IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 October 2025, accepted 2 December 2025, date of publication 4 December 2025,
date of current version 10 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3640360

APPLIED RESEARCH

Optimizing Pesticide Matching: A Comprehensive Multi-Objective Framework

YONGSHENG WANG[1], CONG JIN[2], YANJIE ZHOU[3], (Member, IEEE), WEIHUA FENG[1], GUOHAO ZONG[1], DI WANG[1], AND RUI WANG[1]
[1]China Zhengzhou Tobacco Research Institute, China National Tobacco Corporation, Zhengzhou, Henan 450001, China
[2]Ocean College, Zhejiang University, Zhoushan 316021, China
[3]School of Management, Zhengzhou University, Zhengzhou 450001, China

Corresponding author: Yanjie Zhou (ieyjzhou@zzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 72201252; in part by Ningbo Science and Technology Bureau (Ningbo Commonweal Program), under Project ID 2024S057; and in part by the Key Science and Technology Program of Henan Province, China, under Grant 232102210073.

ABSTRACT Pesticides are indispensable in agricultural production, as they not only effectively prevent pests and diseases, safeguarding crops from damage, but also regulate plant growth, thereby boosting crop yield and quality. Nevertheless, the improper use of pesticides and the persistence of residues pose severe threats to ecological health and environmental sustainability. This study constructs a multi-objective optimization model to mitigate pesticide reliance and promote precision application. This model considers three key factors: pesticide cost, therapeutic efficacy, and spraying frequency. We introduce a multi-objective optimization framework to outline the problem-solving approach systematically. Additionally, through a series of performance metrics, we comprehensively analyze and compare different algorithms and constraint handling techniques. The experimental results validate the rationality and effectiveness of the developed mathematical model. It offers practical and valuable guidance for optimizing pesticide spraying operations, helping to balance agricultural productivity, and environmental protection.

INDEX TERMS Multi-objective optimization, agricultural pesticides matching problem, mathematical model.

I. INTRODUCTION

With the continuous development of global agricultural production, pest control has become one of the core issues for agricultural production. The rational use of pesticides is not only related to the growth effect of crops but also closely related to environmental protection and human health. Pesticides have been widely used in agriculture to destroy or regulate pests for a long history [1]. However, pesticides are a double-edged sword. Pesticides protect food production and cause adverse effects on the environment. Governments have released legislation to reduce the usage of pesticides. Fewer chemical pesticides mean fewer environmental pollutants. To encourage farmers to use pesticides scientifically, recently, the European Commission published a toolbox of good practices [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Arman Oshnoei.

Many previous studies have studied spraying pesticides from different perspectives, such as how weather conditions affect pesticide leaching [3], pesticide contamination [4]. With the development of artificial intelligence, more and more knowledge graphs of agricultural pests and diseases are being developed, which can be used to extract the relationship between pesticides and diseases. According to the study of [5], more than half of the farmers did not scientifically use pesticides. Even though artificial intelligence could help farmers know more about the pesticide it could not help farmers decide on a pesticide to minimize total cost, maximize the therapeutic effect, and reduce spraying operations simultaneously.

Pesticide matching, a critical component of integrated pest management (IPM), refers to the systematic selection and application of appropriate pesticides based on a comprehensive analysis of crop types, pest and disease characteristics, pesticide mechanisms of action, and environmental factors.

This process aims to optimize pest control efficacy while minimizing adverse environmental impacts, aligning with agricultural sustainability principles. In modern agriculture, where precision and efficiency are paramount, pesticide matching has emerged as a pivotal research area for achieving sustainable crop protection. This paper investigates the pesticide matching problem under a practical agricultural scenario where a farmer manages multiple plots of land cultivating diverse crop types, each susceptible to varying pest and disease pressures. The farmer's objective is to procure an optimal set of pesticides and determine an application strategy that simultaneously minimizes total cost, maximizes therapeutic efficacy, and reduces the total number of spraying operations. The contributions of this paper are summarized as follows.

- This paper proposes a comprehensive framework to address the agricultural pesticide matching problem, considering multiple objectives. A novel encoding and decoding methodology is developed to facilitate the formulation and solution of the problem. This methodology, along with the proposed repair operations, enhances the flexibility and efficiency of the model. The repair operations play a crucial role in correcting any inconsistencies or errors in the model, ensuring that the solution process can be easily adapted and applied in real-world scenarios, contributing to more effective pesticide management strategies.
- A series of comprehensive experiments were conducted to validate the proposed mathematical models and the framework presented in this study. Additionally, a detailed case study was carried out to demonstrate the practical applicability of the proposed framework, providing insights into its functionality and effectiveness in real-world scenarios. The results from both the experimental analyses and the case study further confirm the viability and robustness of the approach.
- The findings presented in this paper offer valuable guidance for farmers in optimizing pesticide usage within the agricultural sector. The proposed framework provides a comprehensive approach to improving overall agricultural practices by minimizing pesticide and labor costs while enhancing the therapeutic efficacy of the pesticides. This contributes to cost reduction and promotes more sustainable and efficient pesticide management strategies, ultimately benefiting agricultural productivity and environmental sustainability.
- This paper introduces a novel research direction with significant potential for academic inquiry and practical applications within the agricultural industry. The study opens new avenues for further exploration, innovation, and interdisciplinary collaboration by addressing a previously underexplored aspect of the pesticide matching problem. The insights generated can guide future research efforts while providing actionable strategies for industry professionals seeking to optimize agricultural practices and sustainability.

The remainder of this paper is organized as follows. Section II shows the literature review. Section III introduces the mathematical models. Section IV presents the proposed solution approach. Section V presents the experimental results. A case study is shown in section VI. Finally, section VII gives the conclusions.

II. LITERATURE REVIEW

The matching problem, as a classic problem in optimization theory, is widely used in many fields, such as economics, computer science, operational research, sociology, etc., and shows substantial theoretical guiding value and practical application potential in different scenarios. In agriculture, pesticide matching serves as a core component of precision agricultural practices, playing a pivotal role in enhancing crop yields while mitigating the environmental footprint of pesticide use. However, compared to matching problems in other domains, research on pesticide matching remains in its nascent stages, with a relatively limited body of relevant literature.

Current agricultural research primarily focuses on five key areas: precision agricultural technology, crop growth optimization, soil and water resource management, agricultural environmental protection, and integrated pest control. Reference [6] presents a multi-objective mathematical model for optimizing biological plague control in soy farming, considering both the cost of control measures and the damage caused by the plague. Reference [7] discussed the application of emerging technologies in degrading pesticide residues in different foods. Reference [8] designed a kind of microcapsule with high insecticidal activity and biological safety, which was used to release pesticides intelligently, thus reducing the pollution of pesticides to the environment and promoting the development of green agriculture. Reference [9] expounded the influence and benefits of agrochemicals and looked forward to the prospect of sustainable agriculture. Reference [10] focused on the impact of the policy of replacing pesticides with biological substances.

Reference [11]introduced mathematical models of multi-objective optimization and the concept of Pareto solution set, and evaluates various transformation methods by leveraging simple example problems. Reference [12] introduced an innovative adaptive weighted sum method tailored for tackling multiobjective optimization problems. This method has been shown to generate a well-distributed Pareto front mesh, thereby facilitating effective visualization. Moreover, it can identify solutions in non-convex regions. Reference [13] proposed a multi-objective clustering approach and a hybrid optimization technique called Election-based Aquila Optimizer, which integrates Aquila Optimizer and Election-Based Optimization Algorithm to optimize Cluster Head selection, effectively addressing key challenges in precision agriculture. Reference [14] primarily aimed to evaluate energy usage and environmental indicators, optimize these factors using multi-objective genetic algorithms combined with data envelopment analysis, and identify potential energy-saving opportunities in mushroom production operations. Reference [15] addressed the multi-picking-robot task allocation problem by proposing a novel multi-objective discrete artificial bee colony algorithm. Extensive experimental evaluations in an intelligent orchard setting demonstrate the algorithm's robust performance and effectiveness across various task scales and robot configurations. Reference [16] proposed a mixed-integer quadratically constrained programming model for cropland layout optimization, incorporating multi-objective considerations including economic performance, biodiversity conservation, greenhouse gas emission reduction, and water quality management. Reference [17] explored the identification of sustainable locations for urban farming through a framework that optimally reconciles economic and environmental objectives. The problem is formulated as a multi-objective linear programming model, designed to maximize ecological benefits and crop production yield while minimizing transportation costs, sensor deployment expenses, and CO2 emissions. The proposed mathematical model is solved using a two-phase optimization approach.

Reference [18] proposed an innovative data evaluation method that leverages Mahalanobis distance and entropy to tackle the challenge of insufficient labeled data in intelligent pest identification, enabling effective pest identification with limited datasets and improving accuracy and efficiency in data-scarce scenarios. Reference [19] proposed an AI-based system that integrates real-time IoT data and advanced analytics for the detection, prevention, and control of pests, to reduce reliance on pesticides and mitigate environmental harm. Based on artificial intelligence, [20] developed the PlanteSaine pest management tools, which are capable of real-time pest and disease identification, thereby promoting the sustainable advancement of agriculture. Reference [21] provided a comprehensive overview of the contributions of artificial intelligence to pest management and its future applications. The judicious application of AI enables precise pest identification, early detection, and preventive measures, thereby minimizing unnecessary economic losses.

Reference [22] researched and developed an airborne computer vision component tailored for unmanned aerial vehicles (UAVs). Based on the study of [22], crop monitoring and spraying can be synchronized so that intelligent and accurate spraying can be carried out. Similarly, [23] captured image data through a UAV-borne imaging system, facilitating the accurate monitoring of plant diseases and contributing to optimizing disease control strategies. In terms of economic performance, [24] conducted an economic assessment of UAV utilization by analyzing key factors such as revenue, pesticide costs, pesticide application time, and spraying frequency, using propensity score matching for analysis. The findings indicated that, in terms of marginal income and marginal application time, the optimal area for pesticide spraying using UAVs was identified as 20 hectares. Focusing specifically on, [25] investigated and evaluated the UAV-based spraying system within cotton fields, employing imaging technology in conjunction with the Grey Wolf Optimizer-Artificial Neural Network approach to enhance the flexibility and efficacy of the UAV spraying system. This integration aims to optimize the operational performance of UAVs in precision agriculture, offering improvements in both system adaptability and spraying effectiveness. Beyond agricultural crops, [26] analyzed the temporal variation in the spectral characteristics of healthy and infected trees using UAV-borne hyperspectral imaging. The findings suggest a significant potential for the early detection of forest pests, highlighting the utility of hyperspectral remote sensing in monitoring forest health and enhancing pest management strategies.

TABLE 1. Notations and decision variables.

Set	
N	Set of diseases (pests)
M	Set of pesticides
Index	
i	Index of diseases
j	Index of pesticides
m	Index of pesticides
Parameters	
h_i	Area of land suffering from disease i
p_j	The price of pesticide j per unit (bottle)
g_{ij}	For disease i, the area that each unit of pesticide j can treat
w_{ij}	Therapeutic effect of pesticide j on disease i
e_{ij}	$e_{ij} = 1$, if the pesticide j can treat disease i; $e_{ij} = 0$, otherwise
a_{ij}	Upper bound of y_{ij}
k_{jm}	$k_{jm} = 1$, if the pesticide j_1 and the pesticide j_2 can't be mixed; $k_{jm} = 0$, otherwise
Decision Variables	
x_{ij}	If disease i is treated by pesticide j, $x_{ij} = 1$. Otherwise,$x_{ij} = 0$
y_{ij}	The number of units needed to treat disease i by using pesticide j
c_i	Spraying times of pesticides needed to treat disease i
Auxiliary decision variable	
z_{ij}	$z_{ij} = x_{ij}y_{ij}$
d_{ijm}	If pesticide j and m can be mixed for treating disease i, $d_{ijm} = 1$. Otherwise, $d_{ijm} = 0$

III. PROBLEM STATEMENT AND MATHEMATICAL MODELS

A. PROBLEM STATEMENT

Before introducing the details of the mathematical models proposed by this paper, the notations and variables used in this paper are introduced, which are shown in table 1.

A farmer has several fields planted with different crop types. These crops are infected with pests and diseases. Let N represent the set of diseases (pests).h_i denotes the area of land suffering from disease i, where $i \in N$. M denotes the set of pesticides that could treat these diseases (pests). p_j represents the price of pesticide j per unit (bottle), where $j \in M$. We use $c_{ij} \in \{0, 1\}$ to denote whether pesticide j can treat the disease i or not. $c_{ij} = 1$ means pesticide j can treat the disease i. Usually, a disease of the crop can be treated with different types of pesticides. However, the therapeutic effect will differ for different pesticide types. Thus, we use $w_{ij} \in [0, 1]$ to represent the therapeutic effect of pesticide j on disease i. g_{ij} denotes per unit area of the crop that pesticide j can treat the disease

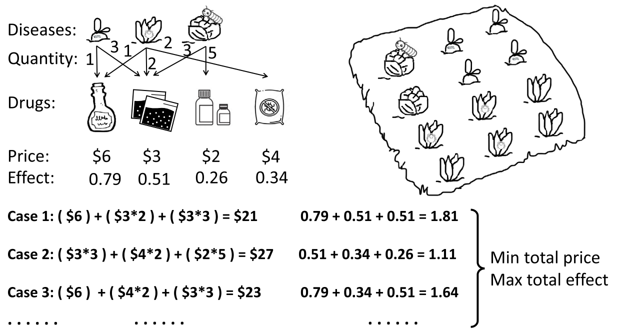


FIGURE 1. An example of the studied problem.

i. Due to the contraindications of chemical pesticides, not all pesticides can be mixed for pesticide spraying. k_{jm} is a binary value to represent pesticide j and m can be mixed or not.

