A Micro Dynamic Multi-objective Evolutionary Algorithm for Small-scale Smart Greenhouse with Low-power Microprocessor

Zhongtian Luo
Jiujiang University
Jiujiang, China
Luo_Zhongtian@aliyun.com

Jianpeng Xiong*
Jiujiang University
Jiujiang, China
XJP1972026766@aliyun.com

Hu Peng*
Jiujiang University
Jiujiang, China
hu_peng@whu.edu.cn

Gaosheng Zhan
Jiujiang University
Jiujiang, China
yuanm1017@aliyun.com

Qingfu Zhang
City University of Hong Kong
Hong Kong, China
qingfu.zhang@cityu.edu.hk

Hui Wang
Nanchang Institute of Technology
Nanchang, China
huiwang@whu.edu.cn

Xinyu Zhou
Jiangxi Normal University
Nanchang , China
xyzhou@jxnu.edu.cn

Wei Li
Jiangxi University of Science and Technology
Ganzhou, China
liwei@jxust.edu.cn

Ying Huang
Gannan Normal University
Ganzhou, China
nhwshy@whu.edu.cn

ABSTRACT

Smart greenhouse is a modern agricultural facility that integrates smart control systems to regulate the plant growth environment through advanced intelligent technology and devices. In recent years, smart greenhouses have received widespread attention and have been applied in agriculture. Due to their high energy demands and costs, current smart greenhouses are often impractical for applications with limited resources. Nevertheless, small-scale smart greenhouse with low-power microprocessor, are more suitable for homes, offices, and other fields. Therefore, this paper proposes a micro dynamic multi-objective evolutionary algorithm (**μDMOEA**) for small-scale smart greenhouse with low-power microprocessor, which applies chaotic mapping to select dynamic response strategies based on the fitness of dominant relationships and k-nearest neighbor environmental selection. μDMOEA performs well in the simulation of small-scale smart greenhouses. It not only outperforms SGEA, DNSGA-II, and RVCP in IGD indicator but also plays a good role in adjusting environmental parameters. It demonstrates the feasibility and effectiveness of micro dynamic multi-objective optimization on small-scale smart greenhouse with low-power microprocessor.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and cyber-physical systems**; • **Computing methodologies** → *Modeling and simulation*; • **Theory of computation** → Design and analysis of algorithms.

KEYWORDS

A micro dynamic multi-objective algorithm, Smart greenhouse, Low-power microprocessor

ACM Reference Format:
Zhongtian Luo, Jianpeng Xiong, Hu Peng, Gaosheng Zhan, Qingfu Zhang, Hui Wang, Xinyu Zhou, Wei Li, and Ying Huang. 2024. A Micro Dynamic Multi-objective Evolutionary Algorithm for Small-scale Smart Greenhouse with Low-power Microprocessor. In *Genetic and Evolutionary Computation Conference (GECCO '24), July 14–18, 2024, Melbourne, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3638530.3654276

1 INTRODUCTION

Smart greenhouses have received widespread attention and have been applied in agriculture [20]. However, current smart greenhouses are characterized by high power and cost, which makes them unsuitable for conditions with limited resources. Small-scale smart greenhouses [17] with low-power microprocessor are more suitable for home, office, and other fields. Low-power microprocessors are a type of electronic component that consume low energy but have good processing capabilities. In small-scale smart greenhouses, sensors measure the parameters of the environment, then low-power microprocessors process the data and give instructions to the actuators. Thus, the plant thrives through this process.

Real-world applications such as small-scale smart greenhouses can be modeled as multi-objective optimization problems (MOPs). Specifically, the type of MOPs [6, 19] with the objective function, constraints, and other environmental parameters change over time can be defined as dynamic multi-objective optimization problems (DMOPs). In DMOPs, multiple objectives are usually in conflict, and no single solution can optimize all objectives simultaneously. The goal of solving DMOPs is to find a time-varying Pareto-optimal set (PS) and Pareto-optimal front (PF) from the candidate set that covers

*Corresponding author

GECCO '24, July 14–18, 2024, Melbourne, Australia

ACM ISBN 979-8-4007-0495-6/24/07.
https://doi.org/10.1145/3638530.3654276

the various best trade-offs between the objectives. Currently, the most popular approach to solving DMOPs is to design a dynamic multi-objective evolutionary algorithm (DMOEA) [13]. However, normal-sized DMOEAs make it difficult to store relevant information and may interrupt program operation midway due to exceeding memory. In this respect, micro multi-objective evolutionary algorithms (μMOEAs) [14] can solve the problems of normal-sized multi-objective evolutionary algorithms (MOEAs) in a simple and computationally efficient way on low-power microprocessors [12]. For the above issues, this paper designed a micro dynamic multi-objective optimization evolutionary algorithm (μDMOEA). The work of this paper also involves applying the mathematical model of a small-scale smart greenhouse to simulation and using the μDMOEA for environmental optimization. The main contributions of this paper can be summarized as follows.

- By analyzing the characteristics of low-power microprocessors and combining the properties of DMOEAs and μMOEAs, this paper suggested the feasibility of applying micro dynamic multi-objective optimization to small-scale smart greenhouses.
- This paper proposes a micro dynamic multi-objective evolutionary algorithm (μDMOEA) based on chaotic mapping dynamic strategy selection.
- Through setting a small-scale smart greenhouse and using its mathematical model, μDMOEA is simulated on the small-scale smart greenhouses problem. The results show that μDMOEA has excellent performance in a small-scale smart greenhouse.

This paper is organized as follows. In Section 2, the preliminaries for the three core research directions of this paper are introduced. Section 3 provides a detailed description of μDMOEA. Section 4 shows the simulation results in a small-scale smart greenhouse. Section 5 is the summary of this paper and prospects for future work.

