Expert Systems With Applications 270 (2025) 126369

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Multi-dimensional assignment model and its algorithm for multi-features decision-making problems

Tian Xie [a,c,*], Yong-jian Huang [b,*], Wei-fan Chen [c]

[a] School of Economics, Management and Law at the University of South China, Tian Xie, No. 28, West Changsheng Road, Hengyang City, Hunan Province, PR China
[b] School of Management at the Shanghai University, Yong-jian Huang, Shanghai University, No. 99, ShangDa Road, Shanghai 200444, PR China
[c] Information Sciences and Technology at The Pennsylvania State University, Wei-fan Chen, 44 University Drive, Dallas, PA 18612, United States

A R T I C L E I N F O

Keywords:
Multi-features
Multi-dimensional assignment model
Optimization problems
VMA

A B S T R A C T

This paper proposes the multi-dimensional assignment model to address complex decision problems involving multi-agent features, multi-stages, resources, constraints, and more. Compared to traditional two-dimensional (2D) assignment models, this model can effectively describe assignment problems ranging from 2D to N-dimensional with multi-feature constraints. To tackle this model, we develop a dimensionality-reducible “Virtual Matching Algorithm” (VMA) based on ideas from Roulette-wheel selection, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Taking organizational assessment and guavas production as real cases, the corresponding model is constructed and the performance of VMA is tested and optimized. In contrast to PSO, Binary PSO (Bi-PSO), GA, and hybrid GA-PSO, VMA demonstrates improved solving speed and the ability to find better solutions. These results exhibit the universality and effectiveness of our proposed model and algorithm.

1. Introduction

In practical management activities, managers frequently face complex decision problems with multi-features such as multi-agents, multi-stages, multi-resources, multi-constraints, multi-objectives. For example, in the context of organizational assessments characterized by one-to-one multi-stages, different employers (executive agents) must be appointed to conduct assessments for employees (task agents), while adhering to specific time-related constraints at each stage. The decision-making problem has a three-dimensional (3D) assignment relationship involving “Executive agents – Task agents – Stages” (Fig. 1 – ①). For another example, in the process of packaging guavas, the decision maker needs to consider the varying ripeness and fresh-keeping capacities of guavas (executive agents) in different positions (resources) in the boxes (task agents). The decision problem has a 3D assignment relationship involving “Executive agents – Task agents – Resources” (Fig. 1 – ②). Four-dimensional or higher-dimensional decision problems can be realized by adding new dimensional features into the original 3D decision problem space, as shown in Fig. 2. These above decision problems differ from the traditional 2D assignment problems in that they can be considered as novel N-dimensional assignment problems with N-dimensional features.

The traditional 2D assignment problem only considers the 2D characteristics of multi-agent and multi-constraint task agents and executive agents. Multiple agents generally refer to task agents (T) and executive agents (E), which often have specific relationships or constraints R_{TE} between them. The costs or benefits (performances, P_{TE}) of assignments by different task agents and executive agents together form a performance matrix for the objective of minimizing cost or maximizing benefit. Each value in the matrix represents a combination. The optimal solutions for the 2D assignment problems can be found among various agents in the matrix. The traditional 2D assignment problem is illustrated in Fig. 1 – ①.

The 3D assignment model proposed in this paper introduces a new dimension that can be expressed as “time or stages”, “positions or resources” or other features, based on the traditional 2D assignment model. This new dimension creates a three-dimensional decision space that incorporates executive agents and task agents, as seen in the “multi-stages organizational assessment problem” decision space (Fig. 1-②) and “multi-resources guavas production problem” (Fig. 1-③). Each coordinate value $P_{TES}(x_1, x_2, x_3)$ in the decision space represents the performance of the executive agents (E) and the task agents (T) as they

* Corresponding authors at: School of Economics, Management and Law at the University of South China, Tian Xie, No. 28, West Changsheng Road, Hengyang City, Hunan Province, PR China (T. Xie). School of Management at the Shanghai University, Yong-jian Huang, Shanghai University, No. 99, ShangDa Road, Shanghai 200444, PR China (Y.-j. Huang).

E-mail addresses: 2014001099@usc.edu.cn (T. Xie), hongwu@shu.edu.cn (Y.-j. Huang), weifan@psu.edu (W.-f. Chen).

https://doi.org/10.1016/j.eswa.2024.126369
Received 27 March 2024; Received in revised form 27 December 2024; Accepted 30 December 2024
Available online 10 January 2025

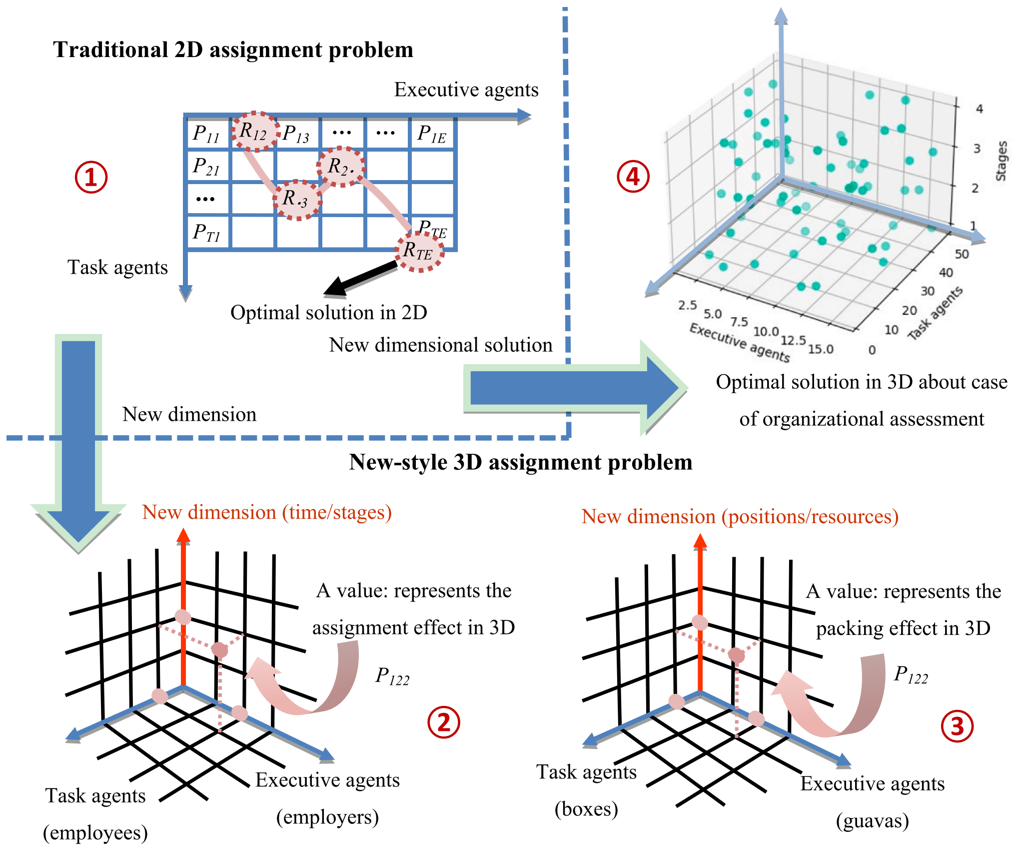

Fig. 1. 2D and 3D assignment model.

complete the assessment together, considering the influence of new dimensional feature (S). For example, in the multi-stages organizational assessment problem, the performance of the previous stage should be taken into account in the subsequent decision-making process. As a result, its constraints become increasingly complex. With the increase in dimensions and constraints, multiple local best solutions emerge in the solving processes (Fig. 1-④).

Similarly, the 3D assignment model can be extended to four, five, or even N-dimensions, depending on the complexity and dimension characteristics of the assignment problem itself. Four-dimensional decision-making problems can introduce a new dimensional feature to the original three-dimensional decision space. For example, in the multi-resources guavas production problem mentioned above (Fig. 1-③), "Stages (seasons)" would be introduced into the original three-dimensional assignment relationship of "Executive agents – Task agents –Positions (resources)", creating a four-dimensional assignment relationship of "Executive agents – Task agents – Resources (positions) – Stages (seasons)". (Fig. 2 – ①). For another example, in the organizational assessment problem mentioned above (Fig. 1-②), the new dimensions of "employers' grade" and "employees' grade" can be introduced into the original three-dimensional assignment relationship of "Executive agents – Task agents – Stages" creating a five-dimensional assignment relationship with "Employers' competence – Employees' performance – Employers' grade – Employees' grade – Stages" (Fig. 2-②), particularly if employers with lower grades cannot assess employees with higher grades. Therefore, as the assignment problems become more complex, additional dimensions can be gradually introduced, leading to the development of common and comprehensive multi-dimensional assignment models (MAM) (Fig. 2-③).

The solution space for decision problems often exponentially expands with the increase in dimensional features, a phenomenon commonly referred to as the "curse of dimensionality" (Bellman, 1966; Keogh & Mueen, 2010). To effectively address this challenge, a specific heuristic algorithm is required. Traditional algorithms, such as PSO (Zou et al., 2021), GA-PSO (Mozafar et al., 2017; Garg, 2016; Kamel et al., 2020), and Bi-PSO (Saremi et al., 2018; Gad, 2022), are primarily designed for solving assignment problems in three or fewer dimensions. However, their performance often falls short in tackling complex multi-dimensional decision problems. Therefore, we propose the Virtual Matching Algorithm (VMA) to address the limitations of the above traditional algorithms, which are typically constrained to solving two-dimensional (2D) allocation models. The VMA is uniquely designed to solve complex MAM involving multi-agent, multi-stage, multi-resource, and multi-constraint decisions. This innovative algorithm introduces dimensionality reduction and virtual matching probabilities, enhancing convergence and solution accuracy in MAM. On the one hand, by the proposed dimensionality reduction method, the VMA can cope with the exponential complexity of high-dimensional spaces, navigating solution spaces with enhanced flexibility and adaptability. On the other hand, using virtual matching sequences, the VMA creates a structured framework that explores feasible domains through a series of virtual matches between dimensions, enabling precise and efficient identification of optimal solutions.

The structure of the paper is as follows: Section 2 reviews relevant literature on multi-dimensional models and optimization algorithms. Section 3 outlines the mathematical model of multi-dimensional problems. Section 4 explains the principles and optimization of the VMA. Section 5 presents two application cases, illustrating the validity of the proposed model and algorithms. Section 6 discusses the scalability and universality of the proposed model, comparing it with other dimensionality reduction methods and algorithms. Finally, Section 7 highlights the contributions and suggests future research directions.

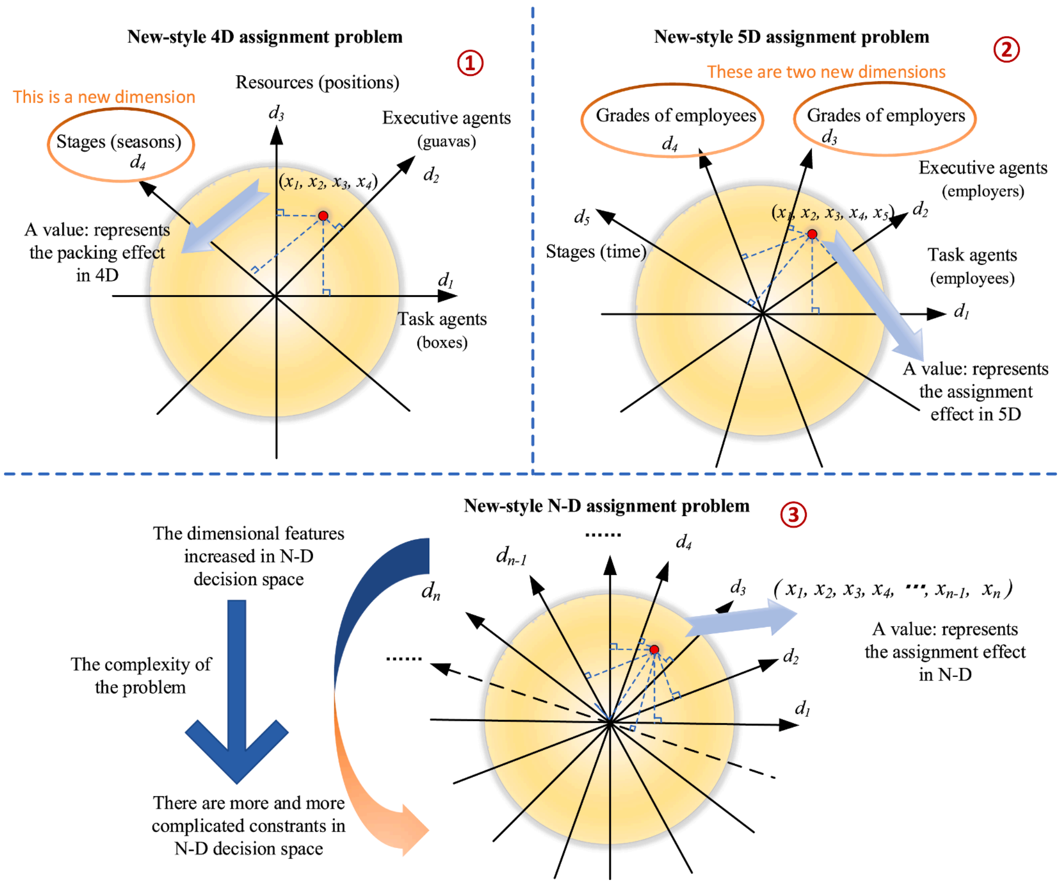

Fig. 2. N-dimensional (N-D) assignment problem model.

2. Literature review

This section explores the development situation of multi-dimensional assignment models (MAM) and their applications. It also discusses popular algorithms and enhances dimensionality reduction techniques like PCA and autoencoders to address the challenges of high-dimensional problems.