To destroy or regulate the disease (pest) of the crop by using pesticides, a farmer has several goals, including minimizing the total cost of pesticides and maximizing the therapeutic effect. Figure 1 shows an example of the studied problem.

B. A BASIC MODEL

To address the core optimization goals of the proposed framework, two key objective functions are formulated, with clear alignment to practical application needs. Specifically, Objective Function 1 is designed to minimize the total cost associated with the spraying operation, ensuring economic feasibility of the solution. Complementarily, Objective Function 2 focuses on maximizing the therapeutic efficacy of the spraying process, which is defined as the degree of pest population suppression post-operation to meet the core agronomic or public health requirements of the scenario. The mathematical expressions for these two objective functions are formally presented as follows:

$$\mathcal{F}_1 = \sum_{i \in N} \sum_{j \in M} y_{ij} p_j \tag{1}$$

$$\mathcal{F}_2 = \sum_{i \in N} \frac{\sum_{j \in M} w_{ij} y_{ij} g_{ij}}{h_i} \tag{2}$$

Constraint (5) ensures that each disease is treated with at least one pesticide. Constraint (6) ensures that pesticide j can treat disease i. Constraint (7) ensures that when $y_{ij} = 0$, then $x_{ij} = 0$, that is, when pesticide j for treating disease i is not purchased, it means that pesticide j is not used to treat disease i. Constraint (8) ensures that when $x_{ij} = 0$, then $y_{ij} = 0$; that is, when pesticide j is not used to treat disease i, it is not necessary to buy pesticide j for treating disease i. Constraint (9) ensures that each area of the disease i is treated. Constraint (10) shows the upper bound of the y_{ij}. Constraints (11) and (12) are the decision variables.

$$P_1 \qquad Minimize \quad \mathcal{F}_1 \tag{3}$$

$$P_2 \qquad Maximize \quad \mathcal{F}_2 \tag{4}$$

subject to

$$\sum_{j \in M} x_{ij} \geq 1 \quad \forall i \in N \tag{5}$$

$$x_{ij} \leq e_{ij} \quad \forall i \in N, \forall j \in M \tag{6}$$

$$x_{ij} \leq x_{ij} y_{ij} \quad \forall i \in N, \forall j \in M \tag{7}$$

$$y_{ij} \leq x_{ij} y_{ij} \quad \forall i \in N, \forall j \in M \tag{8}$$

$$\sum_{j \in M} y_{ij} g_{ij} \geq h_i \quad \forall i \in N \tag{9}$$

$$y_{ij} \leq \begin{cases} \left\lceil \frac{h_i}{g_{ij}} \right\rceil & \forall i \in N, \forall j \in M, g_{ij} \neq 0 \\ 0 & \forall i \in N, \forall j \in M, g_{ij} = 0 \end{cases} \tag{10}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in N, \forall j \in M \tag{11}$$

$$y_{ij} \in \mathbb{N} \quad \forall i \in N, \forall j \in M \tag{12}$$

1) LINEARIZATION OF THE MODEL P_1 AND P_2

The model of P_1 and P_2 contains nonlinear constraints, which cannot be solved by many integer solvers. In this subsection, we introduce an auxiliary decision variable $z_{ij} \in N$, where $z_{ij} = x_{ij} y_{ij}$. The linearized model of P_1 and P_2 are shown in (13) and (14), respectively.

$$P_1^L \qquad Minimize \quad \mathcal{F}_1 \tag{13}$$

$$P_2^L \qquad Maximize \quad \mathcal{F}_2 \tag{14}$$

Subject to (5), (6), (9),(10),(11), (12), and

$$x_{ij} \leq z_{ij} \quad \forall i \in N, \forall j \in M \tag{15}$$

$$y_{ij} \leq z_{ij} \quad \forall i \in N, \forall j \in M \tag{16}$$

$$z_{ij} \leq a_{ij} x_{ij} \quad \forall i \in N, \forall j \in M \tag{17}$$

$$z_{ij} \leq y_{ij} \quad \forall i \in N, \forall j \in M \tag{18}$$

$$y_{ij} - a_{ij} \left(1 - x_{ij}\right) \leq z_{ij} \quad \forall i \in N, \forall j \in M \tag{19}$$

$$z_{ij} \in \mathbb{N} \quad \forall i \in N, \forall j \in M \tag{20}$$

Constraints (15),(16),(17),(18),(19),(20) are the linearization of constraints (7) and (8).

C. MODELS CONSIDERING THE MIXTURE OF TWO PESTICIDES

When applying pesticides to crops, farmers often combine different types of pesticides to reduce the frequency of spraying. However, as noted earlier, not all pesticides are chemically compatible for mixing with one another. If two or more pesticides in a specific group are incompatible for mixing, multiple spraying sessions are required to ensure all required pesticides are fully applied. Figure 2 provides an example of pesticide mixing.

Pesticide mixing combines two or more pesticide preparations to form a mixed liquid for spray application. Scientific and rational use of pesticides can save labour and time. In the following model, we aim to minimize the total number of times for spraying pesticides, considering the constraints that the two types of pesticides can not be mixed. Equation (21)

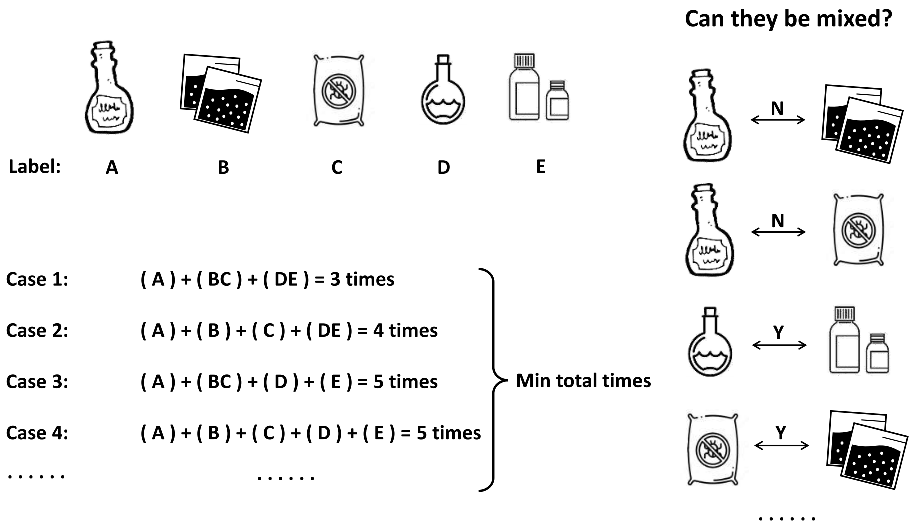


FIGURE 2. An example of pesticide mixing.

shows the objective function for minimizing the total number of times for spraying pesticides. Constraints (22) and (23) show the lower and upper bounds of the c_i. Constraint (24) ensures that only the pesticides allowed to be mixed can be assigned to a group. Constraint (25) ensures that only the pesticides used can be mixed and sprayed. Constraints (26) and (27) ensure that each pesticide can only be mixed with another pesticide. Constraints (28) is a logic constraint d_{ijm}. Constraint (29) defines the c_i, Constraint (30) ensures c_i is an integer.

$$P_3 \qquad Minimize \quad \mathcal{F}_3 = \sum_{i \in N} c_i \tag{21}$$

subject to

$$\frac{1}{2}\sum_{j \in M} x_{ij} \leq c_i \qquad \forall i \in N \tag{22}$$

$$c_i \leq \sum_{j \in M} x_{ij} \qquad \forall i \in N \tag{23}$$

$$d_{ijm}k_{jm} = 0 \qquad \forall i \in N, \forall j \in M, \forall m \in M \tag{24}$$

$$d_{ijm} \leq x_{ij} \qquad \forall i \in N, \forall j \in M, \forall m \in M \tag{25}$$

$$\sum_{m \in M} d_{ijm} \leq 1 \qquad \forall i \in N, \forall j \in M \tag{26}$$

$$\sum_{j \in M} d_{ijm} \leq 1 \qquad \forall i \in N, \forall m \in M \tag{27}$$

$$d_{ijm} = d_{imj} \qquad \forall i \in N, \forall j \in M, \forall m \in M \tag{28}$$

$$c_i = \sum_{j \in M} x_{ij} - \frac{1}{2}\sum_{j \in M}\sum_{m \in M} d_{ijm} \qquad \forall i \in N \tag{29}$$

$$c_i \in \mathbb{N} \qquad \forall i \in N \tag{30}$$

and (5), (6), (9)-(12), (15)-(20).

IV. SOLUTION APPROACH

Multiple objective optimization methods have been widely used for solving combinatorial problems with multiple objectives [27], [28], [29], [30]. This paper proposes a comprehensive framework for solving the studied problem, which is presented as follows.

A. FRAMEWORK

Figure 3 shows the framework for solving multi-objective optimization problems. The process begins with formulating the optimization problem related to pesticide mixture spraying. Subsequently, an appropriate algorithmic framework is selected to address the problem. The third step involves generating an initial solution set, which may be randomly initialized or manually configured based on prior knowledge. Following this, appropriate constraint-handling techniques is applied to ensure the feasibility of the solutions. The fifth step identifies optimal solutions from the multiple Pareto-optimal solutions generated, utilizing Multi-Criteria Decision Making (MCDM) techniques. Ultimately, the final optimal solution is derived.

B. META-HEURISTICS

A variety of meta-heuristic algorithms have been developed for multi-objective optimization. This paper employs seven representative algorithms to solve the pesticide matching problem, covering different optimization paradigms to ensure a comprehensive performance evaluation:

- **Classical algorithm:** Non-dominated Sorting Genetic Algorithm-2 (NSGA-II)
- **Reference point-based algorithms:** Reference Point-based NSGA-II (R-NSGA-II), Reference Point-based NSGA-III (R-NSGA-III)

- High-dimensional optimization algorithms: NSGA-III, Reference Vector Guided Evolutionary Algorithm (RVEA)
- **Indicator-based algorithm:** S-metric Selection Evolutionary Multi-objective Algorithm (SMS-EMOA)
- **Improved algorithm:** Unified NSGA-III (U-NSGA-III)
-

As we all know, the optimal solutions obtained by different algorithms are different in terms of the value of the function, running time, iterations, robustness, and spatial complexity. These algorithms differ in their approaches to convergence, diversity maintenance, computational efficiency, and adaptability. By comparing their performance, we can identify the most suitable algorithm for the pesticide matching problem and ensure the robustness of the solution. Figure 4 shows the selection of algorithms.

C. SOLUTION ENCODING AND DECODING

The meta-heuristic algorithms represent versatile computational frameworks designed to address various optimization problems. A critical component of these meta-heuristics lies in the processes of solution encoding and decoding, which are essential components of meta-heuristic algorithms and significantly influence the algorithms' effectiveness and computational efficiency. Properly designed encoding and decoding mechanisms ensure that the meta-heuristics can effectively explore the solution space and converge to high-quality solutions.

This paper encodes the solution as $Solution = \{Y, X, Z, D, C\}$ of length $3|N||M| + |N||M|^2 + |N|$. Additionally, Y represents $\{y_{11}, y_{12}, \ldots, y_{ij}\}$, X represents $\{x_{11}, x_{12}, \ldots, x_{ij}\}$, Z represents $\{z_{11}, z_{12}, \ldots, z_{ij}\}$, D represents $\{d_{111}, d_{112}, \ldots, d_{ijm}\}$ and C represents $\{c_1, c_2, \ldots, c_i\}$ with $i = 1, 2, \ldots, |N|$,$j = 1, 2, \ldots, |M|$,$m = 1, 2, \ldots, |M|$, where y_{ij} represents that the number of bottles needed to treat disease i with pesticide j, $x_{ij} = 1$ means treating disease i with the pesticide j, z_{ij} is the product of x and y, which is a variable introduced to linearize the model,$d_{ijm} = 1$ denotes the pesticide j_1 and the pesticide j_2 be mixed and c_i represents the spraying times of pesticides needed to treat disease i. Figure 5 shows the structure of the solution, and we can get any variable by index.

Algorithm 1 provides a detailed illustration of the solution process. Initially, the solution structure is defined, and a set of initial solutions is generated through random initialization. Subsequently, crossover and mutation operations are applied to the solutions to produce new candidate solutions. Finally, all solutions are evaluated against predefined criteria, and high-quality solutions are selected for subsequent iterations to refine the solution set progressively.

D. CROSSOVER AND MUTATION OPERATION

For crossover operation, we randomly select two points with a certain probability, and the genetic material between these points is exchanged between the parents. Figure 6 shows the details of the crossover operation. Variable x_{ij} is a binary variable. That is, its value can only be 0 or 1. And $x_{ij} = 1$ means treating disease i with the pesticide j. In Figure 6, we can see that two points were randomly selected from the parent 1 and the parent 2, and the values between these two points were exchanged and updated as the offspring 1 and the offspring 2.

We randomly select several mutation points with a certain probability. For example, because the variable x is a binary variable, the variable x can only mutate from 0 to 1 or from 1 to 0. Figure 6 shows the details of the mutation operation. After the mutation operation, the children 1 and 2 randomly select some points and update them to offspring 1 and offspring 2.

E. CONSTRAINT HANDLING

After crossover and mutation operations, the offspring can result in infeasible solutions, meaning certain constraints may be violated. For instance, when $x_{12} = 0$, it means that the first disease is not treated with the second pesticide, so correspondingly, y_{12} accordingly be 0, that is, the second pesticide is not bought to treat the first disease. Introducing a repair operation enables the problem's solution results to better align with real-world scenarios.

