International Journal of Industrial Engineering Computations 15 (2024) 41–58

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

# A hybrid genetic algorithm with variable neighborhood search for batch dispersion problem to improve traceability

**Minglun Ren[a,b] and Gang Wang[a*]**

[a]*School of Management, Hefei University of Technology, Hefei, 230009 China*
[b]*Key Laboratory of Process Optimization and Intelligent Decision-making of Ministry of Education, Hefei, 230009 China*

C H R O N I C L E

*Article history*:
Received July 28 2023
Received in Revised Format October 12 2023
Accepted December 4 2023
Available online December 4 2023

Keywords:
*Batch dispersion*
*Mixed integer programming*
*Hybrid heuristic algorithms*
*Traceability*
*Discrete manufacturing*

A B S T R A C T

Batch dispersion problem (BDP) restricts batch traceability in large-scale discrete production and negatively impacts batch recall costs. However, previous research has ignored the complexity of the BDP in their analyses. This paper investigates the BDP under the composed bill of materials (BOM) and develops a mathematical model for the BDP with the goal of minimizing the total batch dispersion by utilizing the batch dispersion as a measure of the degree of dispersed usage of part batches. BDP-GAVNS, a hybrid genetic algorithm with variable neighborhood search, is devised for the BDP based on the demonstration that the BDP is an NPC problem. In BDP-GAVNS, memory banks were introduced to increase the diversity of individuals performing crossover operations. Additionally, the encoding method and infeasible solution repair program are designed according to the characteristics of BDP. Numerical experiments validate the viability and effectiveness of BDP-GAVNS in solving BDP. They demonstrate that (1) the optimal combination occurs when the ratio of individuals produced by the three types of population initialization methods, namely global selection (GS), local selection (LS), and random selection (RS), to the population takes values of 0.30, 0.10, and 0.60, respectively; (2) The memory bank enriches the source of individuals required for crossover operations and improves the performance of crossover operations; and (3) The BDP-GAVNS is more effective than the other five heuristic algorithms including genetic algorithms in seeking the optimal solution of BDP.



## 1. Introduction

Traceability of batch impacts not only product quality and the ability to manage the reverse supply chain, but also the manufacturer's brand reputation and consumer confidence in large-scale discrete manufacturing. Existing approaches to enhance batch traceability include the development of traceability systems (Li et al., 2010; Pierini et al., 2016), batch correlation (Van der Spiegel et al., 2013; Zhao et al., 2020), and the application of statistical data models (Zhao et al., 2020), etc. Traceability systems record product composition and location information by tracking the production process, and batch association aims at realizing the recording of information between a batch and its production elements, such as production equipment and people. Compared with traditional traceability methods, such as manually recorded product process cards, traceability systems and batch association improve the traceability accuracy and efficiency of batches in large-scale discrete manufacturing industries, but it is difficult to reduce the number of batch recalls. Statistical data modeling can generally only enhance the traceability of products in process manufacturing, but it is also difficult to reduce the number of batch recalls. The BDP is a frequent issue that affects batch traceability. It is defined by the scattered use of part batches in numerous product batches, which not only limits batch traceability but also raises the cost of recalls. Both Dupuy et al. (2005) and Maity et al. (2021) used batch dispersion to measure the extent of dispersed use of part batches in the food industry and solved the BDP using conventional exact solution methods, however, because the BDP is an NPC problem, their exact solution methods cannot be used to resolve complex problems.

* Corresponding author
E-mail: 2019010076@mail.hfut.edu.cn (G. Wang)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2024 Growing Science Ltd.
doi: 10.5267/j.ijiec.2023.12.002

This paper aims to investigate the issue of batch dispersion in large-scale discrete manufacturing, specifically focusing on the production and assembly process of automobile bodies. The degree of dispersed use of automobile door batches is measured using batch dispersion, which refers to the number of product batches that contain a specific component batch. The findings reveal that the high total batch dispersion is primarily caused by the arbitrary matching combination method employed between automobile door batches and body batches during the assembly process. Consequently, efforts are made to optimize this matching combination method during the production planning stage. The complexity of this optimization problem can be classified as non-deterministic polynomial time complete (NPC) due to the exponential growth of the matching combination approach with the increasing number of batches. In light of the synergistic relationship between genetic algorithms (GA) and variable neighborhood search (VNS) algorithms, which offer complementary global and local search capabilities in the solution space, and their demonstrated efficacy in addressing intricate problems (Wang et al., 2023), a novel hybrid algorithm, referred to as BDP-GAVNS, is proposed for the BDP. This work presents a comprehensive set of tests that substantiate the efficacy of the BDP-GAVNS in efficiently solving the BDP.