2.1. The dynamics of the multi-dimensional model

The two-dimensional assignment problem is a specific integer programming problem discussing how to assign m task executors (executive agents) to complete n tasks (task agents) for the objective of minimizing cost or maximizing benefit. Pierskalla et al. (1968) introduced a three-dimensional model and indicated that the multi-dimensional assignment problems (MAPs) represent a higher-dimensional extension of the two-dimensional assignment problem, incorporating temporal or spatial dimensions. In past years, MAPs have been a focal point of academic research. For example, Walteros et al. (2014) introduce a continuous nonlinear program for solving the multi-dimensional star assignment problem (MSAP). Ma et al. (2022) proposed a multi-dimensional model evaluate the network resilience from three aspects: absorption ability, resistance ability, and recovery ability. Following the concept of four-dimensional assignment problems, Samanta et al. (2024) formulated a multi-period, two-stage 4D transportation problem for breakable items in a neutrosophic environment.

In recent years, with the application and development of modern science and technology, many problems involving three-dimensional features have gained significant attention from scholars. These problems have found applications in varied decision-making fields, including transportation, economics, healthcare, industrial production, military and more. From another perspective, it is evident that the concept of multi-dimensional assignment has garnered the interest of scholars. Many assignment problems with multi-dimensional features can be categorized as one or more types of MAPs. Many practical decision problems involve multi-dimensional agents and variables. For example, Bastani and Bayati (2019) developed a novel K-armed contextual bandit algorithm for decision-making with high-dimensional covariates, leveraging the LASSO estimator and introducing a polylogarithmic bound on cumulative expected regret in the covariate dimension. Previous scholars have developed various 3D assignment models for practical problems. For example, Bansal et al. (2006) introduced a packing problem that differs from the one-dimensional case or two-dimensional packing problem. Sherali and Zhu (2008) introduced a two-stage fleet assignment model for airlines, optimizing aircraft assignments based on stochastic passenger demands and itinerary-based demand to maximize profits. Côté et al. (2020) addresses a stochastic vehicle routing problem with the two-dimensional size (height and width) as well as the weight., using a two-stage stochastic program solved with the integer L-shaped method, introducing new inequalities and lower bounds and reporting computational results on both specifically generated and classical instances. Chaosheng et al. (2019) developed a three-dimensional assignment model for “soldiers, equipment, and missions,” centered around a combat engineering support operations problem. Its primary dimensions encompass the operational level of soldiers, the technological state of equipment, and cooperation.

Scholars have explored dimensional features, including multi-agents (resources/tasks) or multi-stages (time windows) (MS), in 2D or 3D assignment problem models (Côté et al., 2020; Li et al., 2020; Li et al., 2021). Most existing models primarily address assignment problems tailored to specific dimensions or scenarios. There is very limited academic research that considers multiple new dimensional features together, abstracting common characteristics from similar problems, and proposing a universal Multi-dimensional Assignment Model (MAM). Therefore, this study aims to explore MAM for multi-agents (MA), multi-resources (MR), multi-tasks (MT), and multi-stages (MS), along with the study of its “Virtual Matching Algorithm” (VMA).

2.2. Review of solving algorithms

Traditional exact algorithms, such as the branch and bound algorithm and the Hungarian Method, are the mainstream algorithms for solving models of two-dimensional or three-dimensional assignment problems. Hahn et al. (2008) introduced an exact solution method for solving the quadratic three-dimensional assignment problem (Q3AP). Robenek et al. (2014) developed an exact solution algorithm based on a branch-and-price framework to tackle berth allocation and yard assignment problems. Lu et al. (2021) proposed a two-dimensional “weapon-target” linear assignment model and solved it using column enumeration and branch and bound algorithm. Karsu Ö et al (2012) proposed a mixed integer nonlinear programming model for the airport gate assignment problem and developed its beam search and filtered beam search algorithms that employ robust lower and upper bounding mechanisms.

Many scholars utilize heuristics algorithms such as Particle Swarm Optimization Algorithm (PSO), Genetic Algorithm (GA), Binary Particle Swarm Optimization (Bi-PSO), hybrid Genetic Algorithm and Particle Swarm Optimization (GA-PSO), or other heuristic algorithms to solve decision problems with 2D or 3D features within complex constraints. Saremi et al. (2018) extended the Binary PSO application to hand posture estimation, and Gad (2022) emphasized the robustness of Bi-PSO in addressing discrete optimization challenges. Zou et al. (2021) developed a discrete multi-objective PSO to tackle resource allocation during epidemics. Deng et al. (2020) proposed IPOQEA for managing airport gate congestion. In another domain, Zhang et al. (2020) introduced EGATS to address the Quadratic Assignment Problem (QAP). These studies underscore the adaptability and efficiency of PSO and hybrid algorithms in solving complex, real-world optimization problems across varied contexts. Mozafar et al. (2017) leveraged hybrid GA-PSO approaches for multi-objective optimization, such as renewable energy allocation in smart grids and constrained optimization problems. Similarly, Hybrid GA-PSO methodologies were further explored by Kamel et al. (2020) for fault-tolerant control in robotics. Furthermore, Hahn et al. (2008) also introduced four stochastic local search algorithms based on a reformulation linearization technique to solve large and difficult QAP. Ghoseiri & Ghannadpour (2010) introduced a hybrid genetic algorithm to address a multi-depot homogenous locomotive assignment problem with time windows. Karsu Ö and Azizoğlu. (2021) examined the multi-resource agent bottleneck generalized assignment problem and defined a tabu search algorithm and an alpha approximation algorithm.

Expanding on this focus, Harbaoui Dridi et al. (2020) introduced an m-MDPDPTW problem with several variants, encompassing multiple vehicles (m), multiple depots (MD), pickup and delivery problem (PDP) with time windows (TW). They also developed a new algorithm based on PSO to solve the m-MDPDPTW. Zhou et al. (2021) addressed the doctor-patient matching problem in professional nursing, established a stochastic optimization model to minimize matching, and devised a two-stage decomposition algorithm to solve it. Soman and Patil (2023) proposed a scatter search heuristic to solve a two-dimensional vehicle routing problem. Shen et al. (2024) proposed a novel evolution strategy based on decomposition for high-dimensional expensive multi-objective optimization problems (EMOPs).

Bellman (1966) describes the curse of dimensionality in combinatorial optimization problems, which highlights the exponential growth in hypervolume as the problem size increases. To address this challenge, researchers have developed various strategies. Dimensionality reduction methods such as Principal Component Analysis (PCA) introduced by Hotelling (1933) and Multidimensional Scaling (MDS) introduced by Kruskal (1964) remain foundational. Wang et al. (2021) proposed a PCA-Assisted Reproduction operator (PCA-AR) that enhances evolutionary multi-objective algorithms, leveraging PCA for effective dimensionality reduction. Similarly, Tao et al. (2021) optimized a Radial Basis Function (RBF) neural network model for a coke furnace system, employing PCA for dimensionality reduction combined with the Levenberg-Marquardt and genetic algorithms to boost prediction accuracy. Hinton and Salakhutdinov (2006) demonstrated that high-dimensional data can be reduced to low-dimensional codes using deep autoencoder networks with a small central layer. Bi et al. (2023) advanced the autoencoder by developing a self-adaptive teaching–learning-based optimizer for dimensionality reduction. Anowar et al. (2021) further illustrated the critical role of these algorithms in modern computational methods.

Roulette-wheel selection is a commonly used technique in genetic and evolutionary algorithms. Lipowski & Lipowska (2012) introduced a Roulette-wheel selection algorithm based on random acceptance, which can be utilized for sampling with fitness cut-off at a specific value or for sampling without replacement. The concept of variation factor was initially employed in the field of genetic variation in agricultural hybridization but was later integrated into the field of optimal solution search, particularly within genetic algorithms (GA). The variation factor of GA pertains to the replacement of a gene value in the coding string of individual chromosomes with another gene value, thereby generating a new individual. Yu et al. (2018) devised a decision method defined by a variable weight function, examined the combination of variable weights and the absolute risk-aversion coefficient, and applied it to address the hybrid multiple attribute decision-making problem. Li et al. (2020) introduced an effective and stable quantum particle swarm optimization (EQPSO) rooted in multi-agent and multi-task assignment problems. This approach incorporates a historical task mechanism that focuses on high reward acquisition, revenue division, stability insurance, and search acceleration. In summary, Roulette-wheel selection, the Variation factor of GA, and the EQPSO algorithm provide innovative approaches for addressing multi-dimensional assignment decision problem models.

Typical MAPs often exhibit multi-dimensional, multi-constraint, and large cardinality characteristics. It generally becomes NP-hard when the number of dimensions exceeds 3 (Karp 2010; Kammerdiner et al. 2022). In this paper, the objective function of the MAM is nonlinear, and as the number of dimensions increases, so does the computational complexity (refer to formulas 2 and 3 in Section 3). As a result, traditional exact algorithms such as the branch and bound algorithm, the Hungarian method, and others are incapable of swiftly solving this type of model and obtaining a satisfactory solution. Additionally, due to parameter sensitivity, time-consuming processes or unstable convergence, traditional heuristic algorithm face challenges in providing a global optimal solution. Therefore, it becomes essential to develop a customized algorithm for efficiently solving our MAM. Therefore, this paper introduces the VMA to solve the MAM by establishing virtual matching sequences between dimensions and seeking optimal solutions within the feasible domain under given constraints. To enhance efficiency, we propose a dimensionality reduction method tailored for high-dimensional assignment models in the VMA. Compared to Wang’s (2021) knowledge-based approach, which utilizes case-based reasoning and mathematical programming for large-scale instances, our method integrates innovative features such as roulette-wheel selection, multiple mutation factors, individual influence probability, and global influence probability.

3. Mathematical model of MAP

In this section, we primarily focus on studying the common feature model of the multi-dimensional assignment problems (MAPs). MAPs exhibit distinct multi-features, including multi-agents (MA), multi-resources (MR), multi-tasks (MT), multi-stages (MS), and so forth.

These models find widespread application and allow for the construction of problem-specific models tailored to the characteristics of individual problems. Our overarching goal is to identify a satisfactory solution that minimizes cost or maximizes performance (benefit).

3.1. Problem definition

Our problem is characterized by the following parameters:

D: a set of all dimensions. $D = \{d_1, d_2, \cdots, d_n\}$ ($n \geq 2, n \in N^*$), $d_1, d_2, \cdots, d_n$ represent every dimension in MAM, including executive agent, task agent, stage, etc.

$N_{d_1}, \cdots, N_{d_k}, \cdots, N_{d_n}$ ($n \geq k \geq 2, k \in N^*, n \in N^*$): represent the cardinalities of values taken in dimensions $d_1, \cdots, d_n$ respectively. If d_k represents the dimension of assigned individuals, then N_{d_k} (for $k = 1, 2, \cdots, n$) represents the count of available assigned individuals.

$P^{i}_{d_1}, P^{i''}_{d_2}, \cdots, P^{i^{(k)}}_{d_k}, \cdots, P^{i^{(n)}}_{d_n}$ ($2 < k \leq n, k \in N^*$): the performances of different individuals $(i, i'', .., i^{(k)}, \cdots, i^{(n)})$ in each dimension $(d_1, d_2, \cdots, d_k, d_n)$. For example, when the fifth employee (assigned individuals i'') in the dimension 2 participates in an assignment, and then $P^{5}_{d_2}$ represents the performance of the fifth individual in the second dimension.

$P^{i,\cdots,i^{(n)}}_{d_1,\cdots,d_n}$($i \in (1, N_{d_1}), \cdots, i^{(n)} \in (1, N_{d_n})$): the performance of the assignment that was made of multiple dimensions. $i, \cdots, i^{(n)}$ is an individual on the $d_1, \cdots, d_n$ dimension. $i, \cdots, i^{(n)}$ corresponds to dimension $d_1, \cdots, d_n$ different individuals. Take the three dimensions as an example, $P^{i,i'',i^{(3)}}_{d_1,d_2,d_3}$ represents the performance in the 3D assignment problem, consisting of individual i in dimension d_1, individual i'' in dimension d_2, and individual $i^{(3)}$ in dimension d_3. In the subscript, d_1 represents the dimension of employees, d_2 represents the dimension of employers, d_3 represents the dimension of stages. Respectively, i, i'' and $i^{(3)}$ represent the individual of the corresponding dimension from the subscript in the superscript.

We formalize this condition by introducing the 0–1 decision variables:

$$x_{i,\cdots,i^{(n)}} = \begin{cases} 1, \textit{if individuals } i^{(k)} \textit{ are assigned together for } k = 1, 2, \cdots, n \\ 0, \textit{otherwise} \end{cases}, \tag{1}$$

for all $i^{(k)} = 1, 2, \cdots, N_{d_k}$, $i^{(k)} \in d_k$, $.k = 1, 2, .., n$

The nonlinear objective function is stated below:

maximize Z_{max} or *minimize* Z_{min}:

$$Z = \sum_{i=1}^{N_{d_1}} \cdots \sum_{i^{(n)}=1}^{N_{d_n}} \theta P^{i,\cdots,i^{(n)}}_{d_1,\cdots,d_n} x_{i,\cdots,i^{(n)}}, \ \theta \textit{ is a correction factor}, \tag{2}$$

$$P^{i,\dots,(i+y)^{(k)},\cdots,i^{(n)}}_{d_1,\cdots,d_k,\cdots,d_n} = f\left(P^{i,\dots,(i)^{(k)},\cdots,i^{(n)}}_{d_1,\cdots,d_k,\cdots,d_n}\right), \ 1 \leq y \leq N_{d_k}, \ y \in N^*, \tag{3}$$

The objective function Z aims to maximize the total performance (benefit) or minimize the cost achieved through assignment in each dimension. Different assignment behaviors and performances are not entirely independent. In the actual multi-stages assessment problem, the performance of the subsequent assessment is often affected by the preceding assessment process which can be expressed as formula 3. For example, if d_k represents a stage dimension, the function $P^{i,\dots,(i+1)^{(k)},\cdots,i^{(n)}}_{d_1,\cdots,d_k,\cdots,d_n} = f\left(P^{i,\dots,(i)^{(k)},\cdots,i^{(n)}}_{d_1,\cdots,d_k,\cdots,d_n}\right)$ signifies that the performance of the $(i+1)^{(k)}$ stage is impacted by the $(i)^{(k)}$ stage. Therefore, the MAM model proposed in this paper is non-linear.