Step 4 in Figure 3 shows the selection of the constraint handling method, and the appropriate constraint handling method is also determined according to performance indicators such as objective function value, running time, and iteration times.

To convert these invalid constraints into valid constraints, this paper adopts feasibility first (Parameter-less Approach) and constraint violation (CV) methods, which are called repair operations in this paper. Algorithm 2 shows the details of the repair operation.

In Figure 7, we can see that when $x_{12} = 0$, $x_{13} = 0$ and $x_{16} = 0$, $y_{12}y_{13}$ and y_{16} are not equal to 0, so they need to be repaired to be 0.

F. MULTI-CRITERIA DECISION MAKING (MCDM)

After obtaining a set of non-dominated solutions for the problem, we need to determine whether this set of solutions consists of only a small number of solutions or even a single solution—an assessment necessary for subsequent decision-making; this is a process that is primarily divided into the following steps. Algorithm 3 presents the detailed procedure for multi-criteria decision-making (MCDM).

Step 1: Normalize the objective functions to eliminate the influence of different dimensions.

Step 2: Determine the weights of the objective functions using the entropy weight method.

Step 3: Identify the optimal solution from the non-dominated set.

Figure 8 shows the optimal solution selection process from multiple groups of non-dominated solutions by performing MCDM operations.

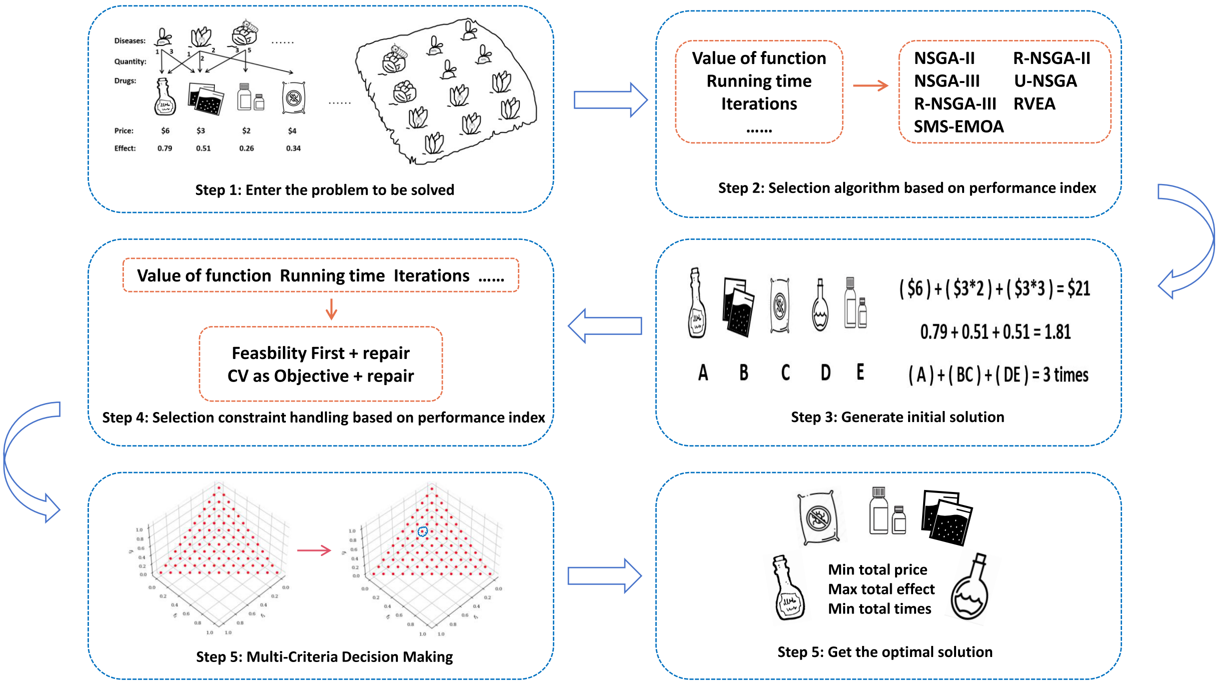


FIGURE 3. Framework for solving multi-objective optimization problems.

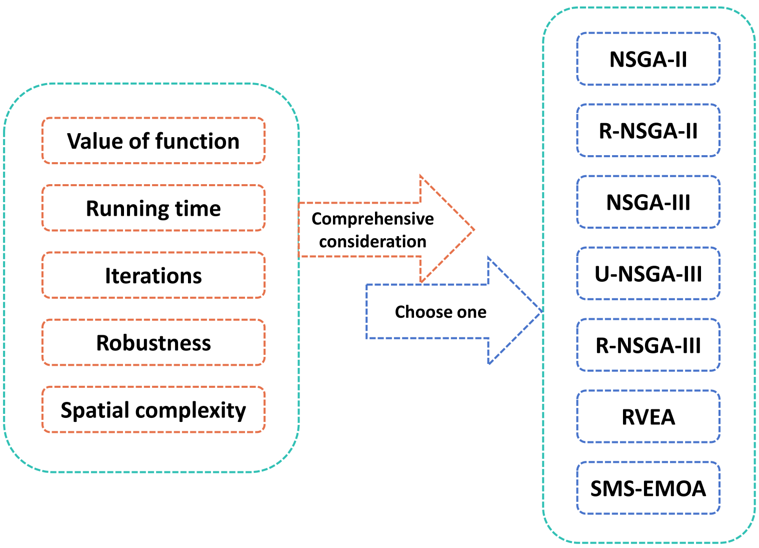


FIGURE 4. The process of algorithm selection.

V. EXPERIMENT

A. PARAMETER SETTING

1) EXPERIMENTAL ENVIRONMENT

In this section, we will carry out experiments to verify the effectiveness of the proposed algorithm. All experiments are run on a computer with an Intel Core i9-13900K CPU @3.0GHz and 16.0GB of RAM. Our algorithms are implemented in the Python programming language (version 3.10.12).

2) INSTANCES

In this paper, thirty instances are adopted with various scales of diseases $|N|$ and pesticides $|M|$, which are shown in Table 2. These cases are composed of small-scale cases, medium-scale cases, and large-scale cases. Because the linear model should linearize the nonlinear model, the number of constraints in the linear model is always greater than that in the nonlinear model.

B. COMPARISON OF NONLINEAR MODEL AND LINEAR MODEL

To evaluate and compare the performance of nonlinear and linear models, we conducted experiments using three distinct objective functions. The nonlinear models are solved using LINGO, while the linear models are solved using the Python MIP library with the CBC solver.

Encoded variable	Index	Actual variable	Type
Y	$0, 1, \dots, NM-1$	$y_{11}, y_{12}, \dots, y_{ij}$	integer
X	$NM, NM+1, \dots, 2NM-1$	$x_{11}, x_{12}, \dots, x_{ij}$	binary
Z	$2NM, 2NM+1, \dots, 3NM-1$	$z_{11}, z_{12}, \dots, z_{ij}$	integer
D	$3NM, 3NM+1, \dots, 3NM+NM^2$-1	$d_{111}, d_{112}, \dots, d_{ijm}$	binary
C	$3NM+NM^2$, $3NM+NM^2$+1,..., $3NM+NM^2$+N-1	$c_1, c_2, \dots, c_i$	integer

FIGURE 5. Structure of solution.

Type	Binary							
Variable	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	...	x_{ij}
Parent 1	1	0	0	1	0	0	...	0
Parent 2	0	1	1	0	0	1	...	0

Crossover

Type	Binary							
Variable	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	...	x_{ij}
Child 1	1	1	1	0	0	0	...	0
Child 2	0	0	0	1	0	1	...	0

Mutation

Type	Binary							
Variable	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	...	x_{ij}
Child' 1	0	1	1	0	0	0	...	1
Child' 2	0	0	0	1	0	0	...	0

FIGURE 6. Crossover and Mutation operation.

Variable 1	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	...	x_{ij}
Initial value	1	0	0	0	1	0	...	0
Variable 1	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	y_{16}	...	y_{ij}
Initial value	0	20	30	0	53	12	...	0

Repair

Variable 1	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	...	x_{ij}
Initial value	1	0	0	0	1	0	...	0
Variable 1	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	y_{16}	...	y_{ij}
Initial value	0	0	0	0	53	0	...	0

FIGURE 7. Repair operation.

Table 3 shows the solution results when optimizing f_1. According to Table 3, it can be found that in small-scale cases, the running time of the linear model solved by PYTHON is slightly longer than that of the linear model solved by LINGO and the nonlinear model solved by LINGO. In medium-scale and large-scale cases, the running time of the linear model solved by PYTHON is the shortest, followed by the linear model solved by LINGO, and the running time of the nonlinear model solved by LINGO is the longest. Table 4 shows the solution results when optimizing f_2, and Table 5 shows the solution results when optimizing f_3. It can be found that in small-scale cases, the running time of the linear model solved by PYTHON is slightly longer than that of the linear model solved by LINGO and the nonlinear model solved by LINGO. In medium-scale and large-scale cases, the running time of the linear model solved by PYTHON is the shortest, followed by the nonlinear model solved by LINGO, and the running time of the linear model solved by LINGO is the longest.

These results confirm that linearization significantly improves the computational efficiency of the model, especially for medium- and large-scale problems, making it more suitable for practical agricultural applications.

C. COMPARISON OF DIFFERENT SOLVERS

This part compares the performance of single-objective models solved by different solvers, including Coin-OR Branch-and-Cut(CBC), GUROBI, and CPLEX. CBC and GUROBI are called by the Python Mixed Integer Programming (MIP) library. CPLEX is called by the DOCPLEX library of PYTHON. Table 6, Table 7 and Table 8 show the results of optimizing f_1, f_2, and f_3, respectively. In Table 6, it can be found that the running time of the GUROBI solver is the shortest in different scale cases, while the running time of the CBC solver is shorter than that of the CPLEX solver in small-scale cases, but the opposite is true in large-scale cases. In Table 7 and Table 8, it can be found that the GUROBI solver has the shortest running time in different scale cases, followed by the CPLEX solver, and the CBC solver has the longest running time.

D. COMPARISON OF DIFFERENT ALGORITHMS

This section evaluates the performance of seven state-of-the-art multi-objective optimization algorithms across varying iteration counts, using a representative instance with N=3 and M=4. The maximum number of iterations ranges from 3,000 to 300,000, allowing for a comprehensive analysis of algorithmic convergence and efficiency. The experimental results are shown in Table 9.

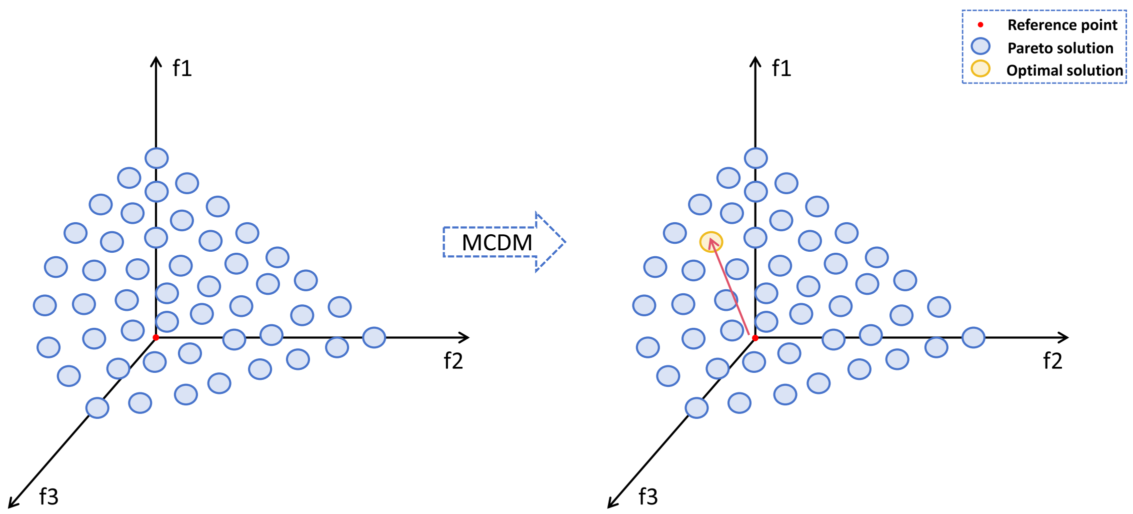


FIGURE 8. MCDM operation.

As shown in Table 9, the SMS-EMOA algorithm consistently generates the highest number of non-dominated solutions across all iteration counts, but its running time is much longer than that of other algorithms. The non-dominated solutions obtained by the R-NSGA-II algorithm and the R-NSGA-III algorithm account for more than 50% of the total solution set. Still, the number of solutions obtained by the R-NSGA-III algorithm is the least among all algorithms, and only three solutions are obtained. Although the RVEA algorithm has the shortest running time, the quality of its solution is poor.

Figure 9 shows the spatial distribution of solutions and Pareto solutions for different iterations. According to Figure 9, it can be observed that with the increase of iteration times, the solutions obtained by each algorithm gradually converge to the Pareto solution.

Figure 10 shows the convergence of each algorithm when it is iterated 500 times.

- NSGA-II algorithm: At least one feasible solution is obtained at Generation 32 with 4775 function evaluations.
- R-NSGA-II algorithm: At least one feasible solution is found at Generation 9 with 1486 function evaluations, and the entire population becomes feasible at Generation 31 with 4632 function evaluations.
- NSGA-III algorithm: At least one feasible solution is found at Generation 9 with 1486 function evaluations, and the entire population becomes feasible at Generation 32 with 4775 function evaluations.
- U-NSGA-III algorithm: At least one feasible solution is found at Generation 9 with 1486 function evaluations, and the entire population becomes feasible at Generation 31 with 4632 function evaluations.
- R-NSGA-III algorithm: At least one feasible solution is found at Generation 8 with 1328 function evaluations, and the entire population becomes feasible at Generation 36 with 5332 function evaluations.
- SMS-EMOA algorithm: At least one feasible solution is found at Generation 9 with 1486 function evaluations, and the entire population becomes feasible at Generation 32 with 4775 function evaluations.
- RVEA algorithm: At least one feasible solution is found at Generation 5 with 914 function evaluations, but not all individuals in the population are feasible even after 500 iterations.