Compared with previous studies, the main contributions of this paper are summarized as follows:

(1) The BDP under the assembled bill of materials (BOM) is explored, and when batch dispersion is used to measure the degree of dispersed use of part batches and the number of recalls is used to evaluate the traceability performance, it is found that the arbitrary matching combination between part batches and product batches leads to BDP, which is not conducive to shrinking the number of recalls and improving the traceability.
(2) Proved that BDP under the assembly BOM is an NPC problem and develop a mathematical model for the BDP with the objective of minimizing the total batch dispersion by using the batch dispersion as a measure of the degree of dispersed usage of part batches.
(3) BDP-GAVNS is devised for solve the BDP. Regarding algorithm optimization strategy, BDP-GAVNS chooses to embed the VNS into the improved GA and builds the memory bank optimization strategy (MBOS) to increase the diversity of individuals performing crossover operations. Regarding algorithm improvement, Encoding and infeasible solution repair strategies are designed based on the characteristics of BDP. Additionally, three types of population initialization methods and four types of neighborhood structures are designed to improve the search capability of BDP-GAVNS.
(4) The feasibility and validity of the MBOS of BDP-GAVNS was verified using the control variable method, and the proportion of individuals generated by the three types of population initialization methods, GS, LS and RS, to the overall population was set to be 0.1, 0.3, and 0.6, respectively, using full-factorial experiments. The superiority of BDP-GAVNS in solving the BDP problem is verified by comparing BDP-GAVNS with five other heuristic algorithms including genetic algorithm.

The paper is structured as follows. Section 2 provides a literature review on traceability and hybrid optimization algorithms. Section 3 describes and builds a mathematical model for BDP, followed by proves that BDP is an NPC problem. Section 4 describes the BDP-GAVNS's algorithmic optimization strategy, genetic algorithm improvements, and neighborhood structure. The numerical experiment section is found in Section 5. Section 6 contains the discussion part of the paper.

## 2. Literature Review

### 2.1 Traceability

This section summarizes the literature on traceability research, including the concept of traceability, the challenges of enhancing traceability, the methods of achieving traceability, and the performance evaluation of traceability. The ISO 8402 standard defines traceability as the ability to obtain information about an entity's history, location, and application through effective recording methods (*ISO*, 1995). Internal traceability is limited to within the organization, whereas external traceability is primarily concerned with changes in material information at the supply chain level (Moe, 1998). To achieve internal traceability, it is necessary to record the entire production and processing information of a part order, including personnel and processing-related equipment. External traceability, on the other hand, necessitates attention to information regarding the transformation of material batches across supply chain nodes, including production, transportation, and storage information (Badia-Melis et al., 2015). Trace and track are the two means by which traceability can be attained. Trace focuses on the ability to match the query criteria and product characteristics with their source and identifies the fundamental cause of a quality issue. Track emphasizes the ability to localize a material based on its characteristics (Astill et al., 2019).

Regarding research on the difficulties of enhancing traceability, numerous works have attempted to assist manufacturing companies in enhancing traceability through the development of traceability systems (Chen et al., 2019; Qian et al., 2020). While the internal difficulties in constructing a traceability system stem primarily from the diversity of parts types in the production process, the phenomenon of dispersed use of parts batches, the change in physical shapes of parts, and the complex associations between parts batches and factors such as people and equipment (Qian et al., 2022), the external difficulties stem primarily from the difficulty of sharing and collaborating production information between organizations. Traceability can be improved by incorporating a traceable resource unit (TRU) into the traceability system (Fan et al., 2019), which must meet uniqueness and identifiability requirements (Olsen & Borit, 2013). Batches are commonly used traceability units, and batch is a management concept that seeks to distinguish the same product from various suppliers (Aung & Chang, 2014). batch dispersion refers to the number of parts from the same batch that are used in different batches of the product; batch dispersion can characterize the dispersed use of parts batches; batch dispersion reduces product traceability and is a significant challenge for the construction of traceability systems (Dupuy et al., 2005).

In terms of research on methods to achieve traceability, the production and processing characteristics of different products are different, and the methods to enhance traceability are also different. For example, in order to enhance the traceability of the whole food supply chain process, Liang et al. (2019, 2012) added small food-grade tracers to the grains to be purchased and realized the effective traceability of the food products. Sardina et al. (2015) utilized the irreproducibility of DNA to successfully identify the breed information of goats and constructed a genetic traceability system of goat breeds. Comba et al. (2013) successfully identified the breed information of goats by defining "ingredient distance" and constructed a genetic traceability system of goat breeds. With the development of information technology, radio-frequency identification (RFID) can effectively enhance the traceability of materials (Luvisi et al., 2012; Yang et al., 2016), but the method will no longer be applicable when undergoing processes that change the physical shape of materials such as cutting.