3.2. Constraint definition

Definition: $\underline{b^d_c}$ and $\overline{b^d_c}$ (for $c = 1, 2, \cdots, n$, $\underline{b^d_c}, \overline{b^d_c} \in (-\infty, +\infty)$, $\underline{b^d_c} < \overline{b^d_c}$, for dimension d) represent the minimum and maximum values of constraint functions in the constraint set c. The number and boundary values of constraints vary for different practical MAPs.

The model includes the following constraints:

Constraint Set (I): stipulates for all individuals ($i^{(k)}$) in dimension d_k, the total number of times an individual, as specified in every other dimension, participates in an assignment cannot exceed $\overline{b^d_1}$ and must be at least $\underline{b^d_1}$. For an example of instantiation, please refer to the explanation of formulas 15, 16, and 17 in Section 5.1.1.

$s.t. i^{(k)} \in d_k, d_k \in D$

$$\underline{b^d_1} \leq \sum_{i^{(k)}=1}^{N_{d_k}} x_{i,\cdots,i^{(n)}} \leq \overline{b^d_1}, \tag{4}$$

for all $i^{(k)} = 1, 2, \cdots, N_{d_k}$, $k = 1, 2, .., n$,

Constraint Set (II): stipulates for all individuals ($i^{(j)}$ and $i^{(k)}$) in dimension d_j and d_k, the times that an individual specified in every other dimension (in the 3D problem, there is only one dimension left) together participate in an assignment, whose maximum cannot exceed $\overline{b^d_2}$ and whose minimum is not less than $\underline{b^d_2}$. For an example of instantiation, see the explanation of formulas 18 and 19 in Section 5.1.1.

$s.t. i^{(j)} \in d_j$, $i^{(k)} \in d_k$, $j \neq k$

$$\underline{b^d_2} \leq \sum_{i^{(j)}=1}^{N_{d_j}} \sum_{i^{(k)}=1}^{N_{d_k}} x_{i,\cdots,i^{(n)}} \leq \overline{b^d_2}, \tag{5}$$

for all $i^{(j)} = 1, 2, \cdots, N_{d_j}$, $i^{(k)} = 1, 2, \cdots, N_{d_k}$, $j = 1, 2, .., n$, $k = 1, 2, .., n$,

Constraint Set (III): stipulates for all individuals ($i^{(j)}$, $i^{(k)}$ and $i^{(l)}$) in dimension d_j, d_k and d_l, the total number of times that an individual, as specified in every other dimension, participates in an assignment cannot exceed $\overline{b^d_3}$ and must at least $\underline{b^d_3}$.

$s.t. i^{(j)} \in d_j, i^{(k)} \in d_k, i^{(l)} \in d_l, j \neq k, j \neq l, k \neq l$

$$\underline{b^d_3} \leq \sum_{i^{(j)}=1}^{N_{d_j}} \sum_{i^{(k)}=1}^{N_{d_k}} \sum_{i^{(l)}=1}^{N_{d_l}} x_{i,\cdots,i^{(n)}} \leq \overline{b^d_3}, \tag{6}$$

for all $i^{(j)} = 1, 2, \cdots, N_{d_j}, i^{(k)} = 1, 2, \cdots, N_{d_k}, i^{(l)} = 1, 2, \cdots, N_{d_l}$,

$j = 1, 2, .., n, k = 1, 2, .., n, l = 1, 2, .., n$,

There are n-1 types of constraint sets in the N-dimensional assignment problem. The three types of constraint sets (I, II, and III) mentioned above exist widely in various assignment problems and are among the most common constraints. Through induction, we can derive general expressions for the Constraint Set IV and for higher sets (from IV to $N-1$). Taking Constraint Set $N-1$ as an example, it can be divided into three conditions. The first condition (formula 7) is the typical one, while the last two conditions (formula 8 and 9) represent variants of the first condition, as follows.

Constraint Set $(\boldsymbol{N}-1)$: formula 7 stipulates for all individuals ($i^{(1)}, ..., i^{(k-1)}, i^{(k+1)}, ..., i^{(n)}$) in every dimension except d_k, the number of times that an individual ($i^{(k)}$), as specified in dimension d_k, participates in an assignment can not exceed $\overline{b^d_{n-1}}$ and must be at least $\underline{b^d_{n-1}}$. By analogy, the Constraint Set $N-1$ applies exclusively to N-dimensional assignment problems.

$s.t. i^{(k-1)} \in d_{k-1}, i^{(k+1)} \in d_{k+1}, d_{k-1} \in D, d_{k+1} \in D$

$$\underline{b^d_{n-1}} \leq \sum_{i^{(1)}=1}^{N_{d_1}} \cdots \sum_{i^{(k-1)}=1}^{N_{d_{k-1}}} \sum_{i^{(k+1)}=1}^{N_{d_{k+1}}} \cdots \sum_{i^{(n)}=1}^{N_{d_n}} x_{i,\cdots,i^{(n)}} \leq \overline{b^d_{n-1}}, \tag{7}$$

for $i^{(k-1)} = 1, 2, \cdots, N_{d_{k-1}}$; $k-1 = 2, .., n-1$

or

Table 1
the number of the constraints in different constraint sets.

Constraint Set	The number of the constraints
Constraint Set I	$C(n,1) = n$
Constraint Set II	$C(n,2) = n(n-1)/2$
Constraint Set x	$C(n,x) = n!/(x! \times (n-x)!)$ and $x \in [1,n-1]$, $n \geq x+1$
Constraint Set $N-1$	$C(n,n-1) = n$

$$s.t.\, i^{(k)} \in d_k, d_k \in D$$

$$\underline{b_{n-1}^{d}} \leq \sum_{i^{(2)}=1}^{N_{d_2}} \cdots \sum_{i^{(k)}=1}^{N_{d_k}} \cdots \sum_{i^{(n)}=1}^{N_{d_n}} x_{i,\cdots,i^{(n)}} \leq \overline{b_{n-1}^{d}}, \tag{8}$$

for all $i^{(k)} = 1,2,\cdots,N_{d_k}$, $k = 1,2,..,n$
or

$$s.t.\, i^{(k)} \in d_k, d_k \in D$$

$$\underline{b_{n-1}^{d}} \leq \sum_{i^{(1)}=1}^{N_{d_1}} \cdots \sum_{i^{(k)}=1}^{N_{d_k}} \cdots \sum_{i^{(n-1)}=1}^{N_{d_{n-1}}} x_{i,\cdots,i^{(n)}} \leq \overline{b_{n-1}^{d}}, \tag{9}$$

for all $i^{(k)} = 1,2,\cdots,N_{d_k}$, $k = 1,2,..,n$

Each of the above constraint sets represents a category of constraints, rather than just one single constraint. In the N-dimensional assignment model, there are typically $n-1$ constraint sets, of which only four common types are presented above due to space limitations. In the N-dimensional assignment model, Constraint Set I contains at most n constraints, while Constraint Set II contains at most $n(n-1)/2$ constraints. We can determine that Constraint Set x contains at most $C(n,x) = n!/(x! \times (n-x)!)$ constraints, where x represents type of constraint set, n represents the number of dimensions, and $x \in [1,n-1]$, $n \geq x+1$). For example, when $x = n-1$, Constraint Set $N-1$ can have at most $C(n,n-1) = n!/((n-1)! \times 1!) = n$ constraints in the N-dimensional assignment model (as shown in Table 1).

Besides the ones mentioned above, the MAM introduces various specific constraints to account for its complexity. For example, one of these constraints is described as limiting the total number of times all individuals in n dimensions can participate in assignments, as shown in formula 10.

In addition to the previously mentioned constraints, there are also other specific constraints for MAMs. For example, in formula 10, it signifies that the total number of times all individuals of n dimensions collectively participate in assignments should fall within the range of $\overline{b_n^d}$ and $\underline{b_n^d}$. The $\overline{b_n^d}$ and $\underline{b_n^d}$ is the maximum and minimum values of the constraint functions.

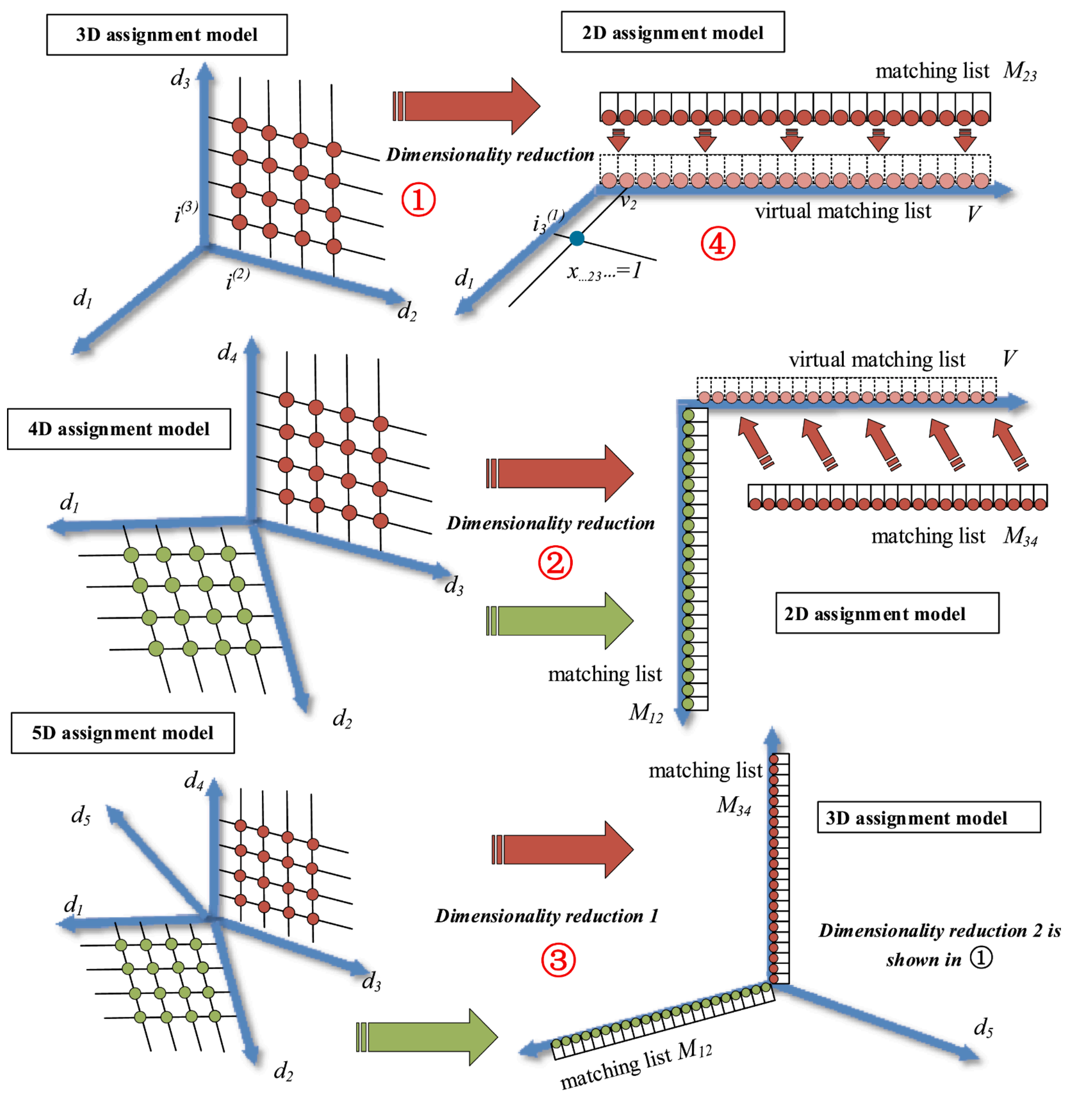

Fig. 3. The dimensionality reduction of MAM.

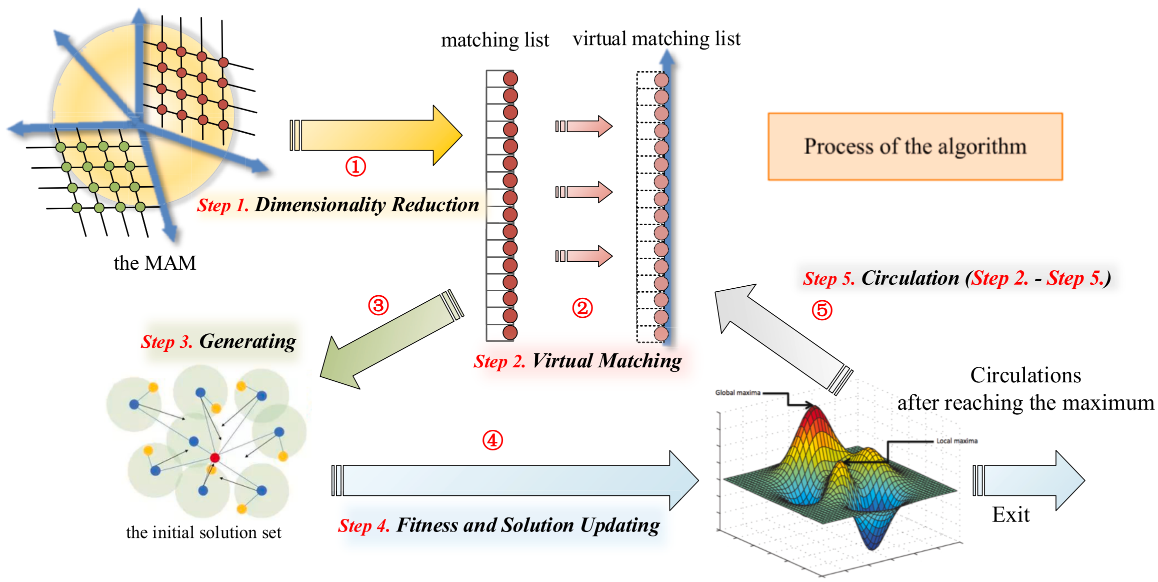

Fig. 4. The framework of VMA.

$$s.t. i^{(k)} \in d_k, d_k \in D$$

$$\underline{b_n^d} \leq \sum_{i^{(1)}=1}^{N_{d_1}} \cdots \sum_{i^{(n)}=1}^{N_{d_k}} x_{i,\cdots,i^{(n)}} \leq \overline{b_n^d}, \tag{10}$$

$$for\ all\ i^{(k)} = 1,2,\cdots,N_{d_k},\ k = 1,2,..,n$$

4. The algorithm

This section presents the Virtual Matching Algorithm (VMA) designed to solve high-dimensional and complex assignment problems. Key innovations include the introduction of virtual matching probabilities sets, dimensionality reduction through matching lists, and enhanced convergence using Roulette-wheel selection. Comparative experiments show that the Improved Virtual Matching Algorithm (IVMA), with additional dimensionality reduction, achieves superior solution quality and computational efficiency over traditional methods like PCA and Autoencoders.