2 PRELIMINARIES

This section focuses on three key directions of the paper, and analyzes the characteristics of low-power microprocessors at the beginning. Subsequently, a detailed mathematical model for a small-scale smart greenhouse with low-power microprocessor is presented, which allows micro dynamic multi-objective evolutionary optimization for application in it.

2.1 Low-power Microprocessor

Low-power microprocessor is a type of micro computational processing unit designed to solve computing tasks that consume as little energy as possible. With the continuous advancement of electronic technology, low-power microprocessors with stronger performance, smaller size, and lower cost are constantly being produced [2]. Low-power microprocessors possess characteristics such as low power consumption and easy scalability, rendering them an ideal choice for IoT devices in various fields, including smart homes [10], smart agriculture [22] and so on. Meanwhile, their low price, approximately \$5, provides room for their application in a wider field. In order to achieve excellent low-power microprocessors, low-power processing technology and the selection of device feature sizes play an important role.

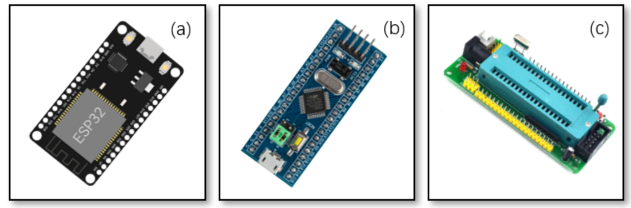


Figure 1: Low-power microprocessors: (a) ESP32, (b) STM32, (c) AT89C51

Nowadays, many low-power microprocessors with outstanding performance have been produced. As shown in Figure 1, it includes three common embedded processors: the ESP32, STM32, and AT89C51. Taking ESP32-WROOM [1] as an example, it provides very powerful performance at a price of \$3, including but not limited to WiFi and Bluetooth. Its core is ESP32-DOWDQ6, which has the characteristics of scalability and adaptability. Two CPU cores can be individually controlled or powered on. The clock frequency range is from 80MHz to 240MHz. Users can cut off the power to the CPU and use low-power coprocessors to continuously monitor the status changes of peripherals or whether certain analog quantities exceed the interval value. ESP32 also integrates a variety of peripherals, including capacitive touch sensors, Hall sensors, low-noise sensing amplifiers, SD card interfaces, Ethernet interfaces, high-speed SDIOSPI, UART, I2S, and I2C. Besides, low-power microprocessors can switch to deep sleep mode when in a non working state, which makes their power consumption very low. They can indicate that low-power microprocessors are capable of handling tasks at low power consumption.

2.2 Small-scale Smart Greenhouse

In recent years, the convergence of technology and agriculture has stimulated innovative applications aimed at optimizing crop cultivation and environmental sustainability. Among these advancements, small-scale smart greenhouses [17] have the potential to serve as focal points for algorithmic research and automation systems, enhancing agricultural productivity and resource efficiency.

The mathematical model for the dynamic multi-objective optimization problem of the small-scale smart greenhouse applied in this paper is referenced from [21, 24]. The following are the state equations for the three main environmental parameters of the model.

(1) *Humidity*

The humidity of the small-scale smart greenhouse is modeled by the following equations.

$$\dot{x}_{steam} = \frac{1}{3600 \cdot GH} \cdot (Trans + WaterInj - EnvExc - CondEvap) \quad (1)$$

where $Trans$ is the transpiration of the plants, $WaterInj$ is the water injection, $EnvExc$ is the exchange with environment through ventilation, $CondEvap$ is the condensation and evaporation on the greenhouse hull. Below are some related formulas:

$$Trans = 100 \cdot LeafSize[month] \cdot PM2 \cdot LeafTrans[month] \cdot TrGrow \quad (2)$$

$$TrGrow = (1 - b_0 \cdot (x_{CO_2} - 600)) \cdot \frac{TrCur}{TrStd} \quad (3)$$

$$\begin{aligned} TrCur = (&b_1 + b_2 \cdot x_{sun} + b_3 \cdot x_{sun}^2 \\ &+ b_4 \cdot f_{RH}(x_{steam}, x_{atempA})) \\ &\cdot f_{SD}(x_{steam}, x_{atempA}) \end{aligned} \quad (4)$$

$$TrStd = 10 \cdot (b_1 + 300 \cdot b_2 + 300^2 \cdot b3 + 60 \cdot b_4) \quad (5)$$

$$WaterInj = CW \cdot u_{water} \cdot (f_{SSP}(x_{atempA}) - f_{SP}(x_{steam}, x_{atempA})) \quad (6)$$

$$EnvExc = (u_{vent} + VM0 + VM1 \cdot v_{wind}) \cdot (x_{steam} - v_{steam}) \quad (7)$$

$$CondEvap = \begin{cases} Cond & if\ Cond > 0 \\ Cond & if\ Cond < 0\ and\ x_{cond} > 0 \\ 0 & if\ Cond < 0\ and\ x_{cond} = 0 \end{cases} \quad (8)$$

$$Cond = Trpo \cdot GR \cdot \frac{f_{SP}(x_{steam}, x_{atempA}) - f_{SSP}(x_{htempA})}{0.5 \cdot RWS \cdot (x_{atempA} + x_{htempA})} \quad (9)$$

$$Trpo = \frac{1.33 \cdot 3600 \cdot \left| x_{atemp} - x_{htemp} \right|^{0.33}}{DA \cdot HCA} \quad (10)$$

$$x_{htemp} = \begin{cases} -2.71 + 0.00811 \cdot v_{sun} + 0.795 \cdot x_{atemp} + 0.289 \cdot v_{atemp} & if\ 5 \leq month \leq 9 \\ \frac{1}{3} \cdot x_{atemp} + \frac{2}{3} \cdot v_{atemp} & Otherwise \end{cases} \quad (11)$$