Among these algorithms, the R-NSGA-II and U-NSGA-III algorithms have the best convergence, while the RVEA algorithm exhibits the worst.

The Hypervolume (HV) index is a performance metric that quantifies the volume of the multidimensional space dominated by the solution set in the objective space, bounded by a predefined reference point. Precisely, it measures the region enclosed by the Pareto-optimal solutions and the reference point, providing a comprehensive indicator of both convergence and diversity of the obtained solutions. A higher Hypervolume value indicates superior performance regarding solution quality and coverage of the Pareto front. The hypervolume index evaluation method is a Pareto-compliant evaluation method. If one solution set S is superior to another solution set S', then the Hypervolume index of the solution set S will be greater than that of solution set S'. Figure 11 shows the Hypervolume of each algorithm. The NSGA-II algorithm, R-NSGA-II algorithm, NSGA-III algorithm, U-NSGA-III algorithm, and SMS-EMOA algorithm are equivalent, while the convergence effects of the RVEA algorithm perform best.

Figure 12 illustrates the runtime performance metrics of each algorithm under consideration. These metrics capture the evolution of the solution set in the objective space across successive generations, utilizing the concept of survival to demonstrate the iterative improvement of the algorithm visually. Additionally, this index serves as a criterion for determining the termination of the multi-objective optimization algorithm, particularly in cases where a predefined termination condition is not explicitly specified. By mon-

Algorithm 1 The Process of Solution

```
input  : Number of diseases |N| and pesticides |M|,
         field area parameters h and g_ij, pesticides'
         prices p_j,therapeutic effect parameters w_ij e_ij
         and k_jm
output: Initial solution
a_ij ← h and g_ij;// Get the value range of y_ij
Set Gen ← 1;
while Gen ≤ n_gen do
    for i ← 0 to N − 1 do
        c_i ← np.random(0, |M|) ;// Randomly generate the value of c_i
        for j ← 0 to M − 1 do
            y_ij ← np.random(0, a_ij) ;// Randomly generate the value of y_ij
            x_ij ← np.random(0, 1) ;// Randomly generate the value of x_ij
            z_ij ← np.random(0, a_ij) ;// Randomly generate the value of z_ij
            for m ← 0 to M − 1 do
                d_ijm ← np.random(0, 1) ;// Randomly generate the value of d_ijm
            end
        end
    end
    sampling ← np.hstack(y_ij, x_ij, z_ij, d_ijm, c_i) ;// Randomly generate a set of solutions
    crossover ← Crossover(prob) ;// Crossover is carried out with a certain probability.
    mutation ← MyMutation(prob) ;// Mutation is carried out with a certain probability.
    Gen ← i + 1 ;
end
```

Algorithm 2 The Process of Repair Operation

```
input  : The initial solution obtained
output: Feasible solution
Set p ← 1;
while p ≤ X.shape[0] do
    get x_ij y_ij z_ij d_ijm and c_i from X[p, :] ;
    for i ← 0 to N − 1 do
        c_i ≤ M ;// Ensure that the value of c_i is within a reasonable range.
        for j ← 0 to M − 1 do
            y_ij ≤ a_ij ;// Ensure that the value of y_ij is within a reasonable range.
            for m ← 0 to M − 1 do
                if x_ij = 0 then
                    y_ij ← 0;// Ensure that the value of y_ij is logical.
                    d_ijm ← 0;// Ensure that the value of d_ijm is logical.
                end
            end
        end
    end
    p ← i + 1 ;
end
```

Algorithm 3 The Process of MCDM

```
input  : Multi-group non-dominant solutions F
output: A set of optimal solutions
F' ← (F − F.min)/(F.max − F.min);// Step 1
weights ← Entropy(F');// Step 2
decomp = ASF();
a = decomp.do(F', 1/weights).argmin();
get the set of optimal solutions F'[a];// Step 3
```

itoring solutions' convergence behavior and diversity, this metric provides a robust mechanism for assessing algorithmic performance and guiding the optimization process. As shown in Figure 12, it can be found that all the algorithms except R-NSGA-III improved significantly before the 25th generation. At the same time, the performance of R-NSGA-III was poor, and the improvement of each generation was not obvious.

E. COMPARISON OF DIFFERENT CONSTRAINT HANDLING METHODS

This section compares two widely used constraint handling techniques—Feasibility First (parameter-less approach) and Constraint Violation (CV)—to evaluate their effectiveness in solving the pesticide matching problem. Experiments are conducted on the same instance with N=3 and M=4. Table 3, Table 4 and 5 shows the results of a single objective function for optimizing f_1, f_2, and f_3, with a focus on both single-objective and multi-objective optimization performance.

According to Table 10, we can find that for solving different objective functions, the constraint violation method can always get the optimal solution with fewer iterations. For solving other objective functions except f_1, the running time of the constraint violation method under different iterations is shorter.

F. SINGLE-OBJECTIVE OPTIMIZATION RESULTS

Notably, both methods achieve identical or nearly identical objective function values for each optimization task, indicat-

TABLE 2. Instance settings.

Case ID	$\lvert N\rvert$	$\lvert M\rvert$	Variables(NL)	Variables(L)	Constraints(NL)	Constraints(L)
case 1	3	6	147	165	339	393
case 2	3	10	363	393	795	885
case 3	4	6	196	220	452	524
case 4	4	7	256	284	580	664
case 5	4	8	324	356	724	820
case 6	4	9	400	436	884	992
case 7	4	10	484	524	1060	1180
case 8	5	8	405	445	905	1025
case 9	5	9	500	545	1105	1240
case 10	5	10	605	655	1325	1475
case 11	9	18	3249	3411	6849	7335
case 12	10	20	4410	4610	9250	9850
case 13	20	40	33620	34420	68900	71300
case 14	30	60	111630	113430	226950	232350
case 15	40	80	262440	265640	531400	541000
case 16	50	100	510050	515050	1030250	1045250
case 17	51	102	541059	546261	1092675	1108281
case 18	54	108	641574	647406	1294974	1312470
case 19	59	118	835499	842461	1685099	1705985
case 20	65	130	1115465	1123915	2248025	2273375
case 21	70	140	1391670	1401470	2803150	2832550
case 22	75	150	1710075	1721325	3442875	3476625
case 23	80	160	2073680	2086480	4173200	4211600

* $\lvert N\rvert$ represents the number of diseases; $\lvert M\rvert$ represents the number of pesticides;
* NL represents nonlinear model; L represents linear model.

TABLE 3. Optimization of function f_1^*.

CASE ID	PYTHON MIP(L)				LINGO(L)				LINGO(NL)			
	f_1^*	f_2	f_3	$t(s)$	f_1^*	f_2	f_3	$t(s)$	f_1^*	f_2	f_3	$t(s)$
1	389.4	1.76	3	0.06	389.4	1.76	3	0.06	389.4	1.76	3	0.06
2	425.2	2.10	5	0.11	425.2	2.07	5	0.08	425.2	2.07	5	0.08
3	485.7	2.12	4	0.06	485.7	2.12	4	0.05	485.7	2.12	4	0.08
4	1280.4	2.41	5	0.74	1280.4	2.41	5	0.06	1280.4	2.41	5	0.07
5	328.3	1.33	4	0.10	328.3	1.33	4	0.05	328.3	1.33	4	0.05
6	615.3	2.05	4	0.10	615.3	2.05	4	0.06	615.3	2.05	4	0.07
7	288.6	2.00	6	0.10	288.6	2.00	6	0.07	288.6	2.00	6	0.09
8	2002.3	2.42	6	0.17	2002.3	2.42	6	0.10	2002.3	2.42	6	0.09
9	342.0	3.17	5	0.10	342.0	3.17	5	0.06	342.0	3.17	5	0.06
10	453.9	1.59	5	0.10	453.9	1.59	5	0.07	453.9	1.59	5	0.07
11	748.1	3.86	11	0.20	748.1	3.86	11	0.21	748.1	3.86	11	0.28
12	983.2	4.94	10	0.24	983.2	4.94	10	0.24	983.2	4.94	10	0.22
13	619.0	12.00	21	1.24	619.0	12.00	21	3.23	619.0	12.00	21	2.84
14	929.0	13.62	35	4.75	929.0	13.62	35	18.55	929.0	13.62	35	18.34
15	2171.1	21.22	59	10.75	2171.1	21.22	59	75.75	2171.1	21.22	59	83.33
16	1717.0	27.21	65	24.12	1717.0	27.21	65	217.61	1725.2	27.37	52	>3600
17	1003.9	27.81	59	23.42	1003.9	27.81	59	239.43	1004.3	27.80	60	>3600
18	737.7	27.45	81	44.86	737.7	27.45	81	322.91	744.4	27.65	61	>3600
19	2161.2	30.06	68	54.94	2161.2	30.06	68	489.13	2166.1	30.06	65	>3600
20	1902.1	34.17	79	53.18	1902.1	34.17	79	788.95	1911.6	34.10	68	>3600
21	2515.3	35.09	96	86.43	2515.3	35.09	96	23.42	2515.3	35.09	96	63.78
22	2277.7	40.96	97	160.77	2277.7	40.86	97	28.87	2277.7	40.91	97	77.68
23	1519.7	43.83	98	167.47	1519.7	43.83	98	34.94	1519.7	43.83	98	94.29

* NL represents nonlinear model;
* L represents linear model;
* >3600 means that the running time to find the optimal solution is longer than 3600s. In this case, the feasible solutions obtained when the running time is 3600s are listed in the table.

TABLE 4. Optimization of function f_2^*.

CASE ID	PYTHON MIP(L)				LINGO(L)				LINGO(NL)			
	f_1	f_2^*	f_3	$t(s)$	f_1	f_2^*	f_3	$t(s)$	f_1	f_2^*	f_3	$t(s)$
1	46577.7	9.69	18	0.06	46577.7	9.69	18	0.06	46577.7	9.69	18	0.06
2	66288.7	16.70	30	0.07	66288.7	16.70	30	0.06	66288.7	16.70	30	0.06
3	13476.5	6.80	17	0.06	13476.5	6.80	13	0.06	13476.5	6.80	12	0.06
4	149452.8	15.89	28	0.06	149452.8	15.89	28	0.05	149452.8	15.89	28	0.06
5	65696.8	15.16	32	0.09	65696.8	15.16	32	0.06	65696.8	15.16	32	0.06
6	36127.2	17.77	36	0.07	36127.2	17.77	36	0.06	36127.2	17.77	36	0.06
7	58178.5	20.24	40	0.07	58178.5	20.24	40	0.06	58178.5	20.24	40	0.06
8	82131.8	18.46	40	0.07	82131.8	18.46	40	0.06	82131.8	18.46	40	0.06
9	77577.9	23.08	45	0.08	77577.9	23.08	45	0.07	77577.9	23.08	45	0.06
10	115568.2	25.90	50	0.08	115568.2	25.90	50	0.07	115568.2	25.90	50	0.06
11	599129.3	83.73	162	0.17	599129.3	83.73	162	0.17	599129.3	83.73	162	0.14
12	502625.3	107.33	200	0.21	502625.3	107.33	200	0.22	502625.3	107.33	200	0.20
13	1766608.9	413.82	800	1.10	1766608.9	413.82	800	2.80	1766608.9	413.82	800	2.54
14	4771101.4	921.70	1800	3.47	4771101.4	921.70	1800	18.11	4771101.4	921.70	1800	17.12
15	8193816.6	1620.40	3200	8.56	8193816.6	1620.40	3200	72.36	8193816.6	1620.40	3200	69.17
16	13746281.9	2594.40	5000	19.96	13746281.9	2594.40	5000	215.65	13746281.9	2594.40	5000	206.10
17	14044425.4	2690.45	5202	21.38	14044425.4	2690.45	5202	236.64	14044425.4	2690.45	5202	227.45
18	1634454.2	2955.79	5832	26.19	1634454.2	2955.79	5832	313.47	1634454.2	2955.79	5832	301.27
19	19186676.8	3553.02	6962	35.75	19186676.8	3553.02	6962	485.27	19186676.8	3553.02	6962	469.94
20	22785091.5	4395.47	8450	52.64	22785091.5	4395.47	8450	783.46	22785091.5	4395.47	8450	760.18
21	25524806.2	5045.33	9800	68.72	25524806.2	5045.33	9800	1121.04	25524806.2	5045.33	9800	1094.65
22	28971403.7	5863.31	11250	91.22	28971403.7	5863.31	11250	1615.61	28971403.7	5863.31	11250	1538.67
23	33906277.2	6653.00	12800	118.48	33906277.2	6653.00	12800	2253.72	33906277.2	6653.00	12800	2128.48

* NL represents nonlinear model;
* L represents linear model.

TABLE 5. Optimization of function f_3^*.