4.1. Introduction of the algorithm

For solving the non-linear MAM of a NP-hard MAP, this section integrates the ideas of classical algorithms and proposes the Virtual Matching Algorithm (VMA). The key ideas are as follows: (1) We propose a method of dimensionality reduction. By replacing the original dimensions with a matching list, a MAP can be transformed into lower-dimensional matching problems. (2) To enhance the algorithm's search capability and prevent it from getting stuck in local optima, we introduce a variation factor into the position transformation of the PSO. (3) We introduce Roulette-wheel selection to improve the convergence speed of the algorithm. The probability of an individual being selected is directly proportional to their performance, meaning that a superior individual is more likely to be selected. As a result, the VMA can fulfill the requirements for quickly and effectively solving the MAM and can be extended to address other large-scale, multi-dimensional and high-order nonlinear integer programming problems.

4.2. Principle of the algorithm

***Step 1*. Dimensionality Reduction** (Fig. 4 – ①): Due to the complexity of MAM, this section introduces a method of dimensionality reduction. In a 3D assignment model, all individuals ($i^{(2)}$ and $i^{(3)}$) with dimensions d_2 and d_3 can be pairwise connected and transformed into a two-dimensional matrix. Upon expansion, the matrix can be treated as a sequence of cross-combined individuals, forming what is referred to as a matching list. This list, along with the remaining dimension d_1, specifies that the matching list and remaining dimension combine to form a new 2D space. This process is termed the dimensionality reduction of the 3D assignment problem (Fig. 3 – ①).

Similarly, in the 4D assignment model, two matching lists can be created (as shown in Fig. 3 – ②). For MAM with more than 4 dimensions, matching lists are considered new dimensions, and additional dimensionality reduction iterations are required. For example, in the 5D assignment model, the M_{12} and M_{34}, generated after the initial dimensionally reduction, need to undergo a second reduction with respect to d_5 (as shown in Fig. 3 – ③). M_{jk} , composed of the all individuals ($i^{(j)} = 1,2,\cdots,N_{d_j}, i^{(k)} = 1,2,\cdots,N_{d_k}$, for $j = 1,2,..,n, k = 1,2,..,n$), is defined as a matching list for the MAM. The length of the M_{jk} is $N_{d_j} \times N_{d_k}$.

***Step 2*. Virtual Matching** (Fig. 4 – ②): A virtual matching list $V = [v_1, \cdots]$ of the same length as the matching list is generated and used to virtually transform the assigned individuals of each dimension. v_1 is a virtual individual in the virtual matching list V, formed by combining after reducing dimensions of certain two dimensions. Using the virtual matching probability pr, virtual matching list is used to probabilistically matched and generate a new solution set $Solution^{new}$. $sol_{id}^{new} = (x_{i,\cdots,i^{(n)}}, \cdots)$ represents a new solution vector in the solution set $Solution^{new}$. For example, if v_2 from the matching list V is matched with another individual $i_3^{(1)}$ from dimension d_1, it means that the decision variable $x_{...23...}$ in the new feasible solution sol_{id}^{new} within the solution set $Solution^{new}$ is set to 1, as shown in Fig. 3 – ④.

The virtual matching probability pr can be defined as formula 11.

$$pr = r_1 \times \mathrm{rand}(V + \eta) + \lambda \times c_1 \times r_2 \times rand\left(\frac{fit_{id}^{max}}{\sum_{id=1}^{nPop} fit_{id}^{max}}\right) + \varepsilon \times c_2 \times r_3 \times rand\left(\frac{fit_{gd}^{max}}{\sum_{id=1}^{nPop} fit_{gd}^{max}}\right), \tag{11}$$

***Step 3*. Initial Solution Generating** (Fig. 4 –③): In a N-dimensional decision space, an initial solution set $Solution = [sol_1, \cdots, sol_u]$ is randomly generated based on a matching list mapped from the

Table 2
Comparison of difference parameter settings for the VMA.

VMA u	r_1	r_2	r_3	Satisfactory solution min	avg	max	relative error	CPU Time(/s) min	avg	max
5	0.1	0.3	0.6	1985.080	1988.733	1991.088	0.30 %	1.250	4.180	7.391
	0.3	0.2	0.5	1984.294	1988.644	1990.538	0.31 %	0.594	2.467	4.313
	0.3	0.3	0.4	1984.130	1988.043	1990.498	0.32 %	1.266	2.721	3.953
	0.5	0.1	0.4	1985.037	1988.199	1990.402	0.27 %	0.531	1.835	2.781
	0.5	0.2	0.3	1985.429	1988.341	1990.737	0.27 %	1.188	2.097	2.734
	0.5	0.25	0.25	1984.526	1988.234	1990.784	0.31 %	0.406	1.701	2.625
10	0.1	0.3	0.6	1988.302	1990.201	1992.892	0.23 %	2.734	6.054	9.844
	0.3	0.2	0.5	1986.922	1990.207	1992.380	0.27 %	1.344	4.089	6.828
	0.3	0.3	0.4	1987.392	1990.225	1992.676	0.27 %	1.328	4.099	6.672
	0.5	0.1	0.4	1988.193	1990.334	1992.718	0.23 %	1.141	3.557	5.063
	0.5	0.2	0.3	1987.803	1989.992	1991.527	0.19 %	1.172	3.210	5.563
	0.5	0.25	0.25	1987.096	1989.634	1991.218	0.21 %	1.172	2.722	4.172
15	0.1	0.3	0.6	1988.942	1990.741	1992.615	0.18 %	4.156	8.829	14.516
	0.3	0.2	0.5	1989.414	1991.100	1992.616	0.16 %	2.563	6.779	10.172
	0.3	0.3	0.4	1988.661	1990.794	1992.820	0.21 %	2.422	5.945	9.953
	0.5	0.1	0.4	1988.978	1990.922	1992.782	0.19 %	2.344	5.703	7.953
	0.5	0.2	0.3	1987.570	1990.653	1992.591	0.25 %	1.578	4.683	6.984
	0.5	0.25	0.25	1988.136	1990.291	1992.009	0.19 %	2.000	4.920	6.672
20	0.1	0.3	0.6	1988.636	1991.188	1993.215	0.23 %	4.375	10.954	20.266
	0.3	0.2	0.5	1989.784	1991.477	1993.374	0.18 %	2.094	6.664	11.359
	0.3	0.3	0.4	1989.706	1990.984	1993.503	0.19 %	2.766	7.699	11.703
	0.5	0.1	0.4	1989.163	1991.236	1992.738	0.18 %	2.359	6.163	9.406
	0.5	0.2	0.3	1989.832	1991.270	1992.935	0.16 %	2.156	5.367	9.141
	0.5	0.25	0.25	1988.843	1991.216	1992.836	0.20 %	2.344	5.752	8.813
25	0.1	0.3	0.6	1989.957	1991.844	1993.496	0.18 %	3.797	11.906	20.031
	0.3	0.2	0.5	1988.741	1991.587	1993.509	0.24 %	3.047	8.758	12.609
	0.3	0.3	0.4	1989.237	1991.712	1993.037	0.19 %	4.172	9.242	14.391
	0.5	0.1	0.4	1989.085	1991.502	1993.116	0.20 %	3.672	8.387	10.953
	0.5	0.2	0.3	1990.347	1991.618	1992.713	0.12 %	2.469	6.999	10.031
	0.5	0.25	0.25	1989.662	1991.289	1993.293	0.18 %	3.063	7.711	10.141

dimensional $d_1, d_2, \cdots, d_n$, consisting of u feasible solutions. Testing suggests that maintaining u within the range of 10 to 25 can achieve a balance between convergence and the algorithm's ability to produce satisfactory solutions. Each individual solution $sol_{id} = \left(x_{i,\cdots,i^{(n)}}, \cdots\right)$ represents a solution vector within solution set *Solution*. Each global solution corresponds to a solution set *Solution*.

η is an inertia coefficient, which represents the step size and is typically set to 1. λ is a variation factor for individual solutions, indicating the probability that an individual solution is not influenced by the historical satisfactory individual solution fit_{id}^{max}. ε is a variation factor for global solutions (solution set *Solution*), indicating the probability that an individual solution is not influenced by the historical satisfactory global solution fit_{gd}^{max}. In line with the tradition of variation factors in genetic algorithm, λ and ε are commonly set to 0.1 (Liu & Gao, 2009; Jia et al., 2018; Ghoseiri & Ghannadpour, 2010). $r_1, r_2, r_3 \in [0, 1]$ are equilibrium factors. To balance each component of probability *pr*, $r_1 + r_2 + r_3 = 1$ is set. Additionally, c_1 and c_2 are the attenuation coefficients. After a specific number of iterations, the influence of variation factors r_1, λ and ε starts to attenuate. fit_{id} represents the fitness of an individual solution sol_{id} within a solution set *Solution*. fit_{id}^{new} represents the fitness of a new individual solution sol_{id}^{new}. fit_{id}^{max} is the historical optimal fitness of an individual solution sol_{id}. fit_{gd}^{max} is the historical optimal fitness of the global solution *Solution*.

***Step 4*. Fitness and Solution Updating** (Fig. 4 – ④): In the VMA, the objective function (1.1) is considered as the fitness function. For current solution set, if a new individual fitness fit_{id}^{new} is greater than the historical optimal fitness fit_{id}^{max}, the fit_{id}^{max} is updated by fit_{id}^{new}. Similarly, if $fit_{id}^{max} > fit_{gd}^{max}$, the fit_{gd}^{max} is updated to fit_{id}^{max}. Algorithm 1 shows the processes of Fitness Updating. The satisfactory solution is a solution vector that corresponds to the global fitness fit_{gd}^{max}. If fit_{id}^{new} is greater than fit_{id}, the current sol_{id} and solution set *Solution* are updated with the new solution $sol-_{id}^{new}$ and solution set $Solution^{new}$ respectively.

Algorithm 1 Calculate fitness:

Input: $Solution^{new}$
Define fitness function
for n in $Solution^{new}$ **do**
Let fitness function $\leftarrow S_{id}^{new}$
if $fit_{id}^{new} > fit_{id}^{max}$:
Update $fit_{id}^{max} \leftarrow fit_{id}^{new}$
end if
if $fit_{id}^{new} > fit_{gd}^{max}$:
Update $fit_{gd}^{max} \leftarrow fit_{id}^{new}$
end if
end for
Outout: fit_{id}^{max} and fit_{gd}^{max}

Step 5*. Circulation** (Fig. 4 – ⑤): the above steps (③-④) are repeated in the circulation. g_{max} is set as the maximum number of circulations. The program performs attenuation based on the attenuation coefficients c_1 and c_2. h is defined as a counter and initially set to 100, and the attenuation period is $\frac{1}{h+1} g_{max}$. After completing g_{max} circulations, the calculation is terminated, and the satisfactory solution is generated. The Algorithm 2 outlines the processes from ***Step 2 to ***Step 5***, and the space complexity of the VMA is $O(n^3)$.

Algorithm 2 Searches for optimal solution by VMA:

Input data; $D = d_1, d_2, \cdots, d_n$; $\{N_{d_1} \cdots N_{d_n}\}$
Define *Solution* $\in$ feasible region; M_{jk}; V; η; λ; ε
for all solution **do**:
Define *pr*;
for dimension d_1 **to** d_n **do**
Get feasible region of solution
while True:
if all (for i in M_{jk}) is spare **then**
Update V
Until V is whole
end if
endwhile

(*continued on next page*)

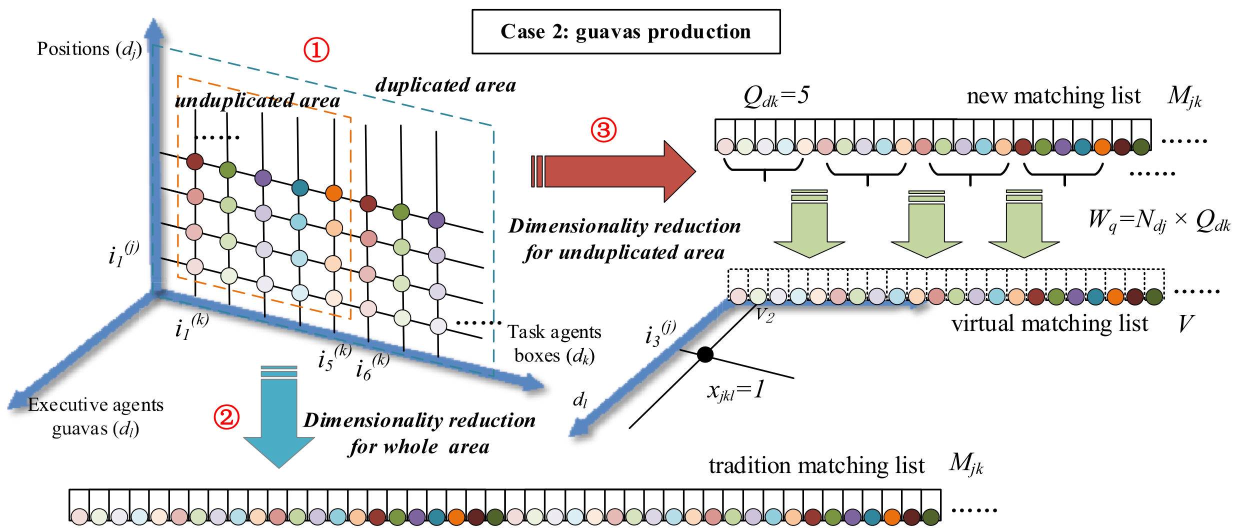

Fig. 5. Experimental result of difficult parameter setting.