(2) *Temperature*

The small-scale smart greenhouse temperature is modeled as follows.

$$\begin{aligned} \dot{x}_{atemp} = \frac{1}{HCap} \cdot (&u_{heat} + HSun - HExVent - HExGround \\ &- HExHull - HCondEvap - HHum) \end{aligned} \quad (12)$$

where $HCap$ is the heat capacity of the air and the plants, u_{heat} is the heating through the heating system, $HSun$ is the heating from the sun, $HExVent$ is the heat exchange with the environment through ventilation, $HExGround$ heat exchange through the ground, $HExHull$ is the heat exchange through the greenhouse hull, $HCondEvap$ is the heat change due to condensation on the greenhouse hull, $HHum$ is the heat change due to change in indoor humidity. The relevant formulas are shown below:

$$\begin{aligned} HCap = LeafSize[month] \cdot LSW \cdot HCW + GH \cdot HCA \cdot DA \\ + GH \cdot HCS \cdot x_{steam} \end{aligned} \quad (13)$$

$$HSun = TS \cdot x_{sun} \quad (14)$$

$$HExVent = \frac{1}{3600}(u_{vent} + VM0 + VM1 \cdot v_{wind}) \cdot (x_{energy} - v_{energy}) \quad (15)$$

$$x_{energy} = HCA \cdot DA \cdot x_{atemp} + x_{steam} \cdot (EEW0 + HCS \cdot x_{atemp}) \quad (16)$$

$$v_{energy} = HCA \cdot DA \cdot v_{atemp} + v_{steam} \cdot (EEW0 + HCS \cdot v_{atemp}) \quad (17)$$

$$HExGround = HG \cdot (x_{atempA} - v_{gtempA}) \quad (18)$$

$$HExHull = GR \cdot (HW0 + HW1 \cdot v_{wind}) \cdot (x_{atempA} - v_{atempA}) \quad (19)$$

$$HCondEvap = \frac{1}{3600} \cdot EEW \cdot CondEvap \qquad (CondEvap\ from\ Eq.8) \quad (20)$$

$$HHum = GH \cdot (EEW0 + HCS \cdot x_{atemp}) \cdot \dot{x}_{steam} \quad (21)$$

(3) *CO_2 density*

The CO2 density is modeled by the following equations.

$$\dot{x}_{CO_2} = \frac{u_{CO_2} - CPhoto - CExVent}{3600 \cdot 10^{-6} \cdot DC \cdot GH} \quad (22)$$

where u_{CO_2} is the artificially injected CO2, $CPhoto$ is the CO2 consumption by the plants through photo-synthesis and transpiration, $CExVent$ is the exchange with the environment through ventilation. The relevant formulas are as follows:

$$\begin{aligned} CPhoto = 100 \cdot LeafSize[month] \cdot PM2 \\ \cdot LeafCO2Ex[month] \cdot CPhGrow \end{aligned} \quad (23)$$

$$CPhGrow = \begin{cases} CPhCur \cdot CPhDec & if\ CPhCur > 0 \\ CPhCur & otherwise \end{cases} \quad (24)$$

$$\begin{aligned} CPhCur = c_1 \cdot (1 - exp(0.5 \cdot -c_2 \cdot x_{sun})) \\ \cdot (1 - exp(-c_3 \cdot x_{CO_2})) \\ \cdot (x_{atemp} + c_4 \cdot x_{atemp}^2) \\ - c_5 \cdot (x_{atemp} + c_6 \cdot x_{atemp}^2) \end{aligned} \quad (25)$$

$$CPhDec = \begin{cases} exp(-c_7 \cdot (d_1 - f_{SD}(x_{steam}, x_{atempA}))^2) & if\ f_{SD}(x_{steam}, x_{atempA}) < d_1 \\ 1 & if\ d_1 \leq f_{SD}(x_{steam}, x_{atempA}) \leq d_2 \\ exp(-c_8 \cdot (d_2 - f_{SD}(x_{steam}, x_{atempA}))^2) & if\ d_2 < f_{SD}(x_{steam}, x_{atempA}) \end{cases} \quad (26)$$

$$CExVent = 10^{-6} \cdot DC \cdot (u_{vent} + VM0 + VM1 \cdot v_{wind}) \cdot (x_{CO_2} - v_{CO_2}) \quad (27)$$

Detailed parameter settings can be found in [21]. The solution to the equations can be approximated by a fourth-order Runge-Kutta formula. Control variables are sampled at each half-hour interval, which are subsequently converted into seconds within the equations. This conversion is essential because the differential equations require approximation through small time slices to ensure both stability and accuracy.

Our work applies the mathematical model of the small-scale smart greenhouse mentioned above to simulation. Based on this real-world problem, this paper uses many hardware devices, such as low-power microprocessors, sensors, etc. The low-power microprocessors that have been burned with algorithms will optimize the greenhouse environment by analyzing environmental parameters and controlling the actuators. The entire process demonstrates the excellent performance of the micro dynamic multi-objective evolutionary algorithm in small-scale smart greenhouses with low-power microprocessor.