CASE ID	PYTHON MIP(L)				LINGO(L)				LINGO(NL)			
	f_1	f_2	f_3^*	$t(s)$	f_1	f_2	f_3^*	$t(s)$	f_1	f_2	f_3^*	$t(s)$
1	1008.0	1.15	3	0.06	29583.0	1.03	3	0.06	9436.0	1.99	3	0.06
2	11522.7	0.93	3	0.08	596.4	2.03	3	0.06	1930.0	2.38	3	0.06
3	829.4	2.46	4	0.11	1189.1	1.30	4	0.06	1456.1	1.29	4	0.06
4	39900.0	2.11	4	0.06	6217.6	2.75	4	0.05	3181.8	2.04	4	0.06
5	2985.6	1.74	4	0.06	2590.8	2.88	4	0.04	2691.5	2.12	4	0.06
6	4404.4	2.83	4	0.07	8466.6	1.56	4	0.05	1901.1	1.49	4	0.06
7	1206.9	1.92	4	0.08	4923.2	2.02	4	0.06	3180.1	1.64	4	0.05
8	8645.0	2.02	5	0.07	3426.3	2.13	5	0.06	10992.6	1.60	5	0.05
9	3019.3	2.30	5	0.07	17395.2	3.31	5	0.06	3256.6	3.95	5	0.06
10	17130.4	2.04	5	0.08	8282.4	2.78	5	0.06	3205.8	2.57	5	0.06
11	10911.2	4.36	9	0.17	21507.8	5.19	9	0.17	20420.4	4.64	9	0.14
12	17134.7	2.94	10	0.21	14385.6	5.91	10	0.22	17134.7	2.94	10	0.20
13	5547.3	12.48	20	1.31	1452.8	11.45	20	2.83	8638.8	12.89	20	2.57
14	109657.2	16.96	30	8.00	53179.0	15.57	30	18.53	110144.7	16.87	30	17.43
15	145923.3	16.76	40	49.46	106735.2	20.80	40	73.29	145923.3	16.76	40	72.18
16	171761.8	24.55	50	27.63	129928.7	27.62	50	216.37	40221.7	29.02	50	210.82
17	248854.5	23.23	51	30.12	109867.8	25.20	51	238.61	47278.7	25.82	51	230.01
18	21384.1	26.16	54	37.68	32056.8	27.31	54	314.32	21384.1	26.16	54	321.49
19	246961.6	29.29	59	52.71	379661.0	31.35	59	488.55	47825.4	29.36	59	470.95
20	198371.6	36.18	65	75.05	188595.0	36.23	65	786.23	52075.1	33.42	65	760.93
21	45723.5	30.12	70	102.74	9657.6	34.30	70	1129.47	54191.0	38.66	70	1106.00
22	54267.2	38.71	75	131.59	199790.4	40.24	75	1609.35	58084.7	40.29	75	1607.30
23	492140.3	43.46	80	169.84	234600.1	38.48	80	2164.55	60676.2	41.44	80	2125.09

* NL represents nonlinear model;
* L represents linear model.

ing that solution quality is not compromised by the choice of constraint handling technique. The primary difference lies in computational efficiency, with CV demonstrating superior speed for most objectives.

TABLE 6. Comparison of f_1^* with different solvers.

CASE ID	CBC				GUROBI				CPLEX			
	f_1^*	f_2	f_3	$t(s)$	f_1^*	f_2	f_3	$t(s)$	f_1^*	f_2	f_3	$t(s)$
1	389.4	1.76	3	**0.08 ± 0.09**	389.4	1.76	3	**0.04 ± 0.01**	389.4	1.76	3	**0.25 ± 0.13**
2	425.2	2.10	5	**0.61 ± 0.09**	425.2	2.10	5	**0.05 ± 0.01**	425.2	2.10	5	**0.21 ± 0.09**
3	485.7	2.12	4	**0.05 ± 0.00**	485.7	2.12	4	**0.04 ± 0.00**	485.7	2.12	4	**0.14 ± 0.10**
4	1280.4	2.41	5	**0.08 ± 0.00**	1280.4	2.41	5	**0.04 ± 0.00**	1280.4	2.41	5	**0.28 ± 0.11**
5	328.3	1.33	4	**0.06 ± 0.00**	328.3	1.33	4	**0.04 ± 0.00**	328.3	1.33	4	**0.25 ± 0.10**
6	615.3	2.05	4	**0.05 ± 0.01**	615.3	2.05	4	**0.05 ± 0.01**	615.3	2.05	4	**0.41 ± 0.04**
7	288.6	2.00	6	**0.06 ± 0.01**	288.6	2.00	6	**0.04 ± 0.00**	288.6	2.00	6	**0.27 ± 0.13**
8	2002.3	2.42	6	**0.09 ± 0.01**	2002.3	2.42	6	**0.04 ± 0.00**	2002.3	2.42	6	**0.45 ±0.11**
9	342.0	3.17	5	**0.07 ± 0.00**	342.0	3.17	5	**0.04 ± 0.00**	342.0	3.17	5	**0.57 ± 0.35**
10	453.9	1.59	5	**0.10 ± 0.05**	453.9	1.59	5	**0.05 ± 0.00**	453.9	1.59	5	**0.44 ± 0.19**
11	748.1	3.86	11	**0.17 ± 0.07**	748.1	3.86	11	**0.26 ± 0.35**	748.1	3.86	11	**1.01 ± 0.27**
12	983.2	4.94	10	**0.18 ± 0.04**	983.2	4.94	10	**0.13 ± 0.09**	983.2	4.94	10	**0.84 ± 0.24**
13	619.0	12.03	21	**0.86 ± 0.05**	619.0	12.03	21	**0.53 ± 0.22**	619.0	12.03	21	**2.4 ± 0.34**
14	929.0	13.62	35	**3.47 ± 0.08**	929.0	13.62	35	**1.43 ± 0.10**	929.0	13.62	35	**5.57 ± 1.05**
15	2171.1	21.22	59	**7.54 ± 0.08**	2171.1	21.22	59	**3.29 ± 0.10**	2171.1	21.22	59	**11.51 ± 0.94**
16	1717.0	27.21	65	**17.73 ± 0.10**	1717.0	27.21	65	**6.29 ± 0.25**	1717.0	27.21	65	**20.1 ± 0.88**
17	1003.9	27.81	59	**17.64 ± 0.68**	1003.9	27.81	59	**6.61 ± 0.14**	1003.9	27.81	59	**21.68 ± 0.93**
18	737.7	27.45	81	**32.62 ± 0.63**	737.7	27.45	81	**7.94 ± 0.27**	737.7	27.45	81	**24.83 ± 1.35**
19	2161.2	30.06	68	**41.26 ± 0.98**	2161.2	30.06	68	**10.28 ± 0.31**	2161.2	30.06	68	**32.25 ± 1.45**
20	1902.1	34.17	79	**38.77 ± 0.20**	1902.1	34.17	79	**13.71 ± 0.32**	1902.1	34.17	79	**40.7 ± 0.99**
21	2515.3	35.09	96	**70.63 ± 2.19**	2515.3	35.09	96	**17.22 ± 0.54**	2515.3	35.09	96	**49.76 ± 1.06**
22	2277.7	40.96	97	**132.35 ± 3.52**	2277.7	40.86	97	**21.40 ± 0.71**	2277.7	40.91	97	**60.69 ± 1.54**
23	1519.7	43.83	98	**142.65 ± 3.82**	1519.7	43.83	98	**26.29 ± 0.77**	1519.7	43.83	98	**73.43 ± 1.39**

TABLE 7. Comparison of f_2^* with different solvers.

CASE ID	CBC				GUROBI				CPLEX			
	f_1	f_2^*	f_3	$t(s)$	f_1	f_2^*	f_3	$t(s)$	f_1	f_2^*	f_3	$t(s)$
1	46577.7	9.69	18	**0.08 ± 0.10**	46577.7	9.69	18	**0.04 ± 0.00**	46577.7	9.69	18	**0.05 ± 0.00**
2	66288.7	16.70	30	**0.08 ± 0.09**	66288.7	16.70	30	**0.06 ± 0.06**	66288.7	16.70	30	**0.06 ± 0.01**
3	13476.5	6.80	17	**0.07 ± 0.08**	13476.5	6.80	17	**0.04 ± 0.00**	13476.5	6.80	17	**0.08 ± 0.08**
4	149452.8	15.89	28	**0.05 ± 0.00**	149452.8	15.89	28	**0.04 ± 0.00**	149452.8	15.89	28	**0.05 ± 0.00**
5	65696.8	15.16	32	**0.05 ± 0.00**	65696.8	15.16	32	**0.04 ± 0.01**	65696.8	15.16	32	**0.06 ± 0.01**
6	36127.2	17.77	36	**0.05 ± 0.00**	36127.2	17.77	36	**0.04 ± 0.01**	36127.2	17.77	36	**0.06 ± 0.00**
7	58178.5	20.24	40	**0.05 ± 0.00**	58178.5	20.24	40	**0.04 ± 0.00**	58178.5	20.24	40	**0.06 ± 0.00**
8	82131.8	18.46	40	**0.05 ± 0.00**	82131.8	18.46	40	**0.04 ± 0.00**	82131.8	18.46	40	**0.06 ± 0.00**
9	77577.9	23.08	45	**0.06 ± 0.01**	77577.9	23.08	45	**0.04 ± 0.01**	77577.9	23.08	45	**0.06 ± 0.00**
10	115568.2	25.90	50	**0.07 ± 0.02**	115568.2	25.90	50	**0.04 ± 0.01**	115568.2	25.90	50	**0.07 ± 0.02**
11	599129.3	83.73	162	**0.16 ± 0.09**	599129.3	83.73	162	**0.13 ± 0.14**	599129.3	83.73	162	**0.17 ± 0.05**
12	502625.3	107.33	200	**0.16 ± 0.02**	502625.3	107.33	200	**0.09 ± 0.01**	502625.3	107.33	200	**0.20 ± 0.02**
13	1766608.9	413.82	800	**0.88 ± 0.24**	1766608.9	413.82	800	**0.51 ± 0.20**	1766608.9	413.82	800	**1.07 ± 0.01**
14	4771101.4	921.70	1800	**2.55 ± 0.17**	4771101.4	921.70	1800	**1.42 ± 0.12**	4771101.4	921.70	1800	**4.14 ± 0.78**
15	8193816.6	1620.40	3200	**6.37 ± 0.19**	8193816.6	1620.40	3200	**3.27 ± 0.08**	8193816.6	1620.40	3200	**8.84 ± 0.57**
16	13746281.9	2594.44	5000	**14.44 ± 0.38**	13746281.9	2594.44	5000	**6.31 ± 0.16**	13746281.9	2594.44	5000	**16.78 ± 0.50**
17	14044425.4	2690.45	5202	**15.57 ± 0.44**	14044425.4	2690.45	5202	**6.67 ± 0.09**	14044425.4	2690.45	5202	**18.13 ± 1.19**
18	1634454.2	2955.79	5832	**19.31 ± 0.51**	1634454.2	2955.79	5832	**7.92 ± 0.23**	1634454.2	2955.79	5832	**21.18 ± 1.44**
19	19186676.8	3553.02	6962	**26.80 ± 1.10**	19186676.8	3553.02	6962	**10.19 ± 0.23**	19186676.8	3553.02	6962	**27.76 ± 1.16**
20	22785091.5	4395.47	8450	**38.75 ± 0.65**	22785091.5	4395.47	8450	**13.68 ± 0.19**	22785091.5	4395.47	8450	**36.07 ± 0.72**
21	25524806.2	5045.33	9800	**51.90 ± 1.96**	25524806.2	5045.33	9800	**17.21 ± 0.38**	25524806.2	5045.33	9800	**45.59 ± 0.54**
22	28971403.7	5863.31	11250	**71.60 ± 4.96**	28971403.7	5863.31	11250	**21.29 ± 0.54**	28971403.7	5863.31	11250	**55.78 ± 1.19**
23	33906277.2	6653.00	12800	**89.80 ± 2.29**	33906277.2	6653.00	12800	**26.15 ± 0.56**	33905580.0	6652.94	12799	**67.44 ± 1.12**

1) MULTI-OBJECTIVE OPTIMIZATION RESULTS

The following contents show the multi-objective optimization of the Feasibility First method and Constraint Violation with different iterations. We did a total of 6 sets of tests, from 3000 iterations to 300000 iterations. According to Table 10, it can be found that the total number of solutions and the number of non-dominant solutions obtained by the Feasibility First method are far more than those obtained by the Constraint Violation method, and the running time of the Constraint Violation as Objective method is shorter under different iterations.

G. SENSITIVE ANALYSIS

A sensitivity analysis is conducted to evaluate the proposed model's robustness to pesticide quantity variations, a critical practical factor. In real agricultural settings, pesticides are often purchased in fixed units (e.g., whole bottles or bags),

TABLE 8. Comparison of f_3^* with different solvers.