(*continued*)

V is moved by pr
Calculate fitness of solution ←V (The Algorithm 1)
Update *Solution*; pr; λ; ε
end for
end for
Output a feasible solution

4.3. Algorithm analysis and optimization

In the 3D problems, by conducting computational experiments on data sets of different sizes, a time complexity function $t(N_1, N_2, N_3) = 1.03788 \times N_3 - 0.31515 N_2 + 2.41288 N_1$ can be fitted. The function indicates the relationship between the amount of data (.

$N_1, N_2, N_3)$ in the dimensions (d_1, d_2, d_3) and the algorithm's execution time. The time complexity of the VMA is $O(t(N_1, N_2, N_3))$ in the 3D problems. For four-dimensional and higher-dimensional models, the time complexity, $O(t(n))$, of the algorithm depends on the specific data volume and data structure analysis.

4.3.1. Parameter Setting

Using the data from Case 1, we conducted 192 tests on a Windows 11 64-bit platform equipped with an Intel Core i7-13700H processor. These tests aimed to examine the influence of parameters on both the quality of satisfactory solutions and the algorithm's average running time (see Appendix D), building on the experimental framework established by Yang et al. (2011) regarding equilibrium factors. By referring to the initial solution count in the classical heuristic algorithm (Kennedy & Eberhart, 1995; Harbaoui et al., 2020), we verified the range of counts [5,10,15,20,25] (denoted as u). The testing results (Table 2) are consistent with Yang's conclusions and demonstrate that as u increases, the satisfactory solution improves, but the running time also increases.

Considering both satisfactory solution and running time comprehensively, we have selected the following thirty parameter settings, in alignment with the approach outlined in Li et al. (2019). The data from 900 tests can be found in Appendix E. In Case 1, the results corresponding to various parameter settings in the algorithm are presented in Table 2. It is evident that the maximum relative error of the satisfactory solutions is mere 0.32 % across the 300 tests. This fact underscores the robustness and steadiness of the VMA. The relative error is calculated using the following formula outlined by Yang et al. (2011), where "*Sati_Solution*" denotes the fitness of satisfactory solution obtained through the VMA.

$$relative\ error = \frac{maximalSati\_Solution - minimal\,Sati\_Solution}{averageSati\_Solution}$$

For the same u and different groups of parameters r_1, r_2 and r_3, the relative error in Table 2 does not exceed 0.32 %. This also highlights the minimal sensitivity of the satisfactory solutions is sensitive to the parameters r_1, r_2 and $.r_3$

4.3.2. Algorithm optimization

Sometimes, there can be a high number of duplicate individuals within a single dimension. For example, in Case 2, as illustrated in Fig. 5 – ①, there are only five distinct classifications (ranging from $i_1^{(k)}$ to $i_5^{(k)}$, representing an unduplicated area), and a substantial number of duplicate boxes are present in the dimension d_k. If the VMA is utilized to address this specific type of special MAP, the resulting matching list may become overly lengthy following dimensionality reduction, potentially leading to redundant calculations (refer to Fig. 5 – ②). To mitigate this issue, we introduce the improved virtual matching algorithm (IVMA). We define Q_{d_k} as the count of distinct classifications of duplicate individuals within dimension k. Subsequently, we create a new virtual matching list V based on individuals from dimension j and the classes of individuals in dimension k. The length of this list is $N_{d_j} \times Q_{d_k}$ (as shown in Fig. 5 – ③). We also define w_q, $q \in (1, Q_{d_k})$ as the capacity for each class q of individuals in the virtual matching list V. When an individual from a particular class is assigned, its capacity w_q is deducted accordingly. When w_q for class q reaches 0, it indicates that all individuals in that class have been assigned. The number of iterations for IVMA is less than VMA by $(N_{d_j} \times N_{d_k} - N_{d_j} \times Q_{d_k}) \times N_{d_l} \cdots$ times.

5. Instantiation of MAM

In this section, we transform the abstract MAM into two concrete 3D models based on the specific MAPs with different real backgrounds. There are many common constraints between the two models. It is worth noting that the third dimension of the former model is time (stages), while that of the latter is resources (positions). The VMA algorithm designed in Sections 4.1 and 4.2 is used to solve general MAPs, such as Case 1. The IVMA algorithm designed in Section 4.3 is used to address specific MAPs characterized by a mass of duplicate individuals in one dimension, such as Case 2.

Table 3
Weight values for performance of stages.

S	S1	S2	S3	S4	Sum
γ	0.15	0.25	0.25	0.35	1

5.1. Case 1: Organizational assessment

At the University of South China, the student organization needs to recruit new members (task agents) every year, contingent upon the abilities of senior members' (executive agents) abilities to conduct one-on-one assessments with them. The annual assessment is divided into four quarterly stages with varying weights (Table 3) and is carried out successively.

According to the assessment principle of the student organization, each senior member cannot assess the same junior members repeatedly (see formula 15 in Section 5.1.1). Each junior member must be assessed four times each year (refer to formula 18) and once at each stage (formula 16). Due to the demands of study and work, each senior member is limited to participating in three assignments (formula 19) throughout the assessment, with one assignment allowed at each stage (formula 17). The competence (P_1^e) of senior members is comprehensively evaluated based on eight indicators, including ideal and belief, personal cultivation, mass foundation, expression and communication, innovation, academic performance, lifestyle, and spiritual outlook. The performance (P_2^t) of junior members is comprehensively evaluated, considering factors such as innovation, academic performance, lifestyle, spiritual outlook, and more. The data for Case 1 is presented in Appendix B. The challenge faced by the organization's managers in this case revolves around determining the optimal assignment of senior members (executive agents) of varying competences to new members (task agents) at each stage, ensuring the maximization of the overall organizational assessment effect.

5.1.1. The MAM of Case 1

Given the MAPs characteristics involving multi-stages (MS), multi-agents (MA), and multi-constraints (MC) in this case, the MAM can be employed to consider the relevant three dimensions, namely executive agents, task agents, and stages. The MA implies the presence of typically more than two agents for this type of problem. The three-dimensional assignment problem comprises two agents (A and B). The primary relationships entail senior member A being assigned to evaluate junior member B to accomplish assessment task T. The assessment is divided into four stages (quarters). Each relationship between A and B is constrained by its respective stage, specifically, in terms of multi-stages (MS). A is designated as the executive agent, and B is referred to as the task agent. Meanwhile, the relationships are subject to additional conditions, including: A can only assess B once in a specific stage; B must complete one task in each stage and totally 4 tasks in all stages; A can assess at most 3 task agents in all stages. These constraints collectively constitute the multi-constraints (MC) characteristics of the problem.

5.1.2. Parameters and variable definitions

d_1, d_2, d_3 represent the three dimensions in the 3D assignment problem. e, t, s represent different assigned individuals within the dimension d_1, d_2, d_3, respectively, where these individuals are executive agents, task agents, and stages. N_1, N_2, N_3 represent the cardinalities of individuals in dimension d_1, d_2, d_3 respectively, with N_1, N_2, and N_3 representing the maximum numbers of executive agents, task agents, and stages, respectively. P_1^e, P_2^t, P_3^s represent the performances of different assigned individuals (e,t,s) in dimension d_1, d_2, d_3 respectively. The decision variable is as follows:

$$x_{ets} = \begin{cases} 1, ifindividualsetsareassignedtogether \\ 0, otherwise \end{cases}, \tag{12}$$

P_{123}^{ets} is the performance of an assignment executed by individuals e, t, s, and as followed.

$$P_{123}^{ets} = \begin{cases} \alpha \times P_1^e + \beta \times P_2^t + \gamma \times P_3^s, \forall s = 1 \\ \alpha \times P_1^e + \beta \times P_2^t + \gamma \times P_3^s + \delta \times P_{123}^{et(s-1)}, \forall s > 1, s \in N^* \end{cases}, \tag{13}$$

α: competence weight of executive agents. β: performance weight of task agents. γ: weight of stages. δ: the effect weight of the executive agents and task agents in the previous stage. The objective function is formulated as:

$$maximize Z_{max} = \sum_{e=1}^{N_1} \sum_{t=1}^{N_2} \sum_{s=1}^{N_3} \frac{P_{123}^{ets} x_{ets}}{N_3}, \tag{14}$$

The objective function Z_{max} aims to maximize the overall performance in the multi-stages assignment. The model contains the following constraints:

5.1.3. Constraint set (I)

Constraint (1): stipulates for all stages in dimension d_3, the times that the executive agent and task agent specified in d_1 and d_2 jointly participate in an assignment should not exceed a maximum of 1. This implies that a single executive agent (senior member) can only assess one task agent (junior member) once across all stages.

$$\sum_{s=1}^{N_3} x_{ets} \leq 1, \forall e \in d_1; \forall t \in d_2, \tag{15}$$

Constraint (2): stipulates for all executive agents in dimension d_1, the times that the task agent and stage specified in d_2 and d_3 jointly participate in an assignment should not exceed a maximum of $\overline{b_{t/s}}$. In this case, $\overline{b_{t/s}} = 1$, indicating that each task agent can only be assessed by any executive agent once within a single stage.

$$\sum_{e=1}^{N_1} x_{ets} = \overline{b_{t/s}}, \forall t \in d_2; \forall s \in d_3, \overline{b_{t/s}} = 1, \tag{16}$$

Constraint (3): stipulates for all task agents in dimension d_2, the times that the executive agent and stage specified in d_1 and d_3 jointly participate in an assignment should not exceed a maximum of $\overline{b_{e/s}}$. In this case, $\overline{b_{e/s}} = 1$, indicating that e each executive agent can only assess any task agent once within a single stage.

$$\sum_{t=1}^{N_2} x_{ets} \leq \overline{b_{e/s}}, \forall e \in d_1; \forall s \in d_3, \overline{b_{e/s}} = 1, \tag{17}$$

5.1.4. Constraint set (II)

Constraint (4): stipulates for all executive agents and stages in dimension d_1 and d_3, the times that the task agents specified in d_2 jointly participate in an assignment should be equal to $\overline{b_t}$. In this case, $\overline{b_t} = 4$, indicating that each task agent can be assessed by any executive agent up to four times once in each stage.

$$\sum_{e=1}^{N_1} \sum_{s=1}^{N_3} x_{ets} = \overline{b_t}, \ \forall t \in d_2, \overline{b_t} = 4, \tag{18}$$

Constraint (5): stipulates for all task agents and stages in dimension d_2 and d_3, the times that the executive agents specified in d_1 jointly participate in an assignment should be equal to $\overline{b_e}$. In this case, $\overline{b_e} = 3$, e indicating that each executive agent can assess any task agent up to three times once in each stage.

$$\sum_{t=1}^{N_2} \sum_{s=1}^{N_3} x_{ets} \leq \overline{b_e}, \ \forall e \in d_1, \overline{b_e} = 3, \tag{19}$$

Through the aforementioned constraints, the 3D assignment model is designed to approximate the 3D assignment problem under the three-dimensional features.

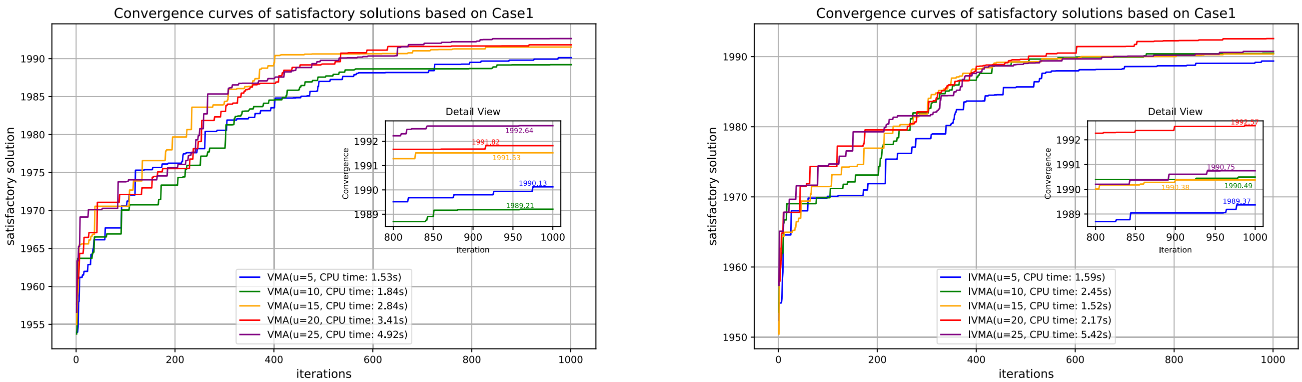

Fig. 6. Convergence curves of satisfactory solutions for Case 1.

Table 4
Schedule of assessment of Case 1.

e index	t index	s index	e index	t index	s index	e index	t index	s index	e index	t index	s index
4	1	1	21	1	2	14	1	3	22	1	4
37	2	1	31	2	2	45	2	3	42	2	4
28	3	1	29	3	2	48	3	3	47	3	4
18	4	1	3	4	2	9	4	3	24	4	4
57	5	1	24	5	2	27	5	3	8	5	4
3	6	1	53	6	2	55	6	3	31	6	4
17	7	1	45	7	2	30	7	3	32	7	4
55	8	1	51	8	2	52	8	3	11	8	4
15	9	1	8	9	2	42	9	3	43	9	4
58	10	1	4	10	2	26	10	3	14	10	4
49	11	1	17	11	2	34	11	3	40	11	4
43	12	1	15	12	2	22	12	3	51	12	4
39	13	1	48	13	2	47	13	3	37	13	4
27	14	1	32	14	2	49	14	3	54	14	4
60	15	1	33	15	2	46	15	3	6	15	4
29	16	1	11	16	2	6	16	3	34	16	4
20	17	1	46	17	2	2	17	3	21	17	4
53	18	1	9	18	2	20	18	3	30	18	4
33	19	1	2	19	2	54	19	3	26	19	4
52	20	1	18	20	2	40	20	3	39	20	4

5.1.5. Solution

On the Windows 11 64-bit platform, equipped with an Intel Core I7-13700H processor, we developed the model and solving algorithm using Python. The length of the matching list is $N_3 \times N_2 = 64$. The variation factor is $\lambda = \varepsilon = 0.1$. The equilibrium factors are $r_1 = 0.3$, $r_2 = 0.2$, and $r_3 = 0.5$. When the length of the initial solution is set to 20 and there are 1000 iterations, the maximum of the objective function after running and solving is 1992.64, achieved in 4.92 s. We draw the convergence curve based on each iteration. Convergence serves as one of the criteria for estimating how quickly and efficiently an algorithm can find a satisfactory solution. As the number of iterations increases, a satisfactory solution can be found, and the graph tends to stabilize after 700 iterations. The result is shown in Fig. 6.