In studies evaluating the efficacy of traceability, the number of product recalls and the cost of recalls are two crucial indicators. The values of these two metrics are dependent on the batch size, decentralized use, and traceability method (Dabbene et al., 2014). Dupuy et al. (2005) estimated the cost of product recall using the batch dispersion of the production system, which they defined as the sum of the batch dispersion of all material batches in the system. In addition, Dabbene and Gay (2011) defined the worst-case recall cost (WCRC) and average recall cost (ARC) in order to quantify the number of products that must be recalled and the associated costs in exceptional circumstances. Based on a synthesis of extant evaluation metrics for traceability performance, Qian et al. (2017) created a quantifiable granularity traceability model. The model consists of two layers, the first of which considers the precision, depth, and breadth of traceability simultaneously, and the second of which consists of seven quantifiable performance evaluation indices for traceability, such as forward and backward tracking distance, information update frequency, etc. Based on the hierarchical analysis method, the model enables the evaluation of traceability performance.

Through the review of traceability-related research literature, we find that although new information technologies, such as radio frequency identification technology, blockchain technology and big data quality prediction technology, have been incorporated into the traceability system and widely used, However, these techniques are only concerned with traceability during the production process and after-sales service phases of the product, whereas the solution to the BDP problem requires optimization of the dispersed use of parts batches during the production planning stage.

### *2.2 Hybrid optimization algorithms*

Hybrid optimization algorithms exploit the benefits of individual algorithms and overcome their limitations by combining multiple heuristic algorithms, thereby enhancing the stability, convergence speed, solution accuracy, and robustness of the algorithms (Mokarram et al., 2019; Zhang et al., 2019). Currently, Several disciplines and sectors currently utilize hybrid optimization algorithms. Hybrids of genetic algorithm and local search algorithm, particle swarm optimization algorithm and simulated annealing algorithm, simulated annealing and forbidden search algorithm, and artificial immune algorithm and other algorithms are common hybrid optimization algorithms (Khalilpourazari & Khalilpourazary, 2019). Genetic algorithm is an optimization algorithm based on natural selection and genetic mechanisms that can be used to find optimal solutions to complex problems (Katoch et al., 2021a). However, due to its stochastic nature, local optimal solutions may be found (Garg, 2016). Genetic algorithms can be combined with local search algorithms to address this problem. In this method, the genetic algorithm is responsible for generating the initial population and conducting a population-wide search. Afterwards, a local search algorithm is used to locally optimize each new solution, and combining the two can accelerate convergence while enhancing solution quality (Deng et al., 2017; Slowik & Kwasnicka, 2020). Informed by the aforementioned research literature, this paper proposes a hybrid genetic algorithm (BDP-GAVNS) for BDP, which exploits the characteristics of both genetic algorithms and variable neighborhood search algorithms by taking advantage of their respective strong abilities in global and local searches and avoids the inherent limitations of individual algorithms by combining the two algorithms organically in terms of functionality.

## 3. Problem descriptions and complexity analysis

This section first explores the BDP of part batches under the bill of materials for assembly, and when batch dispersion is used to measure the degree of dispersed use of part batches and the number of recalls is used to evaluate the traceability performance, it is found that the arbitrary matching combination between part batches and product batches leads to BDP, which is not conducive to shrinking the number of recalls and improving the traceability. Next, proved that BDP under the assembly bill of materials is an NPC problem, and develop a mathematical model for the BDP with the objective of minimizing the total batch dispersion by using the batch dispersion as a measure of the degree of dispersed usage of part batches.

### *3.1 Problem descriptions*

The air supply system (ASS) is one of the seven systems of an engine that provides the engine with clean air appropriate to the engine load and enables the formation of a combustible mixture (Roberts and Brooks, 2014). The air supply system consists of four main parts (see Fig. 1), and when a part fails due to quality issues, it not only poses a threat to passenger safety, but will also force the automaker to recall the product, which will result in huge costs and reputational damage.

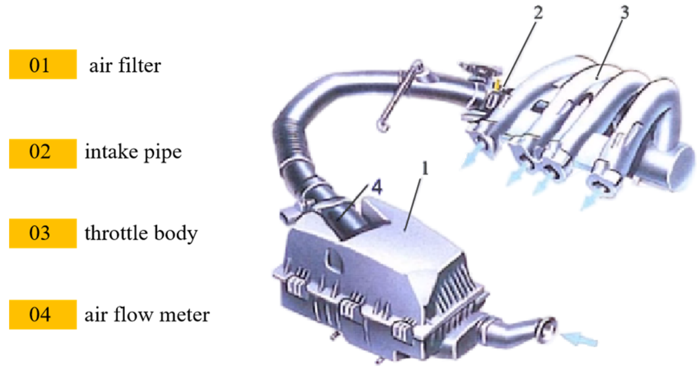



**Fig.1.** Engine air supply system parts diagram

During the assembly process of ASS, automobile companies will manage the parts in the form of batches, i.e., the batch list is used to record the product manufacturing batch and the parts batch to improve the recall precision and efficiency (Liu and Hu, 2007). Figure 2 shows an assembly list diagram of the ASS, and the combination of numbers and English strings indicates the batch number, e.g., "109J00000" is the production batch number of the ASS. The numbers in parentheses indicate the quantity of each part required for a single ASS.