CASE ID	CBC				GUROBI				CPLEX			
	f_1	f_2	f_3^*	$t(s)$	f_1	f_2	f_3^*	$t(s)$	f_1	f_2	f_3^*	$t(s)$
1	1008.0	1.15	3	**0.07 ± 0.08**	1291.4	1.69	3	**0.04 ± 0.01**	1008.0	1.15	3	**0.05 ± 0.00**
2	11522.7	0.93	3	**0.08 ± 0.09**	2466.3	2.16	3	**0.06 ± 0.06**	11522.7	0.93	3	**0.06 ± 0.01**
3	829.4	2.46	4	**0.14 ± 0.11**	1842.6	1.50	4	**0.04 ± 0.00**	922.8	1.97	4	**0.08 ± 0.08**
4	39900.0	2.11	4	**0.05 ± 0.00**	4893.3	1.21	4	**0.04 ± 0.00**	39900.0	2.11	4	**0.05 ± 0.00**
5	2985.6	1.74	4	**0.05 ± 0.00**	17053.7	2.47	4	**0.04 ± 0.00**	2985.6	1.74	4	**0.05 ± 0.00**
6	4404.4	2.83	4	**0.05 ± 0.01**	5613.9	1.54	4	**0.04 ± 0.01**	4404.4	2.83	4	**0.06 ± 0.00**
7	1206.9	1.92	4	**0.05 ± 0.00**	787.8	1.35	4	**0.04 ± 0.00**	1206.9	1.92	4	**0.06 ± 0.00**
8	8645.0	2.02	5	**0.05 ± 0.00**	4029.6	2.98	5	**0.04 ± 0.00**	8645.0	2.02	5	**0.06 ± 0.00**
9	3019.3	2.30	5	**0.05 ± 0.01**	5199.4	2.86	5	**0.04 ± 0.00**	3019.3	2.30	5	**0.06 ± 0.00**
10	17130.4	2.04	5	**0.08 ± 0.03**	2468.9	2.40	5	**0.04 ± 0.01**	17130.4	2.04	5	**0.07 ± 0.01**
11	10911.2	4.36	9	**0.16 ± 0.11**	9618.4	4.64	9	**0.17 ± 0.26**	20420.4	4.64	9	**0.16 ± 0.03**
12	17134.7	2.94	10	**0.18 ± 0.04**	30171.5	6.55	10	**0.09 ± 0.01**	17134.7	2.94	10	**0.20 ± 0.01**
13	5547.3	12.48	20	**1.30 ± 1.08**	13704.5	8.74	20	**0.51 ± 0.21**	13823.7	11.82	20	**1.69 ± 1.52**
14	109657.2	16.96	30	**5.88 ± 0.18**	134938.0	16.33	30	**1.40 ± 0.12**	84955.4	16.31	30	**4.06 ± 0.80**
15	145923.3	16.76	40	**36.99 ± 0.39**	85275.7	20.70	40	**3.22 ± 0.11**	179463.3	18.26	40	**8.80 ± 0.47**
16	171761.8	24.55	50	**20.08 ± 0.35**	117302.0	27.67	50	**6.20 ± 0.20**	209360.6	23.87	50	**16.72 ± 0.27**
17	248854.5	23.23	51	**21.72 ± 0.47**	132595.6	24.72	51	**6.57 ± 0.11**	280224.0	23.21	51	**18.17 ± 0.77**
18	21384.1	26.16	54	**27.16 ± 0.97**	13405.3	28.10	54	**7.85 ± 0.26**	17703.2	27.02	54	**21.01 ± 1.41**
19	246961.6	29.29	59	**37.59 ± 1.13**	160486.7	28.63	59	**10.12 ± 0.30**	260920.2	27.35	59	**27.64 ± 1.35**
20	198371.6	36.18	65	**53.92 ± 0.90**	132169.9	33.49	65	**13.57 ± 0.32**	180552.4	35.87	65	**35.92 ± 0.60**
21	45723.5	30.12	70	**73.99 ± 1.96**	144139.0	38.61	70	**17.04 ± 0.47**	70031.1	32.77	70	**45.22 ± 0.46**
22	54267.2	38.71	75	**101.61 ± 7.00**	155259.3	43.08	75	**21.04 ± 0.64**	81349.3	39.61	75	**55.18 ± 1.04**
23	492140.3	43.46	80	**136.36 ± 3.66**	134946.2	40.19	80	**25.92 ± 0.76**	512999.3	42.21	80	**66.61 ± 0.87**

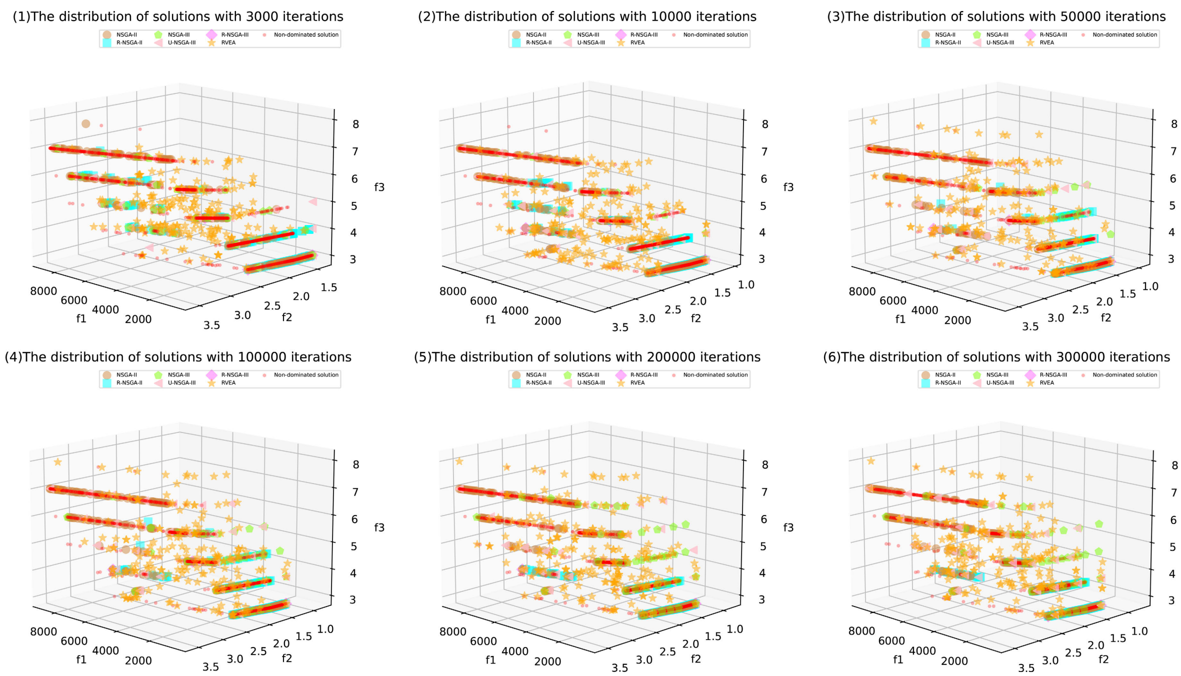


FIGURE 9. The running time of each algorithm varies with the number of iterations.

which may lead to over-purchasing and subsequent waste. We introduce two key metrics to quantify this inefficiency: Δh_i^* and Δh_i.

Δh_i^* Total land area that can be treated by the purchased quantity of pesticides for disease i. Δh_i represents the difference between the land area that can be treated by the amount of pesticides purchased and the actual land area affected by each disease.

For the convenience of calculation, we use the land area difference to replace the remaining pesticide quantity, which is equivalent. The smaller f_4 is, the smaller the difference between the treated land area and the actual diseased land area, the smaller the pesticide residue, and the better the solution quality is.

The objective function (31) is to minimize the difference in land area. Constraint (32) represents the land area that

TABLE 9. **Comparison of different algorithms.**

Iterations	Result	NSGA-II	R-NSGA-II	NSGA-III	U-NSGA-III	R-NSGA-III	SMS-EMOA	RVEA
1000	s1	37	103	13	2	27	64	0
	s2	193	173	157	150	138	200	175
	s1/s2	19.17%	59.54%	8.28%	1.33%	19.57%	32.00%	0.00%
	t(s)	**72.01 ± 0.98**	**78.54 ± 2.57**	**71.95 ± 0.09**	**72.39 ± 0.16**	**72.48 ± 0.13**	**339.35 ± 0.13**	**68.25 ± 0.76**
2000	s1	35	97	25	23	35	94	0
	s2	191	187	160	155	143	200	199
	s1/s2	18.32%	51.87%	15.63%	14.84%	24.48%	47.00%	0.00%
	t(s)	**144.81 ± 1.66**	**158.30 ± 4.44**	**144.97 ± 0.18**	**145.33 ± 0.16**	**146.90 ± 0.21**	**731.62 ± 0.22**	**138.07 ± 1.93**
3000	s1	57	107	32	20	43	123	2
	s2	191	188	155	152	143	200	196
	s1/s2	29.84%	56.91%	20.65%	13.16%	30.07%	61.50%	1.02%
	t(s)	**218.77 ± 1.88**	**238.73 ± 6.03**	**218.94 ± 0.21**	**219.48 ± 0.19**	**222.48 ± 0.27**	**1191.94 ± 0.26**	**210.07 ± 3.20**
4000	s1	56	116	37	30	36	136	12
	s2	189	189	157	154	139	200	183
	s1/s2	29.63%	61.38%	23.57%	19.48%	25.90%	68.00%	6.56%
	t(s)	**293.31 ± 1.69**	**319.78 ± 7.16**	**293.75 ± 0.43**	**294.66 ± 0.59**	**298.89 ± 0.31**	**1565.34 ± 0.68**	**282.87 ± 4.25**
5000	s1	72	124	37	27	39	137	15
	s2	186	198	158	156	141	200	180
	s1/s2	38.71%	62.63%	23.42%	17.31%	27.66%	68.50%	8.33%
	t(s)	**368.57 ± 2.03**	**401.50 ± 9.07**	**369.22 ± 0.72**	**370.88 ± 1.20**	**375.00 ± 0.50**	**1967.86 ± 2.76**	**354.94 ± 5.18**
6000	s1	64	122	41	31	47	146	21
	s2	189	197	157	155	136	200	192
	s1/s2	33.86%	61.93%	26.11%	20.00%	34.56%	73.00%	10.94%
	t(s)	**444.12 ± 2.42**	**483.45 ± 10.74**	**444.62 ± 0.85**	**447.00 ± 1.44**	**452.08 ± 0.59**	**2415.52 ± 4.32**	**429.04 ± 6.41**
7000	s1	64	131	36	33	47	149	15
	s2	184	199	157	153	134	200	188
	s1/s2	34.78%	65.83%	22.93%	21.57%	35.07%	74.50%	7.98%
	t(s)	**520.90 ± 2.45**	**567.64 ± 10.98**	**521.21 ± 1.06**	**523.84 ± 1.45**	**530.34 ± 0.90**	**2839.34 ± 5.56**	**504.74 ± 7.66**
8000	s1	76	140	31	41	50	153	16
	s2	193	199	153	152	138	200	189
	s1/s2	39.38%	70.35%	20.26%	26.97%	36.23%	76.50%	8.47%
	t(s)	**598.88 ± 2.53**	**650.68 ± 13.47**	**599.59 ± 1.61**	**602.36 ± 1.64**	**609.74 ± 1.32**	**3255.93 ± 6.35**	**583.30 ± 13.45**
9000	s1	88	128	41	34	47	168	13
	s2	188	197	150	151	134	200	178
	s1/s2	46.81%	64.97%	27.33%	22.52%	35.07%	84.00%	7.30%
	t(s)	**675.45 ± 2.54**	**743.99 ± 22.16**	**680.60 ± 4.65**	**679.51 ± 2.30**	**690.01 ± 1.62**	**3660.71 ± 4.52**	**682.34 ± 32.65**
10000	s1	91	136	38	35	53	167	13
	s2	190	195	152	146	146	200	188
	s1/s2	47.89%	69.74%	25.00%	23.97%	36.30%	83.50%	6.91%
	t(s)	**764.15 ± 18.76**	**874.44 ± 53.11**	**757.72 ± 4.30**	**760.71 ± 1.81**	**768.39 ± 1.77**	**4099.17 ± 7.53**	**756.77 ± 33.50**

* s_1 represents the number of non-dominant solutions obtained by this algorithm;
* s_2 represents the number of all solutions obtained by this algorithm.

can be treated by the amount of pesticides purchased for each disease. Constraint (33) represents the difference between the land area that can be treated by the amount of pesticides purchased and the actual land area affected by each disease.

$$min \quad f_4 = \sum_{i \in N} \Delta h_i \tag{31}$$

$$\Delta h_i^* = \sum_{j \in M} y_{ij} g_{ij} \tag{32}$$

$$\Delta h_i = \Delta h_i^* - h_i \tag{33}$$

Table 11 shows the randomly selected 10 groups of non-dominant solutions and the value of the objective function f_4. The analysis reveals a strong positive correlation between f_4 and f_1, and a negative correlation between f_4 and f_2. Under this criterion, the non-dominant solution with a smaller f_1 value has less pesticide residue, and the solution performs better.

VI. A CASE STUDY

In this section, a case study is presented by using a field from Xingtai City, Hebei Province, China. Figure 13 shows that the planting area of potatoes is 82,800 square meters. The planting area of cabbage is 61,700 square meters, and the planting area of wheat is 189,600 square meters.

Among them, wheat suffers from powdery mildew, and the pesticides that can be used for treatment are tebuconazole and triadimefon with a concentration of 25%. Potato suffers from bacterial wilt, and the pesticides that can be used for treatment are chloroisobromine-cyanuric acid and ethylicin with a concentration of 80%. There are Pieris Rapae Linne on Chinese cabbage, and the pesticides that can be used for treatment are Lambda-cyhalothrin and avermectin with a concentration of 1.8%. Table 12 shows the specific data. It should be noted that the unit of g is *Mu*, which is a land unit in China, and 1 *Mu* is about 666.67 square meters.