The optimal assessment schedule was obtained using IVMA and is presented in Table 4. In last year's organizational assessment, the satisfactory solution (schedule) obtained through the application of the model and algorithm significantly reduced the working time compared to the previous manually arranged subject approach. Its validity and rationality have been confirmed. Therefore, the MAM and its solving algorithm (IVMA) can be efficiently applied in the organizational assessments, facilitating swift scheduling, and significantly enhancing organizational decision-making efficiency.

5.2. Case 2: Guava production

Fengqing Breeding Co. LTD, an agribusiness in Guangzhou province of China, primarily produces guava, star fruit, orange, and other varieties. These fruits are sourced from Malaysia and other regions. The planting area benefits from fertile soil and a climate favorable for cultivation. Moreover, the company employs advanced planting technology such as greenhouses and interplanting. As a result, Fengqing's fruits are sold globally. Guavas (executive agents) are harvested four times a year, resulting approximately 1000 ($N_1 = 1000$) daily outputs. During the harvesting period, high-quality guavas must be packaged into boxes (task agents). The company is equipped with a non-destructive testing instrument for guava quality, utilizing Near Infrared Reflectance Spectroscopy (NIRS). This instrument analyzes fruit sugar, acidity, ripeness, freshness, and other physical and chemical indicators. The P_1^e represents the quality values of the guavas, as shown in Appendix C. To accommodate diverse demands, five classifications of boxes are available for providing to package different guavas. P_2^t represents the box costs, as detailed in Table 5.

Table 5
Preservation index P_3^s at positions.

Position	1	2	3	4	5	6
P_3^s	0.125	0.2	0.125	0.25	0.05	0.25
Classifications of Box	1	2	3	4	5	
P_2^t	20	35	50	65	85	

Due to labor and output constraints, only 100 boxes ($N_2 = 100$) can be processed daily. Each box is divided into two layers, with each layer having three guava positions ($N_3 = 6$, $\overline{b_{e/s}} = \lceil N_2 \rceil$, $\underline{b_{e/s}} = 1$). Drafty

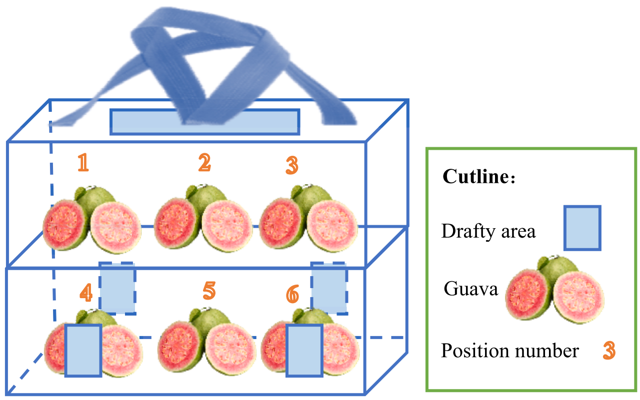

Fig. 7. Illustration of guava packaged into boxes.

positions contribute to maintaining the freshness of guavas for an extended period. The priority ranking of positions is "4 = 6 > 2 > 1 = 3 > 5", as shown in Fig. 7. It is evident that each guava can only be packaged into the same box once ($\overline{b_e} = 1$) and each box can only be used once ($\overline{b_t} = 1$). The P_3^s represents the normalized fresh-keeping capacities of the six positions in the boxes, as detailed in Table 5. To maintain freshness, guavas should be stored in well-ventilated containers. The challenge faced by Fengqing's managers in this case is determining how to package guavas of varying qualities into the six positions of the boxes. This involves ensuring that the guavas remain fresh for as long as possible, thereby maximizing the overall packing effectiveness.

5.2.1. The MAM of Case 2

Most parameters of the MAM in Case 2 are consistent with those in Case 1, with only a few updated parameter values shown in Fig. 8. In Case 2, guava is the dimension of "executive agents" (d_1), the box is the dimension of "task agents" (d_2), and the position in each box is the dimension of positions (or resources) (d_3). It is worth noting that in this case, the positions in the boxes represent the dimension of the "stages S" proposed by the model in Case 1. This indicates that the third dimension is not necessarily "stages" (as in the previous case) but can also be a dimension of positions (as in Case 2) in the 3D models.

5.2.2. Solution

On the Windows 11 64-bit platform, equipped with an Intel Core I7-13700H processor, we constructed the model and IVMA based on Python for Case 2. The length of the matching list is $N_3 \times N_2 = 600$. The variation factor is $\lambda = \varepsilon = 0.1$. The equilibrium factors are $r_1 = 0.3$, $r_2 = 0.2$, and $r_3 = 0.5$. Setting the length of the initial solution to 20 and conducting 1000 iterations, the maximum of the objective function after solving is 1669.83, taking 96.89 s. The convergence curve is drawn in Fig. 9. In conclusion, the satisfactory solution of guavas packaging is presented in Appendix F.

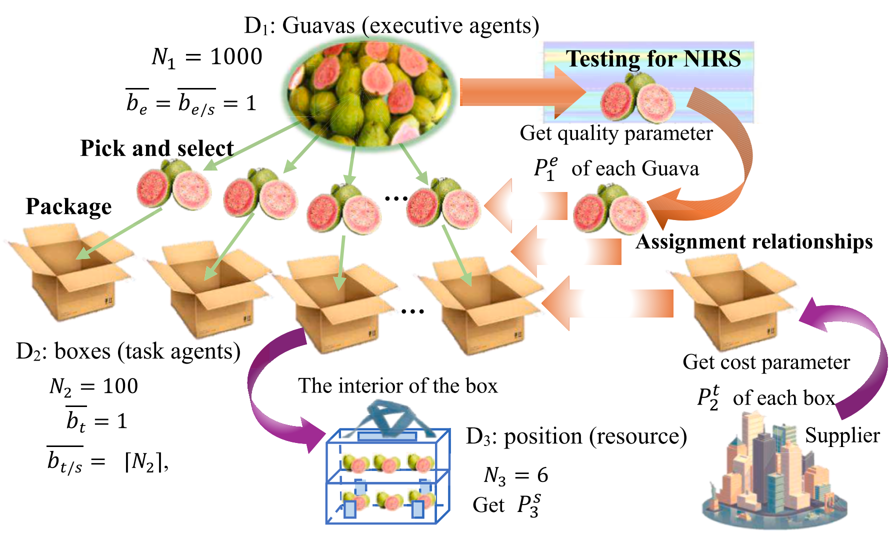

Fig. 8. The 3D dimensional model of guavas production.

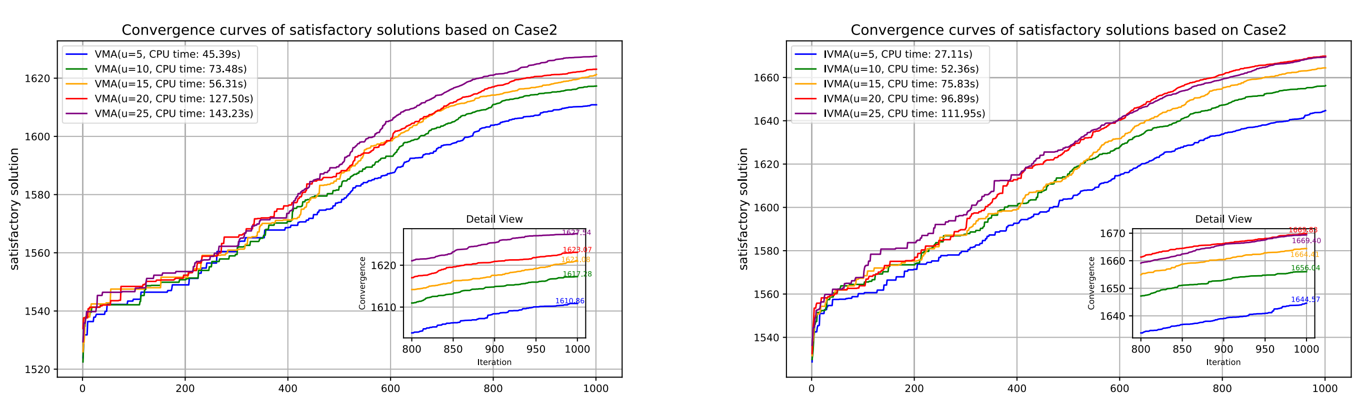

Fig. 9. Convergence curves of satisfactory solution for Case 2.

Table 6
Comparison of different dimensionality reduction methods.

Feature	Autoencoder	PCA	Dimensionality Reduction in IVMA
Computational Complexity	High	Low	Low
Feature Extraction Ability	Strong	Moderate	Strong
Data Requirements	High, performs best with large datasets.	Moderate.	High with various data sizes.
Parameter Setting	Difficult	Simple	Simple
Application Scenarios	Complex data such as images and text	Quick decision-making problems	Multdecision-making problems

6. Discussion

6.1. The advantages of MAM: Scalability and universality

Scalability of Dimensions: Section 2.1 highlights that most significant works focus on assignment problems within specific dimensions (Liu & Gao, 2009; Harbaoui et al., 2020; Li et al., 2021; Samanta et al., 2024). The proposed MAM offers a notable advantage over traditional 2D and 3D models, particularly in addressing complex decision-making scenarios with multi-characteristic constraints. Compared to conventional models, which are often constrained to fixed-dimensional applications, the MAM provides precise descriptions across a spectrum ranging from 2D to N-dimensional assignment problems (Liu & Gao 2009, Harbaoui et al. 2020, Li & Li et al., 2021, Samanta et al. 2024). This scalability allows for the dynamic integration of various critical factors such as time, locations, agents or other characteristics, facilitating the construction of N-dimensional models tailored to practical problems in diverse complex levels.

Universality of Application Scenarios: Section 2.1 indicates that most significant works primarily focus on assignment problems in specific scenarios (Ghoseiri & Ghannadpour, 2010; Harbaoui et al., 2020; Chaosheng et al., 2019; Deng et al., 2020). The proposed MAM demonstrates the ability to describe complex high-dimensional problems, showcasing its versatility across various scenarios. For instance, in Case 1, MAM was used to model an organizational assessment problem characterized by an "Executive Agents – Task Agents – Stages" framework. The multi-stage nature of the model captures the temporal dynamics inherent in organizational evaluations. In Case 2, MAM was applied to a production scheduling problem where employers (executive agents) are assigned to assess employees (task agents), reflecting the spatial layout of guava packaging processes across multiple locations (resources). Moreover, the MAM can be tailored to address diverse case-specific challenges, making it adaptable to a wide range of applications. Compared to traditional models that rely on rigid parameters and constraint settings, the MAM's flexible and universal design enables it to effectively handle complex, multi-dimensional challenges across diverse fields with both precision and efficiency.

6.2. Comparison of dimensionality reduction methods

In Section 4, we introduce a novel dimensionality reduction approach designed to transform MAPs into lower-dimensional matching problems, thereby improving computational efficiency and solution accuracy. Existing dimensionality reduction methods have demonstrated varying effectiveness in simplifying high-dimensional decision-making problems, primarily by converting complex, high-dimensional structures into lower-dimensional spaces. Techniques such as Principal Component Analysis (PCA) (Wang et al., 2021; Anowar et al., 2021) face challenges when dealing with the computational complexity of large datasets, as shown in Table 6. While Autoencoders (Bi et al., 2023) offer effective solutions for high-dimensional problems, they also struggle with time efficiency. To validate the effectiveness of our proposed method, we conducted ablation experiments, comparing it with other established dimensionality reduction techniques, including PCA and Autoencoders. In contrast, our approach within the Improved Virtual Matching Algorithm (IVMA) framework proves to be more practical for real-world applications, where complex interdependencies are often encountered.

Our algorithms, IVMA and VMA, were compared with PCA and Autoencoder, and the results are displayed in Fig. 10. The IVMA's reduced computational complexity highlights its adaptability for multidimensional decision-making tasks. In Case 1 (a small-scale data scenario), although the performance differences among the methods were less pronounced, IVMA can deliver better optimal solutions and shorter running times compared to other approaches. In Case 2 (a large-scale data scenario), the advantages of our dimensionality reduction method became more apparent. Leveraging the proposed dimensionality reduction approach, IVMA outperformed PCA- and Autoencoder-based methods, achieving higher satisfactory solution scores while significantly reducing CPU time. The results of these two cases highlight the robustness of IVMA. Specific results and comparisons are detailed in Table 7.

6.3. Comparison of algorithms

Algorithms such as Particle Swarm Optimization (PSO), Binary PSO (Bi-PSO), Genetic Algorithm (GA), and hybrid GA-PSO are widely recognized as standard heuristic algorithms for solving multi-stage, multi-resource assignment problems. These algorithms have been

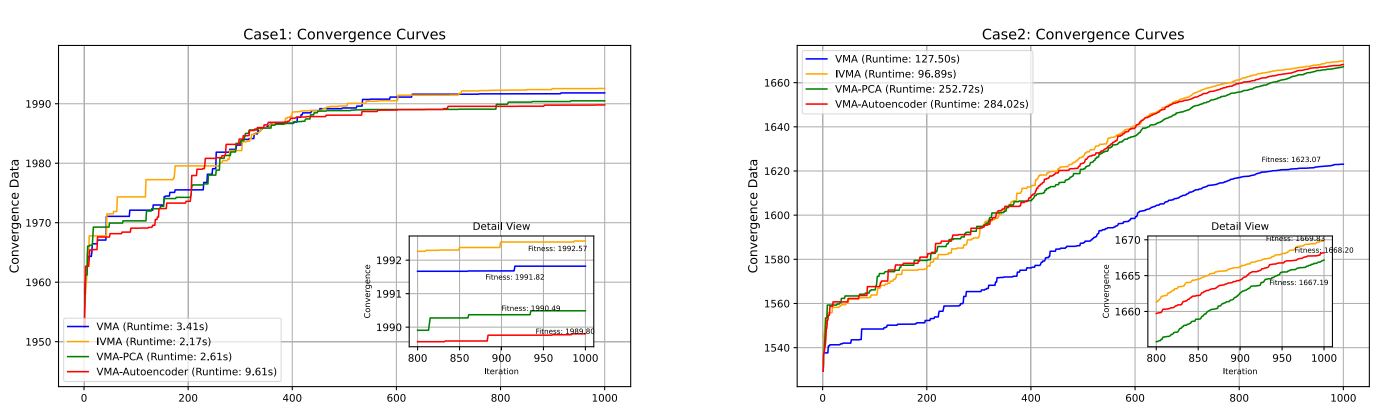

Fig. 10. Convergence curves of different dimensionality reduction methods.