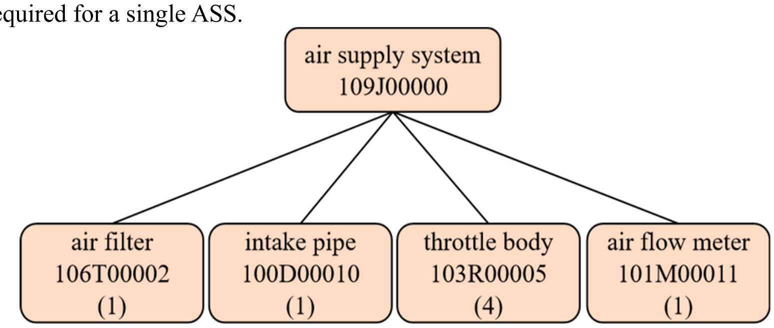



**Fig. 2.** Air supply system batch list diagram

To better illustrate the lot dispersion problem without losing generality, we will only discuss the batch dispersion problem between the air supply system (product) and the air filter (part). Assume that a car company has three batches of parts in stock, each batch identification and the number of contained respectively 106T00002 has 2 pieces, 106T00003 has 3 pieces, 106T00004 has 3 pieces. The car company's existing product batch orders are as follows: 109J00000 requires 3 pieces, 109J00001 requires 2 pieces, and 109J00002 requires 3 pieces (see part a in Fig. 3). The ratio of product to part requirements is 1:1.

The batch dispersion of a part batch is the number of product batches that contain the part batch (Dupuy et al., 2002). When the vehicle manufacturer assembles according to the first-in-first-out (FIFO) collocation of part batches, the batch dispersion of 106T00002, 106T00003, and 106T00004 is 1, 2, and 1, respectively, and the total batch dispersion of the system is 4. The black solid line in part a of Fig. 3 identifies the final use of each air filter batch in each air supply system batch under the FIFO mode (corresponding to the part b1 in Fig. 3).

When assembled in accordance with the LIFO (corresponding to part b2 in Fig. 3), the batch dispersion of 106T00002, 106T00003 and 106T00004 is 1, 2 and 1 respectively, and the total batch dispersion of the system is 4. By reasonably adjusting the collation method to change the matching combination of the part batch and the product batch to further reduce the batch dispersion, it will realize the purpose of reducing the number of batches recalls and improving the traceability. The total batch dispersion of the system can be further reduced by rationally adjusting the receiving method to change the matching combination between part batches and product batches to reduce the total batch dispersion of the system.

The total batch dispersion of the system can be reduced by adjusting the matching method in FIFO mode (corresponding to part a and part b1 in Fig. 3), which is adjusted as follows: we match 106T00002 with 109J00001, 106T00003 with 109J00000, and 106T00004 with 109J00002 The whole adjustment process is shown as the orange dotted line in part a of Fig. 3. After adjustment, the dispersion of each component batch is 1, and the total batch dispersion of the system is reduced to 3 (as shown in part b3 in Fig. 4). At this point, only lot 109J00000 needs to be recalled when 106T00003 has a problem. Compared to the FIFO and LIFO modes, this adjustment achieves the purpose of reducing the number of recalled lots.

In fact, in this production scenario, each product batch has three-part batches to choose from, so there are $C_3^1 \times C_3^1 \times C_3^1 = 27$ combinations of product batches and part batches. By constantly adjusting the way the part batches are used, it is always possible to find the combination that minimizes the batch dispersion of the system. Therefore, the batch dispersion problem refers to the problem of minimizing the total batch dispersion of the system by continuously adjusting the number of part batches used in a product batch given the number of product batches K, the number of products contained in a single product batch $Q_{FP}(k)$, the number of part batches S, the number of parts contained in a single part batch $Q_{SP}(s)$, and the number of parts demanded by the product QD.