2.3 Micro Multi-objective Evolutionary Algorithm

Micro multi-objective evolutionary algorithms (μMOEAs) are efficient, as they consume low computational resources. μMOEAs search the solution space with a micro population to achieve rapid convergence. Consequently, the population prefers to converge prematurely, leading to a deficiency in diversity. Addressing this issue is a significant challenge for researchers. Over the years, although μMOEAs have not been a popular field, there has still been some related work. Coello and Pulido et al., pioneers in μMOEAs, proposed micro-GA [3] and μGA^2 [18], which provided many basic theoretical support. Thereafter, Tiwari et al. introduced many typical diversity strategies into μMOEAs to design AMGA [16] and AMGA2 [15]. They had effectively introduced many excellent technologies from typical MOEAs into μMOEAs. Peng et al. applied μMOEAs to the microgrid energy optimization problem and the proposed μMMABC [14] expands the application field. Moreover, Peng et al. first introduced decomposition ideas into μMOEAs and developed μMOEAs [12]. Meanwhile, they also took the lead in suggesting the application of μMOEAs to low-power microprocessors for resource optimization.

In fact, the environment of many real-world problems will change over time. Among these dynamic real-world problems, there are some problems that normal-sized DMOEAs cannot solve easily. For example, low-power microprocessors with limited computational resources may terminate programs due to insufficient memory during the process. In this case, there is a demand for algorithms [23] that consume low computational resources. Therefore, this paper proposes a micro dynamic multi-objective evolutionary algorithm (μDMOEA).

3 THE PROPOSED μDMOEA

This section contains three subsections closely related to μDMOEA. The first subsection is the main framework of the algorithm proposed in this paper. Subsequently, here introduces the chaotic mapping selection and nominal convergence, where the chaotic mapping selection allows the algorithm to choose different preferences for dynamic response strategies.

3.1 The Framework of μDMOEA

The main idea of μDMOEA is to select response strategies for dynamic changes within the environment through chaotic mapping [14], and optimize the population based on different approaches.

Algorithm 1 μDMOEA

Input: Population size N, ratio of change response solutions ζ
Output: Population Pop

Randomly initialize population Pop ;
while $FEs < MaxFEs$ **do**
  if Environmental change detected **then**
    $selected \leftarrow \{i_1, i_2, ..., i_{\lfloor \zeta * N \rfloor}\}$, i is randomly selected from $\{1, 2, ..., N\}$;
    Generate $Ch = \{ch_1, ch_2, ..., ch_{\lfloor \zeta * N \rfloor}\}$ by using $Eq.(28)$;
    if $mean(Ch) > \zeta$ **then**
      $Pop' \leftarrow Pop(selected)$ using nominal convergence operator ;
      $Pop(selected) = Pop'$;
    else
      $Pop'' \leftarrow Apply\ Eq.(29)\ to\ Pop$;
      $Pop(selected) \leftarrow EnvironmentalSelection(Pop'')$; // $Ref.\ to\ SPEA2$ [26]
    end if
  else
    $Off \leftarrow recombination(Pop)$;
    $Pop \leftarrow EnvironmentalSelection(Pop \cup Off)$;
  end if
end while

When comparing the value of chaotic numbers with the ratio of change response solutions, the approach enables the population to reach nominal convergence, or responds to changes based on Gaussian disturbance [11] and truncation on the population. The method of environmental selection is the diversity maintenance strategy used in SPEA2 [26].

Algorithm 1 presents the pseudocode for μDMOEA, where the inputs consist of the population size and the ratio of change response solutions. The first step is to randomly initialize and generate an initial population of size N. With the implementation of dynamic and static response mechanisms, the algorithm outputs an optimized population.

In the process of static optimization, the parent population undergoes recombination to generate offspring, and the combined population is subsequently subjected to environmental selection, which updates the population through truncation. The environmental selection strategy refers to SPEA2 [26], which calculates fitness based on dominance relationships and k-nearest neighbor selection.

$$x_{k+1} = x_k + \alpha x_k \,(1 - \alpha) \tag{28}$$

where x_k, x_{k+1} are the chaotic numbers, k is the index of individuals. α is a parameter that regulates chaotic mapping.

When environmental changes are detected, μDMOEA selects a ratio of change response solutions from the population and calculates their chaotic mapping values using equation (28). If the average value of the chaotic map is greater than the ratio of environmental response solutions, then the nominal convergence operator is used to update the response individuals. The concept of nominal convergence was proposed in [3], which is a method in μMOEAs that can effectively improve convergence.

$$\begin{cases} x_i' = x_i + N(\mu, \sigma^2), & if\ rand(0,1) < 1/D \\ x_i' = x_i, & otherwise \end{cases} \tag{29}$$

where x and x' are solutions before and after Gaussian disturbance, respectively. D is the number of decision variables.

If not, μDMOEA uses an environmental selection strategy that is familiar with the static optimization process. This involves selected change response solutions from the population after applying equation (29) (Gaussian disturbance [7, 11]), which helps compensate for population diversity. Repeat the above process until the termination conditions are met to terminate μDMOEA.

3.2 Chaos Mapping Selection

The chaotic mapping selection is derived from the environmental change response process in μDMOEA. It includes three steps, randomly selecting the indexes of individuals in the population, comparing the average chaotic numbers with the ratio of change response solutions, and executing the selected strategy.

The first step involves randomly selecting the indices of $\lfloor \zeta * N \rfloor$ individuals from a population of size N. Following that, an equal number of chaotic numbers are calculated to establish a mapping relationship with the selected individuals one by one. By comparing the average value of the obtained chaotic numbers with the predefined ratio of change response solutions, the chaotic mapping selection picks two different approaches. When the average value leans towards the nominal convergence operator, it is executed on the selected individuals in the population to enhance the convergence of the algorithm. On the contrary, when the result leans towards another approach, Gaussian disturbance [7, 11] is applied to the population to generate a new population. The new population is truncated until only $\lfloor \zeta * N \rfloor$ individuals with good diversity remain. Based on the indexes of the previously selected individuals, the next step is to update them accordingly to improve the diversity of the population.