Table 7
Results of different dimensionality reduction methods.

	u	Case 1 Max satisfactory solution	Avg CPU Time	Case 2 Max satisfactory solution	Avg CPU Time
VMA	5	1990.127	1.53	1610.859	45.39
	10	1989.205	1.84	1617.303	73.48
	15	1991.527	2.84	1621.275	56.31
	20	1991.821	3.41	1623.073	127.50
	25	1992.643	4.92	1627.575	143.23
IVMA	5	1989.369	1.59	1644.693	27.11
	10	1990.495	2.45	1656.221	52.36
	15	1990.376	1.52	1664.410	75.83
	20	1992.572	2.17	1669.828	96.89
	25	1990.750	5.42	1669.425	111.95
VMA – PCA	5	1988.398	1.64	1645.943	62.59
	10	1988.614	1.33	1656.665	221.30
	15	1990.431	1.95	1663.636	324.19
	20	1990.488	2.61	1667.217	252.72
	25	1993.445	5.27	1668.861	295.66
VMA – Autoencoder	5	1985.621	5.97	1648.252	87.31
	10	1990.329	9.36	1657.894	126.50
	15	1990.605	9.61	1662.841	163.98
	20	1989.800	12.06	1668.200	284.02
	25	1991.952	12.77	1671.018	298.28

Other parameters: η=1, $\lambda = \varepsilon = 0.1, r_1 = 0.3, r_2 = 0.2$, and $r_3 = 0.5$, Iterations is 1000.
For Autoencoder, the default settings can include an encoding dimension set to one-third of the input features, ReLU as the activation function, Adam as the optimizer with a learning rate of 0.001, 50 epochs for training, and a batch size of 32.
For PCA, the default parameters can retain 95% of the explained variance, with data standardized and mean-centered beforehand.

Table 8
Comparison for VMA, IVMA and other algorithms.

		Case 1		Case 2	
	U	Max satisfactory solution	Avg CPU Time	Max satisfactory solution	Avg CPU Time
VMA	5	1990.127	1.53	1610.859	45.39
	10	1989.205	1.84	1617.303	73.48
	15	1991.527	2.84	1621.275	56.31
	20	1991.821	3.41	1623.073	127.50
	25	1992.643	4.92	1627.575	143.23
IVMA	5	1989.369	1.59	1644.693	27.11
	10	1990.495	2.45	1656.221	52.36
	15	1990.376	1.52	1664.410	75.83
	20	1992.572	2.17	1669.828	96.89
	25	1990.750	5.42	1669.425	111.95
GA	5	1952.652	5.30	1527.188	137.27
	10	1946.249	4.70	1524.277	168.52
	15	1950.400	4.84	1530.191	232.11
	20	1959.816	5.03	1537.759	261.12
	25	1948.855	8.59	1535.505	411.82
BI-PSO	5	1957.465	1.50	1547.462	149.36
	10	1960.925	1.95	1549.096	184.38
	15	1971.312	3.75	1555.242	392.64
	20	1970.987	2.78	1553.361	475.03
	25	1972.457	6.83	1554.78	637.12
PSO	5	1957.851	1.81	1547.561	267.73
	10	1965.653	2.89	1537.271	125.67
	15	1957.167	3.03	1534.870	217.73
	20	1963.898	2.66	1543.490	260.02
	25	1965.366	5.89	1548.159	243.23
GA-PSO	5	1964.784	4.27	1668.861	295.66
	10	1972.759	5.84	1564.556	261.77
	15	1971.312	5.05	1592.822	127.16
	20	1972.280	5.97	1563.404	475.03
	25	1981.402	8.97	1650.951	209.17

Iterations is 1000. $\lambda = \varepsilon = 0.1, r_1 = 0.3, r_2 = 0.2$, and $r_3 = 0.5$ for the VMA and the IVMA. The parameters for BI-PSO and PSO are kept consistent with those of VMA.
For the GA, the parameter u represents the number of chromosomes, with a mutation rate of 0.01 and a crossover rate of 0.7. For the GA-PSO, the parameters are kept consistent with those of PSO and GA.

extensively studied and successfully applied across various domains. Their foundational contributions and versatility make them strong benchmarks for evaluating novel approaches in similar contexts. Building upon the classic PSO proposed by Kennedy & Eberhart (1995), researchers have developed numerous enhancements, such as Bi-PSO and hybrid GA-PSO, to address the limitations of traditional algorithms. For instance, PSO is lauded for its simplicity and efficiency in handling continuous optimization problems (Zou et al., 2021). Bi-PSO, an extension of PSO, excels in managing discrete optimization challenges, particularly those involving binary decision variables (Gad, 2022; Saremi, 2018). GA, on the other hand, is known for its robust exploration of global search spaces through genetic operators like mutation and crossover. Hybrid GA-PSO combines the strengths of GA and PSO, balancing exploration and exploitation to effectively solve constrained multi-objective optimization problems (Kamel et al., 2020).

These algorithms collectively represent a diverse and well-established set of heuristic algorithms for solving complex MAPs. Their proven success and adaptability provide a strong basis for benchmarking new approaches. By comparing the Virtual Matching Algorithm (VMA) and its improved version (IVMA) with these representative algorithms, we aim to evaluate their relative performance and scalability in addressing multi-stage, multi-resource challenges. The comparison was conducted under consistent parameter settings to ensure a fair assessment of each algorithm's computational efficiency and solution quality. To evaluate performance across varying problem sizes, Case 1 focused on small-scale datasets, while Case 2 examined larger datasets.

Table 8 indicates that the proposed VMA and IVMA algorithms surpass the other algorithms in achieving the maximum satisfactory solution and reducing average CPU time, both under small (case 1) and large-scale data (case 2) conditions. The convergence curves are shown in Fig. 11. In Case 1, the IVMA achieved the highest fitness value of 1992.572, representing a 1.08 % improvement over GA-PSO (1970.987), a 1.02 % improvement over Bi-PSO (1972.280), a 1.44 % improvement over PSO (1963.898), and a 1.64 % improvement over GA (1959.816). Additionally, in terms of computational efficiency, the VMA completed the task in 3.41 s, making it 32.21 % faster than GA (5.03 s) and 42.88 % faster than GA-PSO (5.97 s). The IVMA further enhanced

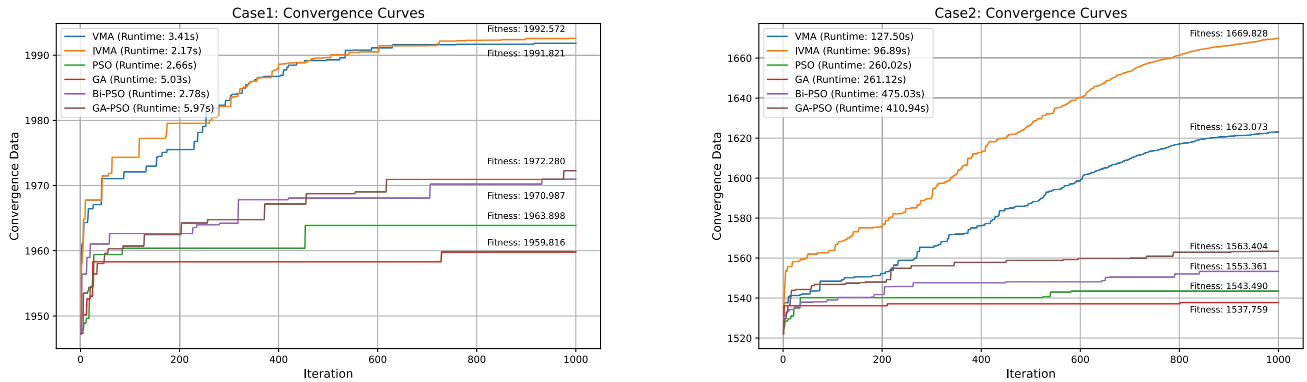

Fig. 11. Comparison for VMA, IVMA and GA, BI-PSO, PSO, GA-PSO.

efficiency, reducing runtime to just 2.17 s, which was 18.42 % faster than PSO (2.66 s) and 21.94 % faster than Bi-PSO (2.78 s).

Similarly, for Case 2, which involved large-scale datasets, the IVMA demonstrated more exceptional performance. The IVMA achieved the highest fitness score of 1669.828, representing a 2.80 % improvement over VMA (1623.073), a 6.37 % improvement over GA-PSO (1563.404), and a 6.97 % improvement over Bi-PSO (1553.361). Compared to PSO (1543.490) and GA (1537.759), the IVMA achieved 7.57 % and 7.91 % higher fitness values, respectively. Notably, the IVMA also significantly reduced computational time, completing the task in just 96.89 s, a 24.01 % reduction compared to VMA (127.50 s). When compared to PSO (260.02 s) and Bi-PSO (475.03 s), the IVMA demonstrated a 62.74 % and 79.60 % reduction in runtime, respectively. These results emphasize the superior solution quality and computational efficiency of the IVMA in addressing high-dimensional, large-scale problems.

7. Contributions and future work

7.1. Contributions

In this paper, we conduct a comprehensive analysis of assignment models with two-dimensional or three-dimensional features and their corresponding solving algorithms. Subsequently, we introduce the multi-dimensional assignment problems (MAPs) and the multi-dimensional assignment model (MAM), which incorporate multiple features as constraints. The MAM is developed as the 3D assignment model involving "executive agent-task agent-stage task." To address this model, we developed the virtual matching algorithm (VMA). This research makes significant contributions in the following areas:

(1) The proposed MAM is versatile and adaptable. Various features, including multi-agents, multi-stages, multi-resources, multi-constraints, are comprehensively integrated as different dimensions within the MAM models to align with real-world decision problems. This flexibility allows for the application of MAM in diverse fields, such as organizational assessment, production planning, resource scheduling, and more.

(2) The developed 3D assignment model serves as the fundamental MAM. It extensively addresses the performance evaluation between the executive agents and the task agents, and it can handle problems with time windows or multi-locations. Consequently, this paper utilizes the 3D assignment model in the context of organizational assessment and Guavas production to demonstrate and discuss the principles and effectiveness of MAM.

(3) The developed VMA draws inspiration from the fundamental concepts of traditional heuristic algorithms, namely GA and PSO, while first proposed the idea of dimensionality reduction for the MAM and incorporating roulette-wheel selection and variation factors. This approach realizes solving MAPs and ensures a balance between the impact of individual solutions and global solutions, thus preventing premature convergence to local optima. Consequently, the VMA effectively addresses N-dimensional assignment models.

(4) Compared to traditional PSO, VMA demonstrates the ability to achieve significantly better satisfactory solutions, as shown in Case 1. Conversely, the Improved Virtual Matching Algorithm (IVMA) excels in handling specialized MAPs scenarios involving large-scale repeated data within a single dimension. As shown in Case 2, IVMA significantly reduces solving time and improves overall efficiency.

7.2. Future work

This research aims to introduce the fundamental MAM and its corresponding heuristic algorithm to formally define and address prevalent MAPs. The methodology outlined in this paper is applicable in a wide range of decision problems featuring multi-feature constraints from varied fields, including healthcare, industrial production, economics, finance, and military decision-making, among others. In fact, N-dimensional assignment problems extending beyond three dimensions are not uncommon. In the introduction section of this paper, we introduced the concept of expanding from lower dimensions to higher dimensions using the examples of organizational assessment and guavas production cases. However, due to the limited length of the paper, we only present two real cases (organizational assessment and guavas production) to demonstrate the principles and effectiveness of MAM. Cases involving four-dimensional, five-dimensional, or even higher-dimensional problems will be the focus of our future research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper is supported by the Key Project on Teaching Reform Research in Ordinary Higher Education Institutions in Hunan Province, of China (HNJG20230582); the National Natural Science Foundation of China (71974090); the State Scholarship Fund (202108430098) from CSC; and the State Scholarship Fund (202208430061) from CSC. The authors would like to thank Krista Chen for contributing to this work.

Appendix 1

Appendix A Variable index (in alphabetical order).