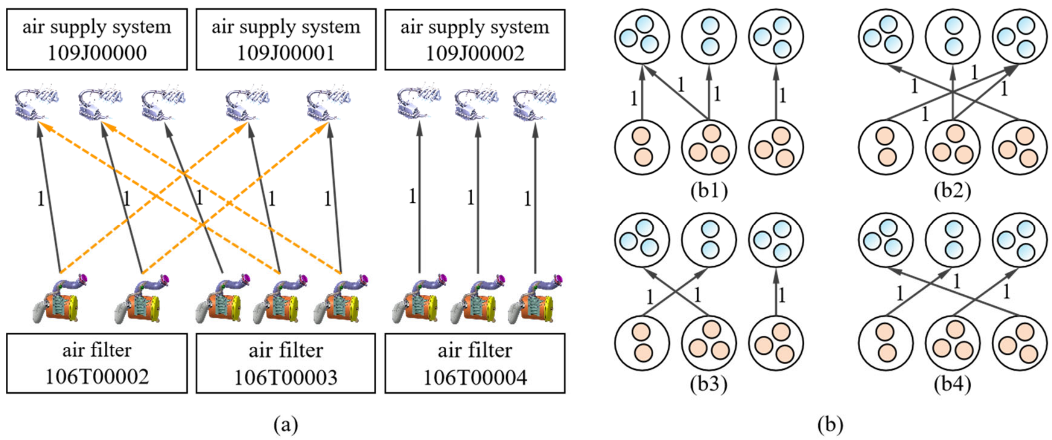



**Fig.3.** Schematic diagram of the BDP in a simplified production situation

Simplified production situations are conducive to the elaboration and understanding of the batch dispersion problem in the assembly of air supply systems, but in real production situations, the number of product batches and the number of required part batches assembled in a working day are at least several dozens of batches, and the combination of product batches and part batches rises exponentially with the growth of batch sizes, and it is difficult to solve the batch dispersion problem in a large-scale situation by means of exact algorithms, such as enumeration, and so it is necessary to design specialized heuristic algorithms for the solution of this problem.

**Table 1**
The notations list

| ***Notations*** | ***Definitions*** |
|---|---|
| ***Data*** | |
| $K$ | Number of batches to be produced, i.e. order quantity |
| $I$ | Number of batches of existing parts |
| $J$ | Number of batches of parts to be outsourced |
| $S$ | Total number of part batches, $S = I + J$ |
| $QD$ | Number of parts required for a single product |
| $Q_{FP}(k)$ | Quantity to be produced for product batch $k$, $\forall k = 1, \cdots K$ |
| $Q_{EP}(i)$ | Number of parts contained in existing part batch $i$, $\forall i = 1, \cdots I$ |
| $Q_{BP}(j)$ | Number of parts included in purchased parts batch $j$, $\forall j = 1, \cdots J$ |
| $Q_{SP}(s)$ | Number of parts included in parts batch $s$, $\forall s = 1, \cdots S$ |
| $k\_list$ | List of product batches and the number of products included in each batch, $k\_list$=[$Q_{FP}(1), \cdots Q_{FP}(k)$] |
| $s\_list$ | List of part batches and number of parts included in each batch, $s\_list$=[$Q_{SP}(1), \cdots Q_{SP}(s)$] |
| $Vhv$ | Very high value |
| ***Variables*** | |
| $x_{MP}(i, k)$ | variable equal to 1 if existing part batch $i$ is used in product batch $k$, 0 otherwise |
| $x_{BP}(j, k)$ | variable equal to 1 if purchased part batch $j$ is used in product batch $k$, 0 otherwise |
| $x_{SP}(s, k)$ | variable equal to 1 if part batch $s$ is used in product batch $k$, 0 otherwise |
| $Q_{MP}(i, k)$ | Quantity of existing part batch $i$ used in product batch $k$ |
| $Q_{BP}(j, k)$ | Quantity of purchased parts batch $j$ used in product batch $k$ |
| $Q_{SP}(s, k)$ | Quantity of part batch $s$ used in product batch $k$ |

**Table 2**
Mathematical model

| | |
|---|---|
| $Minimize\ \ Y = \sum_{i=1}^{I}\sum_{k=1}^{K} x_{MP}(i,k) + \sum_{j=1}^{J}\sum_{k=1}^{K} x_{BP}(j,k)$ | (1) |
| $QD \times Q_{FP}(k) = \sum_{i=1}^{I} Q_{MP}(i,k) + \sum_{j=1}^{J} Q_{BP}(j,k) \quad \forall k = 1, \cdots K$ | (2) |
| $\sum_{k=1}^{K} Q_{MP}(i,k) \times QD \le Q_{EP}(i) \quad \forall i = 1, \cdots I$ | (3) |
| $\sum_{k=1}^{K} Q_{BP}(j,k) \times QD \le Q_{BP}(j) \quad \forall j = 1, \cdots J$ | (4) |
| $x_{MP}(i,k) \le Q_{MP}(i,k) \quad \forall i = 1, \cdots I \quad \forall k = 1, \cdots K$ | (5) |
| $Q_{MP}(i,k) \le x_{MP}(i,k) \times Vhv \quad \forall i = 1, \cdots I \quad \forall k = 1, \cdots K$ | (6) |
| $x_{BP}(j,k) \le Q_{BP}(j,k) \quad \forall j = 1, \cdots J \quad \forall k = 1, \cdots K$ | (7) |
| $Q_{BP}(j,k) \le x_{BP}(j,k) \times Vhv \quad \forall j = 1, \cdots J \quad \forall k = 1, \cdots K$ | (8) |