Adjusting responses to environmental changes through chaotic numbers is attributed to their ergodicity and non-reproducibility. These properties make them more advantageous than fixed values, as they suggest a global search at a higher speed compared to random search [14]. Therefore, the chaotic mapping selection effectively improves the comprehensive performance of the μDMOEA when facing dynamic environments.

3.3 Nominal Convergence

Nominal convergence [3] is a concept from μMOEAs, which is defined as a lower number of evolutionary generations (typically 2 to 5). Goldberg pointed out that regardless of the length of the chromosome, a population size of 3 is sufficient to converge [5]. Meanwhile, Goldberg suggested applying genetic operators to a randomly generated population until it reaches nominal convergence and copying the optimal individuals from the converged population to another population, then the remaining individuals will be generated randomly. So far, the nominal convergence has been applied to some algorithms like micro-GA [3] and μGA^2 [18]. In this work, nominal convergence is applied to a genetic operator, which generates new solutions for updating response solutions. In the dynamic optimization process, when the chaotic mapping selection tends towards this approach, the operator is used to converge the population.

4 SIMULATION

This section introduces the settings of the simulation and explains the results. The simulation setup mainly focuses on the introduction of hardware and experimental environment. The second subsection provides an analysis of the Inverted Generational Distance metric results for μDMOEA in comparison with other algorithms, as well as discussing its regulatory influence on environmental parameters.

4.1 Simulation Setting

(1) *Hardware Devices*

a) Control system: microprocessor unit (MPU), sensors, actuators, cloud platform, and more details can be found in Figure 2 and Figure 3.
b) Devices:
MPU: STM32F103C8T6.
Sensors: DHT11 (air temperature and humidity), SGP30 (CO2 density), YL-69 (soil humidity), DS18B20 (soil temperature), and BH1750 (illuminance).
Actuators: fan, pump, humidifier, and lamp.
Cloud platform: Alibaba Cloud.

Table 1: The detailed upper and lower bounds and thresholds of environmental parameters. It is acceptable for the environmental parameter values to be within the threshold range.

Environmental parameter	threshold	upper and lower bound
H_{soil} (%rh)	>30	[0, 100]
T_{soil} (℃)	<18	[0, 30]
H_{air} (%rh)	>30	[0, 100]
T_{air} (℃)	<21.6	[0, 36]
CO_2 (ppm)	<800	[400, 1200]
I (lx)	>100	[0, 1000]

(2) *Environment Setting*

a) Population size: The population size in comparison algorithms and real-world problems is set to 20.
b) Maximum number of function evaluations ($MaxFE$): The value of $MaxFE$ in the simulation is set to 10000.
c) Problem: the real-world problem SSG (small-scale smart greenhouse problem) in Section 2.2.
d) Comparison algorithms: SGEA [8], DNSGA-II [4], and RVCP [25]. The population size of the algorithms is set to 20, the maximum number of function evaluations is 10000, and other parameters are consistent with the original paper.
e) Crossover and Mutation: The operator used in the simulation is the genetic operator, which combines simulated binary crossover (SBX) and polynomial mutation (PM). The crossover rate of SBX is 1, and the mutation rate of PM is $1/D$, where D is the number of decision variables. The distribution indexes of them are both set to $\eta_c = 20, \eta_m = 20$.

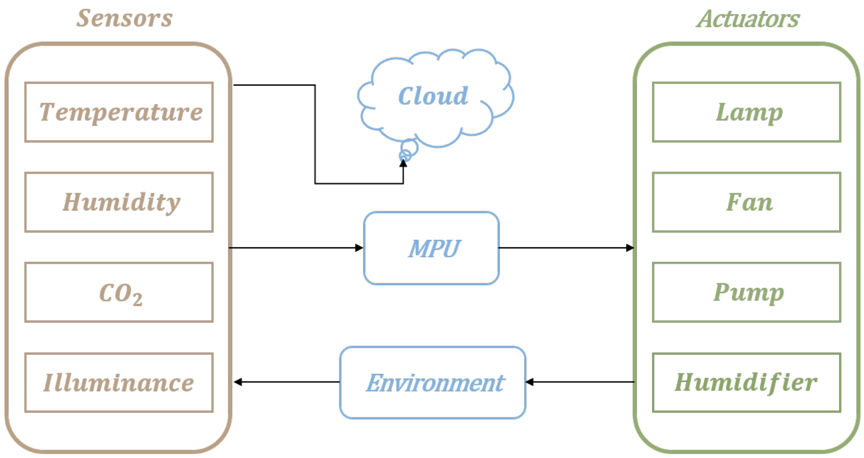


Figure 2: The control structure of small-scale smart greenhouse. The sensors measure environmental parameters and send them to the MPU for analysis and processing, which provides feedback to the actuators to adjust the environment. The above process data is transmitted to the cloud platform.