Symbol	meaning	locations in the manuscript
$\underline{b_c^d}$	the minimum values of constraint functions in the Constraint Set c	Section 3.2
$\overline{b_c^d}$	the maximum values of constraint functions in the Constraint Set c	Section 3.2
$\overline{b_{t/s}}$	represents the maximum number of assignment that task agents can participate in each stage in 3D assignment model	Section 5.1.2
$\overline{b_{e/s}}$	represents the maximum number of assignment that executive agents can participate in each stage in 3D assignment model	Section 5.1.2
$\overline{b_t}$	represents the maximum number of assignment that task agents can participate in all stages in 3D assignment model	Section 5.1.2
$\overline{b_e}$	represents the maximum number of assignment that executive agents can participate in all stages in 3D assignment model	Section 5.1.2
c_1, c_2	attenuation coefficients	Section 4.2
$D = \{d_1, d_2, \cdots, d_n\}(n \geq 2, n \in N^*)$	a set of all dimensions	Section 3.1
$d_1, d_2, \cdots, d_k, d_n(n \geq 2, n \in N^*)$	represent every dimension in MAM, including executive agent, task agent, stage, etc.	Section 3.1
d_1, d_2, d_3	represent three dimensions in 3D assignment model	Section 5.1.2
e, t, s	represent different assigned individuals in the dimension 1, 2, 3. These are executive agents, task agents, and stages in 3D assignment model	Section 5.1.2
fit_{id}	a fitness of an individual solution sol_{id}	Section 4.2
fit_{id}^{new}	a fitness of a new individual solution sol_{id}^{new}	Section 4.2
fit_{id}^{max}	a historical optimal fitness of an individual solution sol_{id}	Section 4.2
fit_{gd}^{max}	a historical optimal fitness of a global solution *Solution*	Section 4.2
g_{max}	the maximum number of iterations	Section 4.2
h	a times of stage decay	Section 4.2
$i, i'', .., i^{(k)}, \cdots, i^{(n)}(n > k \geq 2, n \in N^*)$	different individuals in each dimension. And the $i_1^{(k)}$, $i_2^{(k)}$, … represent each individual in dimension k	Section 3.1
$j, k-1, k, k+1, l, \cdots, n$	used to list a dimension, such as d_j, d_k etc	Section 3.1 & 3.2
M_{jk}	a matching list which is a sequence of cross-combined individuals after dimensionality reduction of dimension j and k	Section 4.2
$N_{d_1} \cdots N_{d_k} \cdots N_{d_n}(n \geq k \geq 2, k \in N^*, n \in N^*)$	the cardinalities of values taken in dimensions $d_1, \cdots, d_n$ respectively.	Section 3.1
N_1, N_2, N_3	represent the cardinalities of the dimension. N_1 is number of executive agents, N_2 is number of task agents, and N_3 is number of stages in 3D assignment model	Section 5.1.2
$P_{d_1}^{i}, P_{d_2}^{i''}, \cdots, P_{d_k}^{i^{(k)}}, \cdots, P_{d_n}^{i^{(n)}}(n > k \geq 2, n \in N^*)$	the performances of different individuals $(i, i'', .., i^{(k)}, \cdots, i^{(n)})$ in each dimension $(d_1, d_2, \cdots, d_k, d_n)$.	Section 3.1
$P_{d_1, \cdots, d_n}^{i, \cdots, i^{(n)}} i \in (1, N_{d_1}), \cdots, i^{(n)} \in (1, N_{d_n})$	the performance of the assignment that was made of multiple dimensions. $i, \cdots, i^{(n)}$ is an individual on the $d_1, \cdots, d_n$ dimension. $i, \cdots, i^{(n)}$ corresponds to dimension $d_1, \cdots, d_n$ different individuals.	Section 3.1
P_{TE}	represents a performance (costs or benefits) of assignments by different task agents (T) and executive agents (E) in the performance matrix of 2D assignment model	Section 1
$P_{TES}(x_1, x_2, x_3)$	The third-dimension stage is introduced based on P_{TE}, and it represents a performance of different individuals x_1, x_2 and x_3 in 3D assignment model	Section 1
P_1^e, P_2^t, P_3^s	represent the performance of different assigned individuals (e, t, s) in dimension d_1, d_2, d_3 of 3D assignment model	Section 5.1.2
P_{123}^{ets}	the performance of an assignment executed by individuals i, j, k in 3D assignment model, and $P_{123}^{ets} = \alpha \times P_1^e + \beta \times P_2^t + \gamma \times P_3^s, \forall s = 1 P_{123}^{ets} = \alpha \times P_1^e + \beta \times P_2^t + \gamma \times P_3^s + \delta \times P_{123}^{et(s-1)}, \forall s > 1, s \in N^*$	Section 5.1.2
pr	a probability of virtual offset list, and $pr = r_1 \times \mathrm{rand}(V + \eta) + \lambda \times c_1 \times r_2 \times rand\left(\frac{fit_{id}^{max}}{\sum_{id=1}^{nPop} fit_{id}^{max}}\right) + \varepsilon \times c_2 \times r_3 \times rand\left(\frac{fit_{gd}^{max}}{\sum_{id=1}^{nPop} fit_{gd}^{max}}\right)$	Section 4.2
Q_{d_k}	the number of different kinds of one dimension k	Section 4.3
q	a class of individual in one dimension	Section 4.3
R_{TE}	represents relationships of different task agents (T) and executive agents (E) in 2D assignment model	Section 1
r_1, r_2, r_3	equilibrium factors	Section 4.2
$Solution = [sol_1, \cdots, sol_u]$	an initial solution set consisting of u feasible solutions.	Section 4.2
$sol_{id} = (x_{i, \cdots, i^{(n)}}, \cdots)$	a solution vector in solution set *Solution*	Section 4.2
$Solution^{new}$	a new feasible solution set	Section 4.2
$sol_{id}^{new} = (x_{i, \cdots, i^{(n)}}, \cdots)$	a new feasible solutiona new solution vector in the solution set $Solution^{new}$	Section 4.2
u	count of feasible solutions in the initial solution set	Section 4.2
$V = [v_1, \cdots, v_k, \cdots, v_n]$	a virtual matching list. v_k is a virtual individual.	Section 4.2
$x_{i, \cdots, i^{(n)}}$	0–1 decision variable, and $x_{i, \cdots, i^{(n)}} = 1$ if individuals $i^{(k)}$ (for $k = 1, 2, \ldots, n$) are assigned together otherwise it equals 0.	Section 3.1
$w_q, q \in (1, Q_{d_k})$	the capacity of different classifications q of one dimension k	Section 4.3
I,II,III,IV,…,x, …,$N-1$	used to list a Constraint Set, represent Constraint Set c (for x = I,II,III,IV,…,$N-1$)	Section 3.2
α	assessment ability index of executive agents	Section 5.1.2
β	learning ability index of task agents	Section 5.1.2
γ	weight values for performance of stages	Section 5.1.2
δ	the effect of the executive agents and task agents in the previous stage.	Section 5.1.2
ε	a variation factor of global solution	Section 4.2
η	an inertia coefficient	Section 4.2
θ	a correction factor in objective function	Section 3.1
λ	a variation factor of individual solution	Section 4.2

Appendix B. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eswa.2024.126369.

Data availability

Data will be made available on request.

References

Anowar, F., Sadaoui, S., & Selim, B. (2021). Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Computer Science Review, 40*, Article 100378.

Bi, J., Wang, Z., Yuan, H., Zhang, J., & Zhou, M. (2023). Self-adaptive teaching-learning-based optimizer with improved RBF and sparse autoencoder for high-dimensional problems. *Information Sciences, 630*, 463–481.

Bansal, N., Correa, J. R., Kenyon, C., & Sviridenko, M. (2006). Bin packing in multiple dimensions: Inapproximability results and approximation schemes. *Mathematics of operations research, 31*(1), 31–49.

Bellman, R. (1966). Dynamic programming. *Science, 153*(3731), 34–37.

Chaosheng, Z. H. A. N. G., Jian, W. A. N. G., Zhengmin, L. I., & Xin, Y. A. N. G. (2019). A solider-equipment-task assignment model for combat engineering support operation and its solution. *Acta Armamentarii, 40*(7), 1511.

Deng, W., Xu, J., Zhao, H., & Song, Y. (2020). A novel gate resource allocation method using improved PSO-based QEA. *IEEE Transactions on Intelligent Transportation Systems, 23*(3), 1737–1745.

Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: A systematic review. *Archives of Computational Methods in Engineering, 29*(5), 2531–2561.

Garg, H. (2016). A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation, 274*, 292–305.

Ghoseiri, K., & Ghannadpour, S. F. (2010). A hybrid genetic algorithm for multi-depot homogenous locomotive assignment with time windows. *Applied Soft Computing, 10*(1), 53–65.

Bastani, H., & Bayati, M. (2019). Online decision making with high-dimensional covariates. *Operations Research, 68*(1), 276–294.

Harbaoui Dridi, I., Ben Alaïa, E., Borne, P., & Bouchriha, H. (2020). Optimisation of the multi-depots pick-up and delivery problems with time windows and multi-vehicles using PSO algorithm. *International Journal of Production Research, 58*(14), 4201–4214.

Sherali, H. D., & Zhu, X. (2008). Two-stage fleet assignment model considering stochastic passenger demands. *Operations Research, 56*(2), 383–399.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology, 24*(6), 417.

Hahn, P. M., Kim, B. J., Stuetzle, T., Kanthak, S., Hightower, W. L., Samra, H., & Guignard, M. (2008). The quadratic three-dimensional assignment problem: Exact and approximate solution methods. *European Journal of Operational Research, 184*(2), 416–428.

Côté, J.-F., Gendreau, M., & Potvin, J.-Y. (2020). The vehicle routing problem with stochastic two-dimensional items. *Transportation Science, 54*(2), 453–469.

Jia, Z., Yu, J., Ai, X., Xu, X., & Yang, D. (2018). Cooperative multiple task assignment problem with stochastic velocities and time windows for heterogeneous unmanned aerial vehicles using a genetic algorithm. *Aerospace Science and Technology, 76*, 112–125.

Kammerdiner, A., Semenov, A., & Pasiliao, E. L. (2022). Multidimensional Assignment Problem for multipartite entity resolution. *Journal of Global Optimization, 84*(2), 491–523.

Keogh, E., & Mueen, A. (2010). Curse of Dimensionality. *Encyclopedia of Machine Learning*. Ed. by C. Sammut and GI Webb.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika, 29*(1), 1–27.

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.

Karp, R. M. (2010). *Reducibility among combinatorial problems* (pp. 219–241). Berlin Heidelberg: Springer.

Kamel, M. A., Yu, X., & Zhang, Y. (2020). Real-time fault-tolerant formation control of multiple WMRs based on hybrid GA–PSO algorithm. *IEEE Transactions on Automation Science and Engineering, 18*(3), 1263–1276.

Karsu, Ö., Azizoğlu, M., & Alanlı, K. (2021). Exact and heuristic solution approaches for the airport gate assignment problem. *Omega, 103*, Article 102422.

Karsu, Ö., & Azizoğlu, M. (2012). The multi-resource agent bottleneck generalised assignment problem. *International Journal of Production Research, 50*(2), 309–324.

Lipowski, A., & Lipowska, D. (2012). Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications, 391*(6), 2193–2196.

Li, F., Xu, Z., & Li, H. (2021). A multi-agent based cooperative approach to decentralized multi-project scheduling and resource allocation. *Computers & Industrial Engineering, 151*, Article 106961.

Liu, L., & Gao, X. (2009). Fuzzy weighted equilibrium multi-job assignment problem and genetic algorithm. *Applied Mathematical Modelling, 33*(10), 3926–3935.

Li, M., Liu, C., Li, K., Liao, X., & Li, K. (2020). Multi-task allocation with an optimized quantum particle swarm method. *Applied Soft Computing, 96*, Article 106603.

Li, Y., Chen, Y., Zhong, J., & Huang, Z. (2019). Niching particle swarm optimization with equilibrium factor for multi-modal optimization. *Information Sciences, 494*, 233–246.

Mozafar, M. R., Moradi, M. H., & Amini, M. H. (2017). A simultaneous approach for optimal allocation of renewable energy sources and electric vehicle charging stations in smart grids based on improved GA-PSO algorithm. *Sustainable Cities and Society, 32*, 627–637.

Ma, Z., Yang, X., Wu, J., Chen, A., Wei, Y., & Gao, Z. (2022). Measuring the resilience of an urban rail transit network: A multi-dimensional evaluation model. *Transport Policy, 129*, 38–50.

Pierskalla, W. P. (1968). The multidimensional assignment problem. *Operations Research, 16*(2), 422–431.

Robenek, T., Umang, N., Bierlaire, M., & Ropke, S. (2014). A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports. *European Journal of Operational Research, 235*(2), 399–411.

Soman, J. T., & Patil, R. J. (2023). Two-dimensional loading in vehicle routing problem with release and due dates. *Expert Systems with Applications, 232*, Article 120862.

Shen, J., Wang, P., Dong, H., Wang, W., & Li, J. (2024). Surrogate-assisted evolutionary algorithm with decomposition-based local learning for high-dimensional multi-objective optimization. *Expert Systems with Applications, 240*, Article 122575.

Samanta, S., Chakraborty, D., & Jana, D. K. (2024). Neutrosophic multi-period two stage four-dimensional transportation problem for breakable items. *Expert Systems with Applications, 246*, Article 123266.

Saremi, S., Mirjalili, S., Lewis, A., Liew, A. W. C., & Dong, J. S. (2018). Enhanced multi-objective particle swarm optimisation for estimating hand postures. *Knowledge-Based Systems, 158*, 175–195.

Tao, J., Yu, Z., Zhang, R., & Gao, F. (2021). RBF neural network modeling approach using PCA based LM–GA optimization for coke furnace system. *Applied Soft Computing, 111*, Article 107691.

Walteros, J. L., Vogiatzis, C., Pasiliao, E. L., & Pardalos, P. M. (2014). Integer programming models for the multidimensional assignment problem with star costs. *European Journal of Operational Research, 235*(3), 553–568.

Wang, Q. (2021). Knowledge-based approach for dimensionality reduction solving repetitive combinatorial optimization problems. *Expert Systems with Applications, 184*, Article 115502.

Wang, R., Dong, N. J., Gong, D. W., Zhou, Z. B., Cheng, S., Wu, G. H., & Wang, L. (2021). PCA-assisted reproduction for continuous multi-objective optimization with complicated Pareto optimal set. *Swarm and Evolutionary Computation, 60*, Article 100795.

Yu, G. F., Fei, W., & Li, D. F. (2018). A compromise-typed variable weight decision method for hybrid multiattribute decision making. *IEEE Transactions on Fuzzy Systems, 27*(5), 861–872.

Yang, L., Gao, Z., & Li, K. (2011). Railway freight transportation planning with mixed uncertainty of randomness and fuzziness. *Applied Soft Computing, 11*(1), 778–792.

Zhang, H., Liu, F., Zhou, Y., & Zhang, Z. (2020). A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem. *Information Sciences, 539*, 347–374.

Zhou, S., Li, D., & Yin, Y. (2021). Coordinated appointment scheduling with multiple providers and patient-and-physician matching cost in specialty care. *Omega, 101*, Article 102285.

Zou, X., Fang, Z., & Xiong, S. (2021). A discrete particle swarm optimization method for assignment of supermarket resources to urban residential communities under the situation of epidemic control. *Applied Soft Computing, 98*, Article 106832.