*3.3 Complexity analysis*

**Proof 1: BDP is an NP problem.**

Suppose $K = 1$, $S = 5$, $QD = 1, Q_{FP}(1) \times QD \le \sum_{s=1}^{5} Q_{SP}(s)$, without considering supply and demand constraints, when $\sum_{s=1}^{5} x_{SP}(s,1) = 0, s = 1, \cdots 5$, The number of part batches used to produce the product can be combined up to $C_5^0$ ways, when $\sum_{s=1}^{5} x_{SP}(s,1)$takes the values 1、2、3、4 and 5, respectively，The number of part batches used to produce the product can be combined up to $C_5^1$, $C_5^2$, $C_5^3$, $C_5^4$ and $C_5^5$ respectively. Therefore, Without considering supply and demand constraints, part batches can be combined in as many as $C_5^0 + C_5^1 + C_5^2 + C_5^3 + C_5^4 + C_5^5 = 2^5$=32 ways in order to produce a product。When the number of part batches is small, the enumeration method can be used to enumerate all the combinations and then find the optimal production method for the combination of part batches, but when the values of $K$ and $S$ are gradually increasing, The maximum number of combinations of parts to be enumerated is up to $C_n^0 + C_n^1 + C_n^2 + \cdots + C_n^n = 2^n$, That is, the number of combinations of part batches grows exponentially with $K$ and $S$ and is difficult to solve in polynomial time. Therefore, the BDP under the assembly BOM is an NP problem.

**Proof 2: BDP is an NPC problem.**

The 0-1 knapsack problem is an NPC problem in which the number of items $i$, the weight $w_i$ and value $v_i$ of each item, and the maximum knapsack load $C$ are known, and $x_i$ is used to denote whether or not to load the $i$th item into the knapsack in such a way as to maximize the value of the items in the knapsack without violating the constraints on the knapsack's load capacity. The mathematical model of the 0-1 knapsack problem is as follows:

$$max\ \ Y = \sum_{i=1}^{I} v_i x_i$$

$$\sum_{i=1}^{I} w_i x_i \le C$$

$$x_i \in \{0,1\}, i = 1,2, \cdots I$$

Assume that $K = 1, J = 0, QD = 1, Q_{FP}(1) \times QD \le \sum_{s=1}^{S} Q_{SP}(s)$. In this case, the BDP can be described as based on the known number of part batches $S$, the number of parts $Q_{SP}(s)$ contained in each part batch $s$, and the number of parts required by the product batch $Q_{FP}(1) \times QD$, and using $x_s$ to indicate whether or not to use the $s$th batch of parts in the product batch, such that without violating the supply-demand constraints, the total batch dispersion is minimized. The simplified mathematical model of BDP is as follows:

$$min\ \ Y = \sum_{s=1}^{S} x_s$$

$$Q_{SP}(s,1) \le Q_{SP}(s)$$

$$QD \times Q_{FP}(1) = \sum_{s=1}^{S} Q_{SP}(s,1)$$

$$Q_{SP}(s,1) \le x_{SP}(s,1) \times Vhv$$

$$x_{SP}(s,1) \le Q_{SP}(s,1)$$

$$x_s \in \{0,1\}, s = 1,2, \cdots S$$

We equate the maximum loading of the backpack in the 0-1 backpack problem to the number of parts demanded by the product in the BDP, and the objective of maximizing the value of the items in the 0-1 backpack problem to the objective of minimizing the total batch dispersion in the BDP. Through the above transformation, it has been shown that the BDP can be reduced to the 0-1 knapsack problem. Therefore, the BDP proposed in this paper is an NPC problem.

## 4. The proposed BDP-GAVNS

In this section, the BDP-GAVNS solution algorithm is created. For algorithm optimization strategy, BDP-GAVNS chooses to embed the variable neighborhood search (VNS) algorithm into the improved genetic algorithm(GA) to enhance the local search capability of BDP-GAVNS; memory bank were introduced to increase the diversity of individuals performing crossover operations. For algorithm improvement, BDP-GAVNS designs a unique coding method based on the characteristics of BDP, under this coding strategy, the solution can be viewed as being composed of segments, each segment is the process of selecting a part batch for a particular product batch; for the emergence of infeasible solutions during the algorithm running process, the infeasible solution repairing strategy is provided; Considering the importance of population initialization in evolutionary algorithms, The GLR part batch selection method is designed for BDP-GAVNS to improve the quality of the initial solution, the GLR methods include global selection (GS), local selection (LS), and random selection (RS). Finally, four kinds of neighborhood structures are designed based on the swapping operation, the reversal operation, the insertion operation and the mutation operation.

### *4.1 Algorithm Optimization Strategy*

The primary BDP-GAVNS optimization strategies are algorithm hybrid strategy and memory bank optimization strategy (MBOS). The algorithm hybrid strategy selects the embedded structure to embed the VNS into the improved GA to improve the local search capability of the GA; the MBOS breaks through the limitation that the interaction of genetic information only comes from the population unilaterally by realizing the self-renewal of the individuals during the iterative process of the algorithm, and the MBOS increases the abundance of the individuals that perform the crossover operation.

#### *4.1.1 algorithm hybrid strategy*

The most common algorithm blending structures are parallel, serial, and embedded. In this paper, we choose the embedded structure to embed the VNS into the enhanced GA so that the GA can provide high-quality initial solutions for the VNS in the process of global optimization, and give full play to the characteristics of the GA with strong global search capability; simultaneously, the powerful neighborhood search capability of the variable neighborhood search algorithm for the region close to the initial solution is exploited to explore the better solution. This embedding structure simultaneously improves the global and local search capabilities of both algorithms, achieving a balance between extensive and centralized search. Fig. 4 depicts the BDP-GAVNS process framework using the embedding structure.