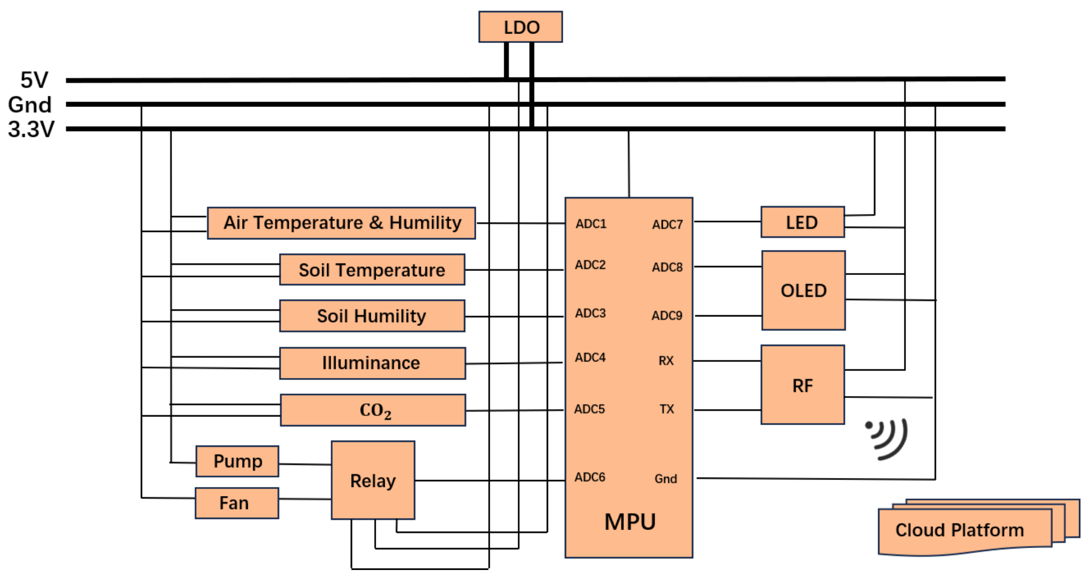


Figure 3: Schematic diagram of the small-scale smart greenhouse. It includes the wiring distribution of components such as MPU, sensors (the temperature and humidity of soil and air, illuminance, CO2 density), and actuators (fan, LED, pump, etc.) in small-scale smart greenhouses.

f) Upper / lower bounds and thresholds: The upper / lower bounds and threshold range of environmental parameters are shown in Table 1.

(3) *Performance Metrics*

The inverse generation distance (IGD) [9] is a frequently used comprehensive performance metric in evolutionary multi-objective optimization (EMO), which can simultaneously evaluate convergence and diversity. The principle can be described as the average Euclidean distance between the true Pareto front (PF) and the solutions output by the algorithm., with smaller value indicating better algorithm performance, in the following form:

$$IGD(P, PF_{true}) = \frac{\sum_{p^* \in PF_{true}} min\ dist(p_i, p^*)}{|PF_{true}|} \tag{30}$$

where P is the output of the algorithm, and PF_{true} is the sampling points on the true PF. $dist(p_i, p^*)$ calculates the Euclidean distance from the $p_i \in P$ to $p^* \in PF_{true}$.

Table 2: Mean and standard deviation IGD of comparison with SGAE, DNSGA-II, RVCP on real-world problem SSG (small-scale smart greenhouse problem).

Problem	M	D	SGEA	DNSGA-II	RVCP	μDMOEA
SSG	2	6	1.0673e-1 (9.99e-3) -	1.2133e-1 (1.34e-2) -	1.0615e-1 (1.23e-2) -	**9.5554e-2 (7.77e-3)**
+/-/=			0/1/0	0/1/0	0/1/0	—
Rank			3.00	4.00	2.00	**1.00**

4.2 Simulation Results and Analysis

This paper focuses on the simulation of the small-scale smart greenhouse. As shown in Figure 2, the simulation structure mainly includes electronic components (microprocessors, sensors, actuators, etc.), the greenhouse environment, and the cloud platform. Sensors measure environmental parameters such as humidity and temperature, and send data to the MPU. After receiving the data from sensors, the MPU will perform computational analysis and process the response to find a suitable solution on the PF for feedback. The feedback will be sent to the actuators, which will regulate the environment to enable the plants in the greenhouse to thrive. As shown in Figure 4, the data recorded during the above process will be uploaded to the IoT platform based on Alibaba Cloud. The platform can record simulation data in real-time, providing users with more convenient data recording and more efficient monitoring feedback.

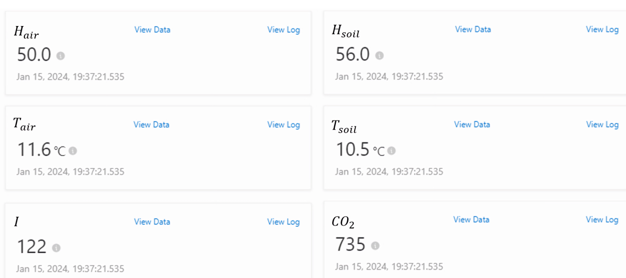


Figure 4: Environmental parameter values on Alibaba Cloud, allowing real-time display of environmental parameter values, including soil temperature, soil humidity, air temperature, air humidity, illuminance, and CO2 density.

Figure 3 shows a schematic diagram of the small-scale smart greenhouse with the microprocessor used in the simulation of this paper. Some of the sensors and actuators used in this paper are shown in Figure 5, and the selected MPU is STM32. The external power source supplies a 5V input to the system, which is then regulated to 3.3V through an LDO for stabilization. Both the water pump and humidifier necessitate a 5V power supply. The μDMOEA burned to the MPU controls each module. Each sensor module collects data for display on an OLED screen and wirelessly transmits it to the IoT platform, while also visualizing the data in the cloud.