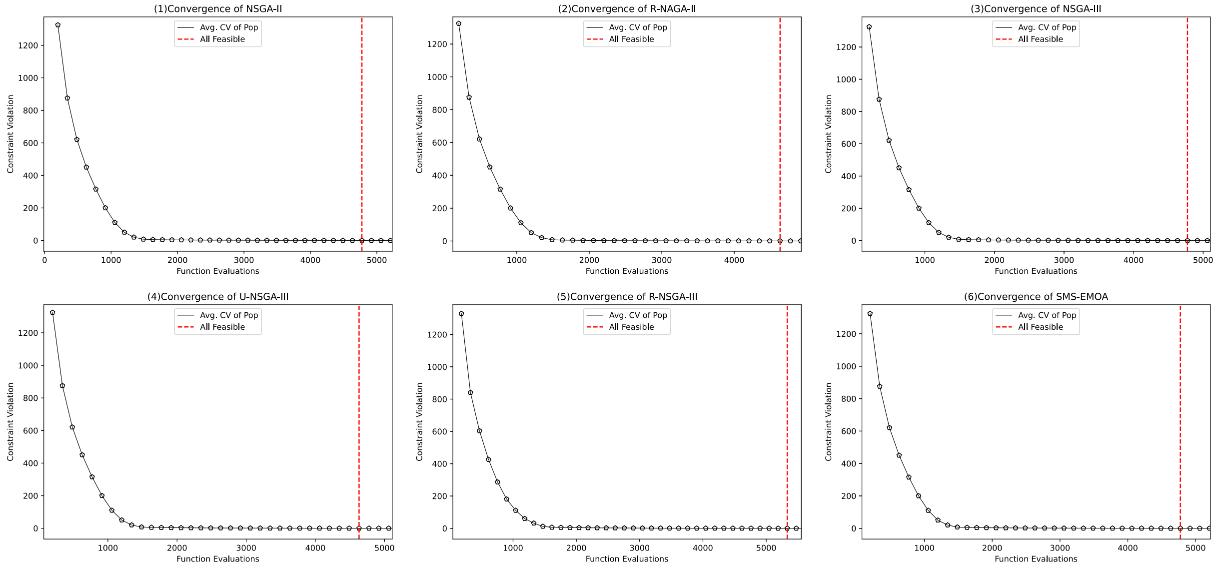


FIGURE 10. Convergence of each algorithm.

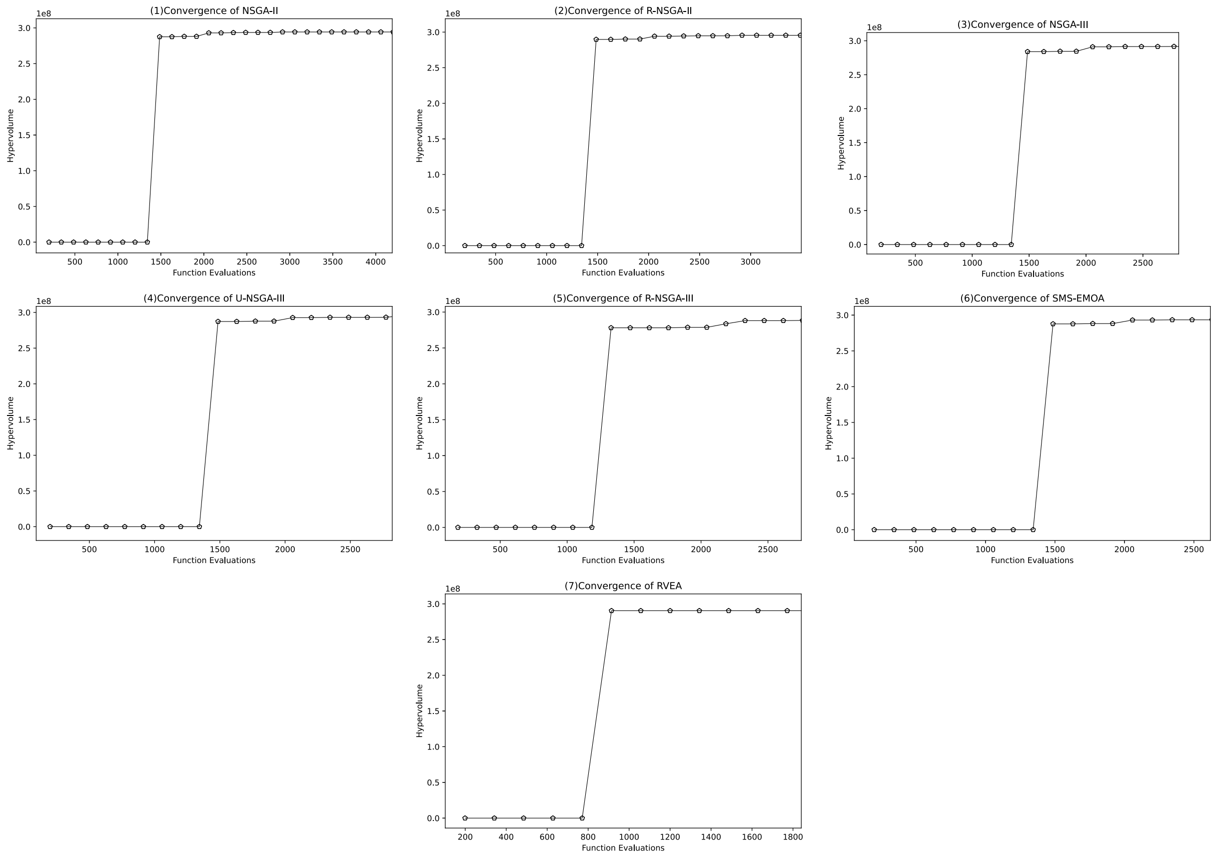


FIGURE 11. Hypervolume of each algorithm.

For this problem, we use the PYMOO library of PYTHON to solve it, in which the number of iterations is 300,000, and 200 solutions are obtained. We selected four of the 200 solutions for analysis. Table 13 shows the specific values of the solutions. Among them, solution 1 is the solution with the smallest f_1 among the 200 solutions. Solution 2 has the largest f_2, and solution 3 has the smallest f_3. Multi-Criteria Decision Making selects solution 4.

TABLE 10. Comparison of different constraint handling with different iterations.

Iterations	Result	Feasibility first	Constraint violation
3000	s1	45	4
	s2	200	54
	s1/s2	22.5%	7.4%
	t(s)	376.35	359.17
10000	s1	67	5
	s2	200	53
	s1/s2	33.5%	9.4%
	t(s)	1203.38	1210.87
50000	s1	91	3
	s2	200	49
	s1/s2	45.5%	6.1%
	t(s)	6008.30	6001.61
100000	s1	97	5
	s2	200	50
	s1/s2	48.5%	10.0%
	t(s)	12325.41	11938.05
200000	s1	97	7
	s2	200	57
	s1/s2	48.5%	12.3%
	t(s)	24446.79	23895.58
300000	s1	108	7
	s2	200	48
	s1/s2	54.0%	14.6%
	t(s)	36206.23	35792.67

* s_1 represents the number of non-dominant solutions obtained by this algorithm;

* s_2 represents the number of all solutions obtained by this algorithm.

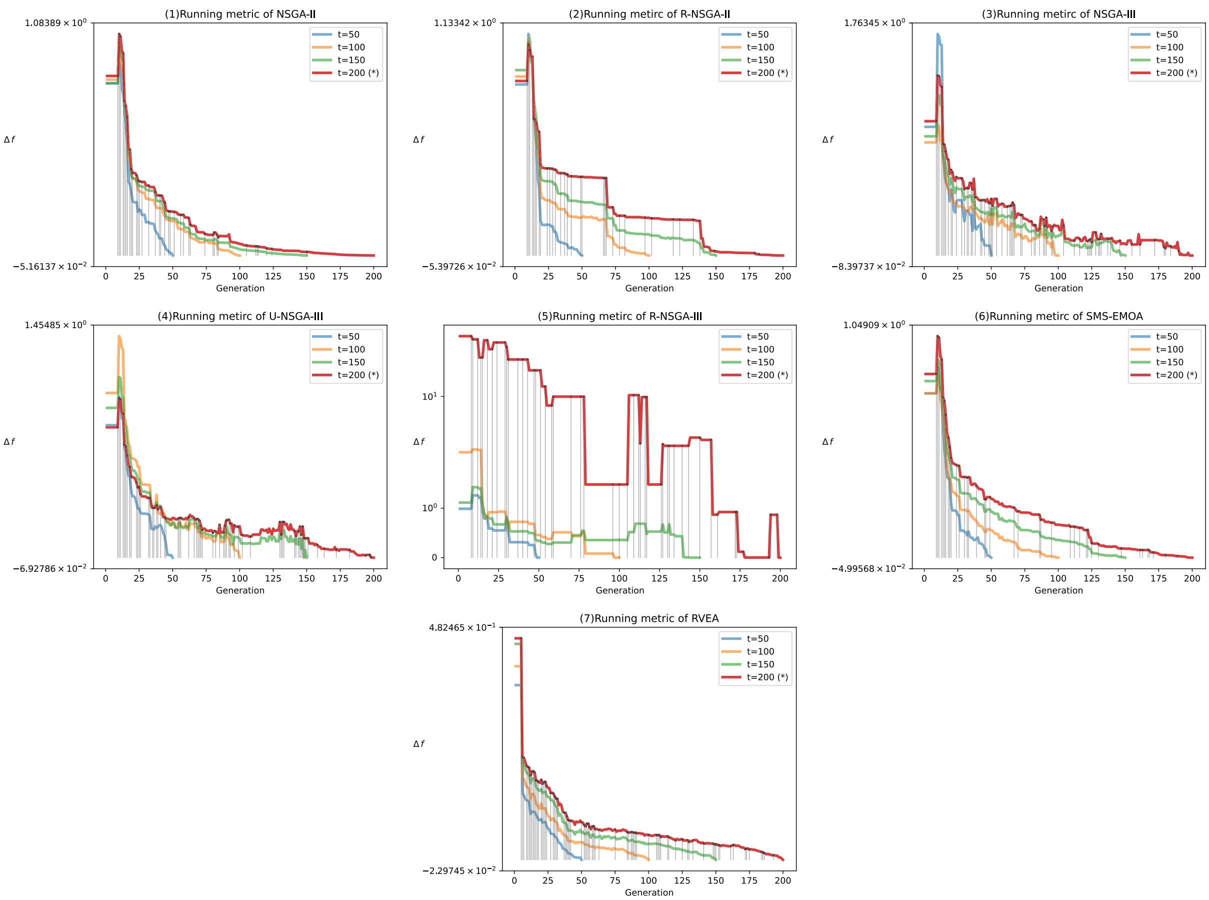


FIGURE 12. The running metric of each algorithm.

TABLE 11. Randomly selected 10 groups of non-dominant solutions.

Solution ID	f_1	f_2	f_3	f_4	The value of the variable y			
1	497.5	1.41	3	8	0	0	0	77
					0	0	0	50
					53	0	0	0
2	799.1	2.06	3	519	0	67	39	0
					0	0	0	50
					53	0	0	0
3	858.1	2.14	3	649	0	67	49	0
					0	0	0	50
					53	0	0	0
4	986.5	2.33	4	961	0	67	69	4
					0	0	0	50
					53	0	0	0
5	1038.9	2.41	4	1091	0	67	77	6
					0	0	0	50
					53	0	0	0
6	1153.3	2.57	4	1663	0	67	77	50
					0	0	0	50
					53	0	0	0
7	1750.5	2.79	5	2524	0	67	77	77
					170	0	0	50
					53	0	0	0
8	2521.3	2.97	6	3400	0	67	77	77
					334	0	0	50
					53	64	0	0
9	3309.4	3.08	7	4080	0	67	77	77
					334	0	0	50
					53	167	62	0
10	4589.7	3.19	8	4297	0	67	77	77
					334	0	0	50
					53	167	279	0

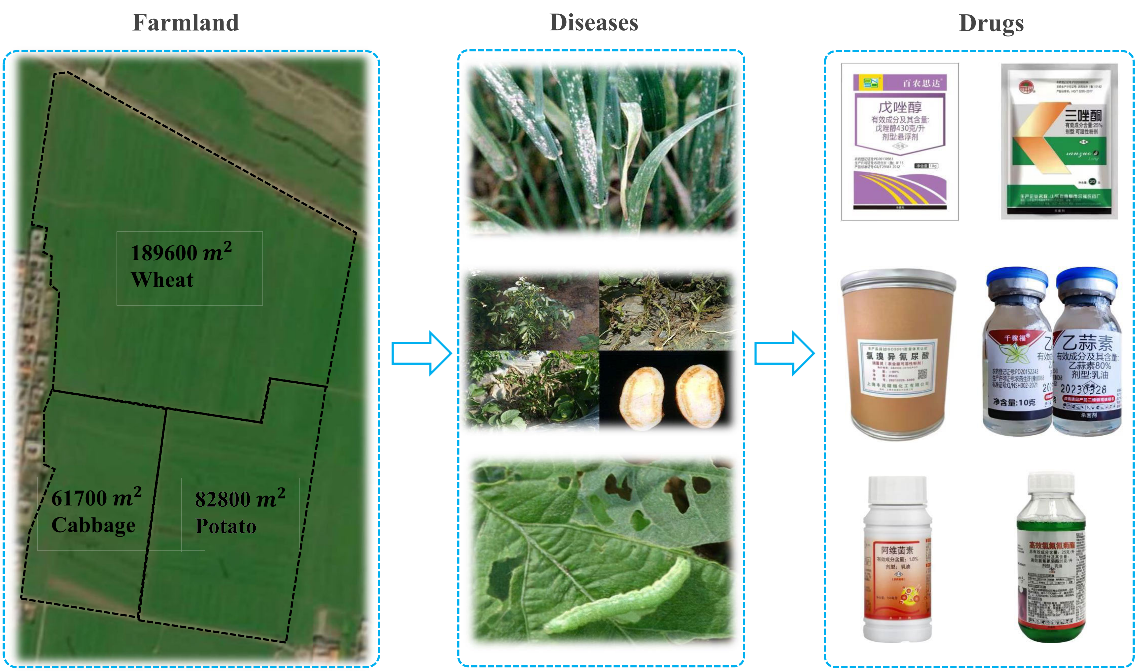


FIGURE 13. A farmland in Xingtai City, Hebei province, china.