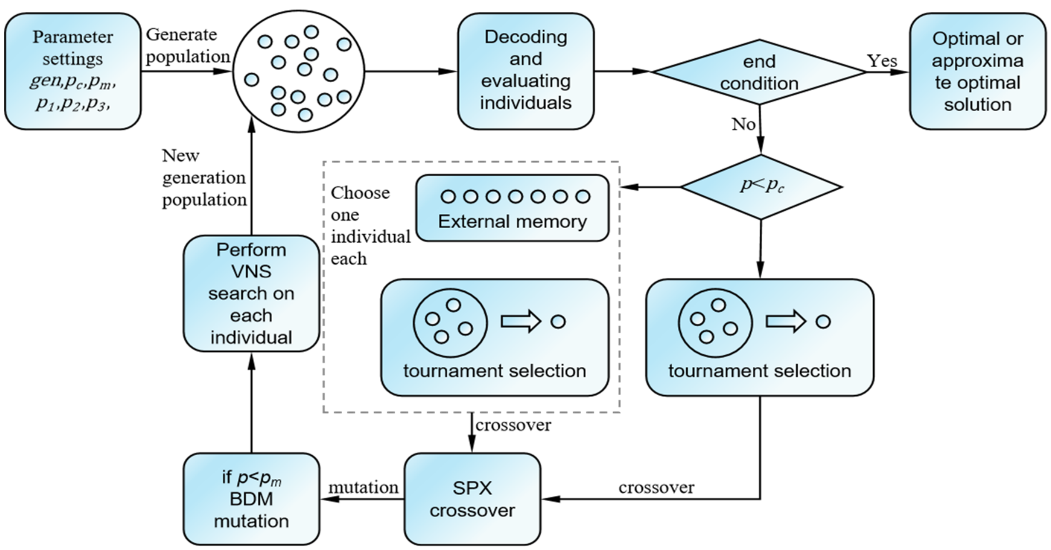



**Fig. 4.** The framework of the proposed BDP-GAVNS

#### *4.1.2 memory bank optimization strategy*

To overcome the precocity-prone nature of traditional genetic algorithms, the information sharing mechanism of memory bank optimization strategy (MBOS) is proposed to circumvent the restriction that the interaction of genetic information between individuals only occurs unilaterally from the population. Simultaneously, the (MBOS) can safeguard the good individuals that emerge during the evolution process and accelerate the solution's iterative convergence process. Consider that the mapping of the solution of the BDP in the search space (SSP) and the feasible solution space (FSS) is a many-to-one relationship (as shown by a in Fig. 3). For example, when $k\_list$ = [10,20,20], $s\_list$ = [20,30], $QD$ =1, according to the coding strategy proposed in Section 4.2.1, Two individuals, 001110 and 010101, can exist simultaneously in the search space of the BDP solution, and both feasible solutions have a minimum batch dispersion of 3 and having the same objective function value. Therefore, the concept of Hemming distance is introduced to support the process of updating the external memory bank in order to retain the individuals with equal optimal values but with differences in the optimal solutions. The Hemming distance is the number of different locations in two individuals with different values (Ye, 2014), or H-distance for short, as shown in Fig. 5, b, where the Hemming distance between two individuals is 3.

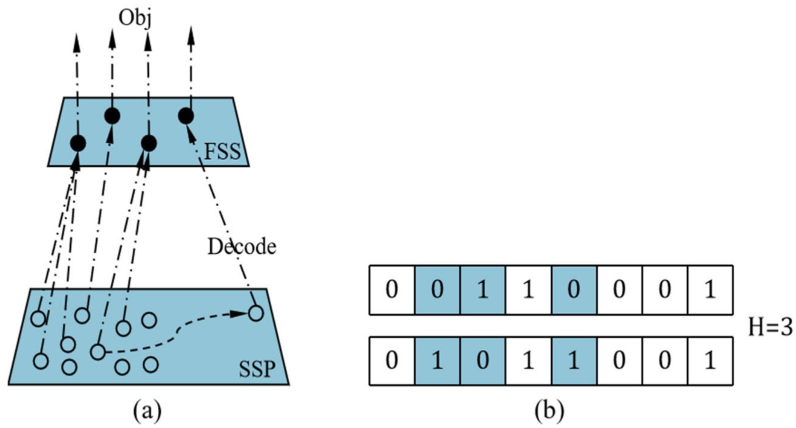



**Fig. 5.** SSP and FSS mapping diagram and Hemming distance schema

During the iteration of the algorithm, a 2-step process of memory bank updating is performed. First, determine whether the objective function value of the new good individual is better than one of the individuals in the memory bank, if yes, replace the individual. Second, if the objective function value of the new excellent individual is equal to one of the individuals in the memory bank, the Hemming distance between the two is judged, if the Hemming distance is not zero, the worst individual in the memory bank is replaced, if it is zero, it is not replaced. The cycle is executed until all new good individuals are compared. The introduction of the memory bank allows individuals to perform crossover operations with two crossover methods to choose from, one is that the individuals involved in the crossover operation are all from the population, and the other is that the individuals are from the memory bank and the population respectively. In the early iteration of BDP-GAVNS, the second method can accelerate the convergence speed of the algorithm; as the number of iterations increases, the number of good individuals increases, to avoid the algorithm falling into the local optimum, the first crossover method is selected to increase the diversity of the population in the late iteration of BDP-GAVNS.