In order to verify the effectiveness of μDMOEA in small-scale smart greenhouses with microprocessor, three dynamic multi-objective evolutionary algorithms are compared, including typical dynamic multiobjective evolutionary algorithms SGEA, DNSGA-II, and RVCP. The experimental results are shown in Table 2. The IGD value of

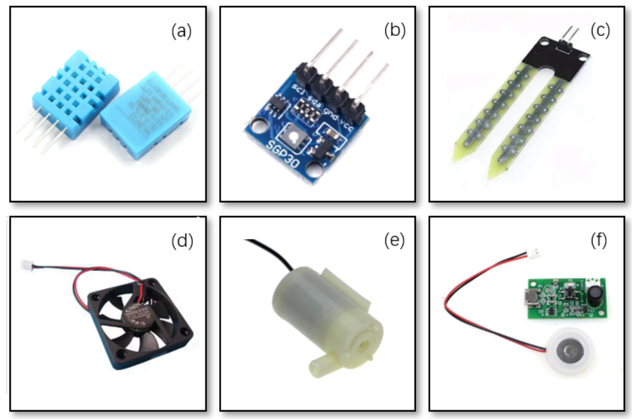


Figure 5: Sensors: (a) DHT11, (b) SGP30, (c) YL-69; Actuators: (d) fan, (e) pump, (f) humidifier.

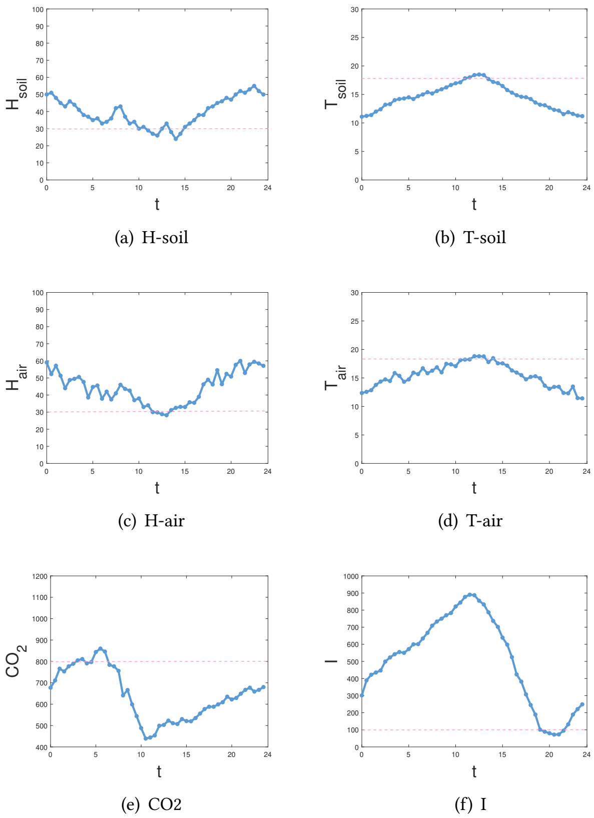
μDMOEA is smaller than the other three algorithms and ranks higher, indicating that μDMOEA performs better in the small-scale smart greenhouse.

Furthermore, Figure 6 presents the continuous regulatory performance of the μDMOEA over 24-hour in the small-scale smart greenhouse with microprocessor. This paper sets thresholds for different environmental parameters, and when the environmental parameter values are above/below the threshold, it indicates that they exceed the values required for a normal growth environment. The upper and lower bounds and thresholds of environmental parameters are detailed in Table 1. When the small-scale smart greenhouse environment changes, μDMOEA processes the data received from sensors and provides feedback to the actuators. This dynamic process ensures that the plants within the small-scale smart greenhouse can maintain healthy growth.

The above simulation results not only demonstrate the excellent performance of μDMOEA on small-scale smart greenhouse, but also demonstrate the feasibility of micro dynamic multi-objective evolutionary optimization in it.

5 CONCLUSION

This paper has suggested a micro dynamic multi-objective evolutionary algorithm, called μDMOEA. In the dynamic optimization process, chaotic mapping is used to compare with the ratio of change response solutions, and nominal convergence or environmental selection after Gaussian disturbance is chosen for dynamic change response. The environment selection of the algorithm refers



(a) H-soil (b) T-soil

(c) H-air (d) T-air

(e) CO2 (f) I

Figure 6: Trend curves of environmental parameters including soil humidity, soil temperature, air humidity, air temperature, CO2 density, and illuminance within 24 hours (red line = threshold).

to fitness based on dominance relationships and k-nearest neighbor selection.

In the simulation, the control structure and circuit distribution of the small-scale smart greenhouse set up in this paper are introduced there. To assess the performance of μDMOEA, it was compared with three DMOEAs. Additionally, its processing capability was tested by implementing it on an MPU. Not only does μDMOEA outperform the comparison algorithms, but the MPU embedded with μDMOEA also effectively controls the environment by processing information on environmental parameters. It demonstrated the feasibility of applying micro dynamic multi-objective evolutionary algorithms in a small-scale smart greenhouses with microprocessor.

From the research, it can be seen that there is potential for micro dynamic multi-objective evolutionary algorithms in a small-scale smart greenhouse with microprocessor. As for the future, we will continue to develop micro dynamic multi-objective evolutionary algorithms and apply them to small-scale smart greenhouse with microprocessor.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (62266024, 62366022), Jiangxi Provincial Natural Science Foundation (20232BAB202048).

REFERENCES

[1] Isa Anshori, Ghani Faliq Mufiddin, Iqbal Fawwaz Ramadhan, Eduardus Ariasena, Suksmandhira Harimurti, Henke Yunkins, and Cepi Kurniawan. 2022. Design of smartphone-controlled low-cost potentiostat for cyclic voltammetry analysis based on ESP32 microcontroller. *Sensing and Bio-Sensing Research* 36 (2022), 100490.

[2] W. Aspray. 1997. The Intel 4004 microprocessor: what constituted invention? *IEEE Annals of the History of Computing* 19, 3 (1997), 4–15.

[3] Carlos A. Coello Coello Coello and Gregorio Toscano Pulido. 2001. A Micro-Genetic Algorithm for Multiobjective Optimization. In *Evolutionary Multi-Criterion Optimization*, Vol. 1993. Springer, 126–140.

