Syst, 2019, 11: 363–374

18 Li Y, Ge S S. Human-robot collaboration based on motion intention estimation. IEEE/ASME Trans Mechatron, 2014, 19: 1007–1014

19 Wakita K, Huang J, Di P, et al. Human-walking-intention-based motion control of an omnidirectional-type cane robot. IEEE/ASME Trans Mechatron, 2013, 18: 285–296

20 Chen L F, Wu M, Zhou M, et al. Dynamic emotion understanding in human-robot interaction based on two-layer fuzzy SVR-TS model. IEEE Trans Syst Man Cybern Syst, 2020, 50: 490–501

21 Chen L F, Su W J, Feng Y, et al. Two-layer fuzzy multiple random forest for speech emotion recognition in human-robot interaction. Inf Sci, 2020, 509: 150–163

22 Fang H, Shang C S, Chen J. An optimization-based shared control framework with applications in multi-robot systems. Sci China Inf Sci, 2018, 61: 014201

23 Chevtchenko S F, Vale R F, Macario V. Multi-objective optimization for hand posture recognition. Expert Syst Appl, 2018, 92: 170–181

24 Saremi S, Mirjalili S, Lewis A, et al. Enhanced multi-objective particle swarm optimisation for estimating hand postures. Knowledge-Based Syst, 2018, 158: 175–195

25 Wang J W, Wang H F, Ip W H, et al. Predatory search strategy based on swarm intelligence for continuous optimization problems. Math Problems Eng, 2013, 2013: 87–118

26 Yi J H, Xing L N, Wang G G, et al. Behavior of crossover operators in NSGA-III for large-scale optimization problems. Inf Sci, 2020, 509: 470–487

27 Tarokh M, Zhang X. Real-time motion tracking of robot manipulators using adaptive genetic algorithms. J Intell Robot Syst, 2014, 74: 697–708

28 Datta R, Pradhan S, Bhattacharya B. Analysis and design optimization of a robotic gripper using multiobjective genetic algorithm. IEEE Trans Syst Man Cybern Syst, 2016, 46: 16–26

29 Chen C H, Liu T K, Chou J H. A novel crowding genetic algorithm and its applications to manufacturing robots. IEEE Trans Ind Inf, 2014, 10: 1705–1716

30 Zacharia P T, Tsirkas S A, Kabouridis G, et al. Planning the construction process of a robotic arm using a genetic algorithm. Int J Adv Manuf Technol, 2015, 79: 1293–1302

31 Wang R, Lai S, Wu G, et al. Multi-clustering via evolutionary multi-objective optimization. Inf Sci, 2018, 450: 128–140

32 Xiang S, Xing L N, Wang L, et al. Comprehensive learning pigeon-inspired optimization with Tabu list. Sci China Inf Sci, 2019, 62: 070208

33 Liang X, Jiang Z, Wang B. Recognition algorithm and optimization experiments on tomato picking robots. Int J Signal Process Image Process Pattern Recogn, 2016, 9: 389–400

34 Mohamed Z, Kitani M, Kaneko S, et al. Humanoid robot arm performance optimization using multi objective evolutionary algorithm. Int J Control Autom Syst, 2014, 12: 870–877

35 Mukhopadhyay A, Maulik U, Bandyopadhyay S, et al. A survey of multiobjective evolutionary algorithms for data mining: part I. IEEE Trans Evol Computat, 2014, 18: 4–19

36 Mahmoodabadi M J, Safaie A A, Bagheri A, et al. A novel combination of Particle Swarm Optimization and Genetic Algorithm for Pareto optimal design of a five-degree of freedom vehicle vibration model. Appl Soft Comput, 2013, 13: 2577–2591

37 Chen D, Gao Z. A multi-objective generic algorithm approach for optimization of building energy performance. In: Computing in Civil Engineering. Reston: American Society of Civil Engineers, 2011. 51–58

38 Gupta H P, Chudgar H S, Mukherjee S, et al. A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. IEEE Sens J, 2016, 16: 6425–6432

39 Han H, Lu W, Zhang L, et al. Adaptive gradient multiobjective particle swarm optimization. IEEE Trans Cybern, 2018, 48: 3067–3079

40 Creaco E, Pezzinga G. Embedding linear programming in multi objective genetic algorithms for reducing the size of the search space with application to leakage minimization in water distribution networks. Environ Model Softw, 2015, 69: 308–318

41 Deb K. Multi-objective optimization. In: Search Methodologies. Boston: Springer, 2014. 403–449

42 Chen L, Zhou M, Su W, et al. Softmax regression based deep sparse autoencoder network for facial emotion recognition in human-robot interaction. Inf Sci, 2018, 428: 49–61