For solution 1, we can find that when we strive to reduce the value of f_1, we will reduce the number of bottles of pesticides purchased as much as possible. The value of the variable y means we bought 566 bags of medicine tiadimefon, 83 bottles of medicine chloroisobrominc-cyanuric acid, and 37 bottles of medicine avermectin. And the total cost is 4379.5 Yuan,

TABLE 12. Parameter setting of the case study.

Crop	Disease	Pesticide	p	g	w
wheat	powdery mildew	tiadimefon	3.00	0.50	0.62
		tebuconazole	2.50	0.25	0.77
potato	bacterial wilt	ethylicin	12.00	0.50	0.83
		chloroisobromine-cyanuric-acid	29.90	1.50	0.51
Chinese cabbage	PierisrapaeLinne	Lambda-cyhalothrin	8.09	1.00	0.50
		avermectin	5.40	2.50	0.50

TABLE 13. Specific values of four groups of solutions.

Solution ID	f_1	f_2	f_3	The value of the variable y					
1	4379.5	1.64	3	566	0	0	0	0	0
				0	0	0	83	0	0
				0	0	0	0	0	37
2	10937.9	3.74	6	566	1132	0	0	0	0
				0	0	248	83	0	0
				0	0	0	0	93	37
3	5426.4	1.96	3	566	0	0	0	0	0
				0	0	248	0	0	0
				0	0	0	0	93	0
4	7113.7	2.86	5	566	595	0	0	0	0
				0	0	248	0	0	0
				0	0	0	0	93	37

the utility is 1.64, and it needs to be sprayed 3 times. For solution 2, when trying to increase the value of f_2, we will buy all kinds of pesticides as much as possible. For solution 3, when efforts are made to reduce f_3, only one pesticide will be purchased to treat each disease.

VII. CONCLUSION

Agriculture is essential for the nature-society system to sustain the food chains of natural ecosystems. Humans have a long history of crop cultivation. In modern agriculture, the application of pesticides has further enhanced crop yields. However, farmers' awareness of proper pesticide use and the risks of overuse remains limited. Safe and scientific pesticide application is therefore essential for ensuring safe spraying practices.

This paper presents a comparative analysis of the performance of linear and nonlinear models, various solvers, different algorithms, and distinct constraint-handling methods targeting the pesticide matching problem. Regarding the comparison between linear and nonlinear models, it is evident that nonlinear models exhibit consistently longer solution times than linear models across all problem scales. This highlights the necessity of linearizing nonlinear models—a modification that can significantly shorten solution time.Regarding solver comparison, the GUROBI solver exhibits shorter solution times across all problem scales compared to the CPLEX and CBC solvers, with all three solvers achieving equivalent solution quality. When evaluating algorithms against four key metrics—solution quantity, solution quality, solution time, and convergence performance—the R-NSGA-II algorithm outperforms the other compared algorithms.

Concerning the comparison of different constraint-handling methods, the Feasible First method exhibits a shorter runtime when optimizing f_1, whereas the Constraint Violation method shows a shorter runtime when optimizing f_2 and f_3. Both methods achieve comparable solution quality. At the same time, the experimental results show that the mathematical model established in this paper is reasonable and effective, which provides meaningful guidance for pesticide spraying practice.

REFERENCES

[1] F. H. M. Tang, M. Lenzen, A. McBratney, and F. Maggi, "Risk of pesticide pollution at the global scale," *Nature Geosci.*, vol. 14, no. 4, pp. 206–210, Apr. 2021.

[2] *Using Less Chemical Pesticides: European Commission Publishes Toolbox of Good Practices*, European commission, Brussels, Feb. 2023. [Online]. Available: https://agriculture.ec.europa.eu/media/news/using-less-chemical-pesticides-european-commission-publishes-toolbox-good-practices-2023-02-28_en

[3] J. Vuaille, M. Holbak, M. Perslev, E. Diamantopoulos, S. M. Jensen, M. E. Styczen, C. T. Petersen, B. W. Strobel, and P. Abrahamsen, "Can information on past and near-future weather and field conditions predict the safest pesticide application day?" *Comput. Electron. Agricult.*, vol. 203, Dec. 2022, Art. no. 107454.

[4] K. Sellamuthu and V. Kumar Kaliappan, "Q-learning-based pesticide contamination prediction in vegetables and fruits," *Comput. Syst. Sci. Eng.*, vol. 45, no. 1, pp. 715–736, 2023.

[5] B. Sachan, S. D. Kandpal, A. K. Singh, A. Kaushik, S. Jauhari, and A. Ansari, "Agricultural pesticide use and misuse: A study to assess the cognizance and practices among north Indian farmers," *J. Family Med. Primary Care*, vol. 11, no. 10, pp. 6310–6314, 2022.

[6] A. R. da Cruz, R. T. N. Cardoso, E. F. Wanner, and R. H. C. Takahashi, "A multiobjective non-linear dynamic programming approach for optimal biological control in soy farming via NSGA-II," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 3093–3099.

[7] S. A. Mir, B. N. Dar, M. M. Mir, S. A. Sofi, M. A. Shah, T. Sidiq, K. V. Sunooj, A. M. Hamdani, and A. Mousavi Khaneghah, "Current strategies for the reduction of pesticide residues in food products," *J. Food Composition Anal.*, vol. 106, Mar. 2022, Art. no. 104274.

[8] M. Zhao, Z. Chen, L. Hao, H. Chen, X. Zhou, and H. Zhou, "CMC based microcapsules for smart delivery of pesticides with reduced risks to the environment," *Carbohydrate Polym.*, vol. 300, Jan. 2023, Art. no. 120260.
[9] I. Mubeen, M. Fawzi Bani Mfarrej, Z. Razaq, S. Iqbal, S. A. H. Naqvi, F. Hakim, W. F. A. Mosa, M. Moustafa, Y. Fang, and B. Li, "Nanopesticides in comparison with agrochemicals: Outlook and future prospects for sustainable agriculture," *Plant Physiol. Biochemistry*, vol. 198, May 2023, Art. no. 107670.
[10] A. Aulagnier, "Substitution policies as a categorization process. Making biological alternatives a solution for the French national pesticide reduction plan," *Environ. Sci. Policy*, vol. 146, pp. 37–46, Aug. 2023.
[11] R. T. Marler and J. S. Arora, "Function-transformation methods for multi-objective optimization," *Eng. Optim.*, vol. 37, no. 6, pp. 551–570, Sep. 2005.
[12] I. Y. Kim and O. de Weck, "Adaptive weighted sum method for multiobjective optimization," in *Proc. 10th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf.*, Aug. 2004, p. 4322.
[13] V. Pandiyaraju, S. Ganapathy, N. Mohith, and A. Kannan, "An optimal energy utilization model for precision agriculture in WSNs using multi-objective clustering and deep learning," *J. King Saud Univ.- Comput. Inf. Sci.*, vol. 35, no. 10, Dec. 2023, Art. no. 101803.
[14] N. Taherzadeh-Shalmaei, M. Rafiee, A. Kaab, M. Khanali, M. A. Vaziri Rad, and A. Kasaeian, "Energy audit and management of environmental GHG emissions based on multi-objective genetic algorithm and data envelopment analysis: An agriculture case," *Energy Rep.*, vol. 10, pp. 1507–1520, Nov. 2023.
[15] L.-L. Dai, Q.-K. Pan, Z.-H. Miao, P. N. Suganthan, and K.-Z. Gao, "Multi-objective multi-picking-robot task allocation: Mathematical model and discrete artificial bee colony algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 6, pp. 6061–6073, Jun. 2024.
[16] C. H. Geissler, N. L. Haan, B. Basso, A. Fowler, D. A. Landis, T. J. Lark, and C. T. Maravelias, "A multi-objective optimization model for cropland design considering profit, biodiversity, and ecosystem services," *Ecol. Model.*, vol. 500, Jan. 2025, Art. no. 110954.
[17] D. Haloui, K. Oufaska, M. Oudani, K. E. Yassini, A. Belhadi, and S. Kamble, "Sustainable urban farming using a two-phase multi-objective and multi-criteria decision-making approach," *Int. Trans. Oper. Res.*, vol. 32, no. 2, pp. 769–801, Mar. 2025.
[18] L. Zhu, Z. Li, and S. Hu, "Efficient data-centric pest images identification method based on Mahalanobis entropy for intelligent agriculture," *J. Electron. Imag.*, vol. 32, no. 5, p. 052406, 2023.
[19] M. Akkas Ali, R. Kumar Dhanaraj, and S. Kadry, "AI-enabled IoT-based pest prevention and controlling system using sound analytics in large agricultural field," *Comput. Electron. Agricult.*, vol. 220, May 2024, Art. no. 108844.
[20] O. Appiah, K. O. Hackman, B. A. A. Diallo, K. O. Ogunjobi, S. Diakalia, O. Valentin, D. Abdoul-Karim, and G. Dabire, "PlanteSaine: An artificial intelligent empowered mobile application for pests and disease management for maize, tomato, and onion farmers in burkina faso," *Agriculture*, vol. 14, no. 8, p. 1252, Jul. 2024.
[21] B. Kariyanna and M. Sowjanya, "Unravelling the use of artificial intelligence in management of insect pests," *Smart Agricult. Technol.*, vol. 8, Aug. 2024, Art. no. 100517.
[22] Z. Qin, W. Wang, K.-H. Dammer, L. Guo, and Z. Cao, "Ag-YOLO: A real-time low-cost detector for precise spraying with case study of palms," *Frontiers Plant Sci.*, vol. 12, Dec. 2021, Art. no. 753603.
[23] C. Gu, T. Cheng, N. Cai, W. Li, G. Zhang, X.-G. Zhou, and D. Zhang, "Assessing narrow Brown leaf spot severity and fungicide efficacy in Rice using low altitude UAV imaging," *Ecol. Informat.*, vol. 77, Nov. 2023, Art. no. 102208.
[24] X. Quan, Q. Guo, J. Ma, and R. Doluschitz, "The economic effects of unmanned aerial vehicles in pesticide application: Evidence from Chinese grain farmers," *Precis. Agricult.*, vol. 24, no. 5, pp. 1965–1981, Oct. 2023.
[25] S. Prakash Kumar, D. Jat, R. K. Sahni, B. Jyoti, M. Kumar, A. Subeesh, B. S. Parmar, and C. R. Mehta, "Measurement of droplets characteristics of UAV based spraying system using imaging techniques and prediction by GWO-ANN model," *Measurement*, vol. 234, Jul. 2024, Art. no. 114759.
[26] R. Yu, Y. Luo, and L. Ren, "Detection of pine wood nematode infestation using hyperspectral drone images," *Ecol. Indicators*, vol. 162, May 2024, Art. no. 112034.
[27] Y. Zhou and G. M. Lee, "A bi-objective medical relief shelter location problem considering coverage ratios," *Int. J. Ind. Eng., Theory, Appl. Pract.*, vol. 27, no. 6, pp. 971–988, 2020.
[28] Z. Zhong, H. Jin, Y. Sun, and Y. Zhou, "Two incentive policies for green shore power system considering multiple objectives," *Comput. Ind. Eng.*, vol. 194, Aug. 2024, Art. no. 110338.
[29] V. Huilcapi, C. Castillo, D. Sanchez, and R. Cajo, "Optimization of an air pressure system: A multi-objective control and modeling approach," *IEEE Access*, vol. 12, pp. 96691–96703, 2024.
[30] N. Khodadadi, L. Abualigah, E.-S.-M. El-Kenawy, V. Snasel, and S. Mirjalili, "An archive-based multi-objective arithmetic optimization algorithm for solving industrial engineering problems," *IEEE Access*, vol. 10, pp. 106673–106698, 2022.

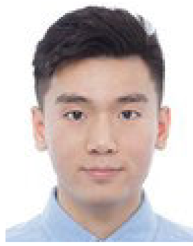
YONGSHENG WANG received the M.S. degree in computer science and technology from Guizhou University, Guizhou, China, in 2018. He is currently an Engineer with China Zhengzhou Tobacco Research Institute, China National Tobacco Corporation. His current research interests include deep learning, data mining, and knowledge graph.

CONG JIN received the bachelor's degree in logistics management from Zhengzhou University, in 2024. She is currently pursuing the master's degree with the Ocean College, Zhejiang University. Her research direction is AGV dispatching.

YANJIE ZHOU (Member, IEEE) received the B.S. and M.S. degrees in computer science and computer application technology from Zhengzhou University, in 2012 and 2015, respectively, and the Ph.D. degree in industrial engineering from Pusan National University, in 2020. He is currently an Associate Professor with the School of Management, Zhengzhou University. His research focuses on addressing real-world optimization problems through the application of artificial intelligence techniques. Additionally, he holds the position of an Associate Editor for *Engineering Optimization*.

WEIHUA FENG received the M.S. degree in computer science and technology from Zhengzhou University, Zhengzhou, China. He is currently a Professor of engineering with China Zhengzhou Tobacco Research Institute, China National Tobacco Corporation. His current research interests include big data, data mining, and data science.

GUOHAO ZONG received the M.S. degree in data science from Monash University, Melbourne, Australia, in 2018. He is currently an Engineer with China Zhengzhou Tobacco Research Institute, China National Tobacco Corporation. His current research interests include machine learning, computer vision, and multi-modal learning.



DI WANG received the M.S. degree in software engineering from the College of Information and Computer Engineering, Northeast Forestry University, Harbin, China, in 2017. He is currently a Researcher with China Zhengzhou Tobacco Research Institute, CNTC, China. His current research interests include deep learning, chemometrics, and spectral analysis.

RUI WANG received the M.S. degree in computer science and technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2017. She is currently a Senior Engineer with China Zhengzhou Tobacco Research Institute, China National Tobacco Corporation. Her current research interests include deep learning, data mining, and natural language processing.