[4] Kalyanmoy Deb, Udaya N, and Karthik Sindhya. 2007. Dynamic Multi-objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-thermal Power Scheduling. In *Evolutionary Multi-Criterion Optimization*. Springer, 803–817.

[5] David E. Goldberg. 1989. Sizing Populations for Serial and Parallel Genetic Algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*. 70–79.

[6] Cheng Gong, Yang Nan, Lie Meng Pang, Qingfu Zhang, and Hisao Ishibuchi. 2023. Effects of Including Optimal Solutions into Initial Population on Evolutionary Multiobjective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '23)*. ACM, 661–669.

[7] Zhihui He, Hu Peng, Changshou Deng, Yucheng Tan, Zhijian Wu, and Shuangke Wu. 2019. A Spark-based Gaussian Bare-bones Cuckoo Search with dynamic parameter selection. In *2019 IEEE Congress on Evolutionary Computation (CEC)*. 1220–1227.

[8] Shouyong Jiang and Shengxiang Yang. 2017. A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 21, 1 (2017), 65–82.

[9] Miqing Li, Shengxiang Yang, Ke Li, and Xiaohui Liu. 2014. Evolutionary algorithms with segment-based search for multiobjective optimization problems. *IEEE Transactions on Cybernetics* 44, 8 (2014), 1295–1313.

[10] Vikas Kumar Maurya and Satyasai Jagannath Nanda. 2023. Time-varying multi-objective smart home appliances scheduling using fuzzy adaptive dynamic SPEA2 algorithm. *Engineering Applications of Artificial Intelligence* 121 (2023), 105944.

[11] Hu Peng, Changshou Deng, Hui Wang, Wenjun Wang, Xinyu Zhou, and Zhijian Wu. 2018. Gaussian bare-bones cuckoo search algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '18)*. ACM, 93–94.

[12] Hu Peng, Fanrong Kong, and Qingfu Zhang. 2023. Micro Multiobjective Evolutionary Algorithm With Piecewise Strategy for Embedded-Processor-Based Industrial Optimization. *IEEE Transactions on Cybernetics* (2023), 1–12. https://doi.org/10.1109/TCYB.2023.3336369

[13] Hu Peng, Changrong Mei, Sixiang Zhang, Zhongtian Luo, Qingfu Zhang, and Zhijian Wu. 2023. Multi-strategy dynamic multi-objective evolutionary algorithm with hybrid environmental change responses. *Swarm and Evolutionary Computation* 82 (2023), 101356.

[14] Hu Peng, Cong Wang, Yupeng Han, Wenhui Xiao, Xinyu Zhou, and Zhijian Wu. 2022. Micro multi-strategy multi-objective artificial bee colony algorithm for microgrid energy optimization. *Future Generation Computer Systems* 131 (2022), 59–74.

[15] Santosh Tiwari, Georges Fadel, and Kalyanmoy Deb. 2011. AMGA2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization. *Engineering Optimization* 43, 4 (2011), 377–401.

[16] Santosh Tiwari, Patrick Koch, Georges Fadel, and Kalyanmoy Deb. 2008. AMGA: an archive-based micro genetic algorithm for multi-objective optimization. In *Conference on Genetic Evolutionary Computation*. 729–736.

[17] Fotios Tolis, Taxiarchis-Foivos Blounas, and Dimitrios Tsipianitis. 2023. Implementation of a small scale smart greenhouse structure using Fuzzy Logic and IOT. In *2023 14th International Conference on Information, Intelligence, Systems Applications (IISA)*. 1–8. https://doi.org/10.1109/IISA59645.2023.10345847

[18] Gregorio Toscano Pulido and Carlos A. Coello Coello. 2003. The Micro Genetic Algorithm 2: towards Online Adaptation in Evolutionary Multiobjective Optimization. In *Evolutionary Multi-Criterion Optimization*. 252–266.

[19] Hai-Long Tran, Long Doan, Ngoc Hoang Luong, and Huynh Thi Thanh Binh. 2023. A Two-Stage Multi-Objective Evolutionary Reinforcement Learning Framework for Continuous Robot Control. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '23)*. ACM, 577–585.

[20] Pradyumna K. Tripathy, Ajaya K. Tripathy, Aditi Agarwal, and Saraju P. Mohanty. 2021. MyGreen: An IoT-Enabled Smart Greenhouse for Sustainable Agriculture.

IEEE Consumer Electronics Magazine 10, 4 (2021), 57–62.
[21] RasmusK. Ursem, Thiemo Krink, and Bogdan Filipič. 2002. A Numerical Simulator for a Crop-Producing Greenhouse. *EVALife Technical Report* (2002), 11.
[22] Xing Yang, Lei Shu, Jianing Chen, Mohamed Amine Ferrag, Jun Wu, Edmond Nurellari, and Kai Huang. 2021. A Survey on Smart Agriculture: Development Modes, Technologies, and Security and Privacy Challenges. *IEEE/CAA Journal of Automatica Sinica* 8, 2 (2021), 273–302.
[23] Ruizhen Yu. 2013. *An efficient dynamic micro multi-objective optimization method and its applications on PID control.* Master's thesis. Hunan University.
[24] Zhuhong Zhang. 2008. Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Applied Soft Computing* 8, 2 (2008), 959–971.
[25] Jinhua Zheng, Qishuang Wu, Juan Zou, Shengxiang Yang, and Yaru Hu. 2023. A dynamic multi-objective evolutionary algorithm using adaptive reference vector and linear prediction. *Swarm and Evolutionary Computation* 78 (2023), 101281.
[26] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001. SPEA2: Improving the strength pareto evolutionary algorithm. *TIK-Report* 103 (7 2001), 95–100.