*4.2 Genetic Algorithm Improvements*

In the aspect of algorithm improvement, it mainly contains three items, namely, BDP-GAVNS coding design, population initialization method and mutation operation design. The infeasible solution repair strategy is given for the emergence of infeasible solutions during the running of the algorithm; Considering the importance of population initialization in evolutionary algorithms, a GLR part batch selection method is designed for BDP-GAVNS to improve the initial solution quality, and the GLR methods include global selection (GS), local selection (LS), and random selection (RS). In the genetic operation part of BDP-GAVNS, the tournament selection method is used for the selection operation, and the single point crossover (SPX) method is selected for the crossover operation.

*4.2.1 Encoding strategy*

A one-dimensional vector with $(S \times K)$ elements is used in the coding process, where $S$ denotes the total number of part batches, $K$ denotes the number of product batches, and each element in the vector has a value of either 0 or 1. Under this encoding strategy, the solution consists of $K$ segments, each of which can be regarded as a specific product batch to select a part batch. The indexes of the elements in each segment range from $[(k-1) \times S + 1]$ to $(k \times S)$. In each segment, if the value of an element is equal to 1, it means that the $s$th batch of components is used in the production of the $k$th batch of product, $s = 1, \cdots S$, $k = 1, \cdots K$, otherwise the value of the element is equal to zero. This encoding strategy highlights the characteristics of BDP and facilitates the design of subsequent operations of BDP-GAVNS. Under this encoding strategy, the solution can be written as $X = (x_{SP}(1,1), \cdots x_{SP}(S,1), x_{SP}(S+1,2), \cdots x_{SP}(2 \times S, 2), \cdots x_{SP}(S \times K, K))$.

Each element in the initial solution is first randomly generated. The infeasible solution is then repaired by a solution correction procedure. The feasible solution has to satisfy the following 3 constraint restrictions: (1) each element in the solution should be either 0 or 1; (2) the sum of the number of batches of part used to produce a particular product batch should be greater than or equal to the number of parts demanded by that product batch; (3) The number of parts in a particular batch should be greater than or equal to the number it provides to each product batch. The solution is determined based on the above three constraint limitations, and if it is found to violate any one of the constraints, the solution is repaired, and the pseudo-code of the repair strategy is shown in Table 3.

**Table 3**
Pseudo code of solution correction strategy

**Algorithm 1:Pseudo code of solution correction strategy**

```
Input: X, s_list , k_list , K , S , QD
Output: A feasible solution X
For i←1 to length(k_list) do
  ck_list(i)←k_list(i) / QD
end for
X[1,...,S×K]←0
Index1←find(ck_list≠0)
If index1 is not empty then
  q_sum←0
  list1←[]
  For j←1 to length(index1) do
    Index2←find(s_list≠0)
    For i←1 to length(index2) do
      list1←Merge(list1,index2(i))
      q_sum←q_sum+s_list(index2(i))
      If q_sum=ck_list(index1(j)) then
        s_list(list1)←0
        ck_list(index1(j))←0
        X((index1(j)-1)×S+list1)←1
        list1←[]
        q_sum←0
        break
      Else if q_sum>ck_list(index1(j)) then
        s_list(list1)←0
        s_list(index2(i))←q_sum-ck_list(index1(j))
        ck_list(index1(j))←0
        X((index1(j)-1)×S+list1)←1
        list1←[]
        q_sum←0
        break
      End if
    End for
  End for
End if
```

*4.2.2 GLR population initialization methods*

Population initialization is a key issue in evolutionary algorithms, and the quality of the initial solution has a very strong impact on the speed and quality of variable neighborhood genetic algorithm solution. In this section, a GLR part batch selection method is proposed to improve the quality of the solution, and the GLR method includes GS, LS and RS. GS focuses on balancing the dispersion of the overall component batches; LS aims to reduce the dispersion of certain component batches; and RS aims to enrich the population diversity.

This section sets up a case to explain the GLR part batch selection method. In the case, $\mathrm{K} = \mathrm{S} = 5$, $\mathrm{QD} = 1$, k_list = [6,9,7,5,4], s_list = [7,8,6,6,8].

*4.2.2.1 Global selection (GS)*

Step 1: Set up a one-dimensional integer array of length $(K \times S)$ and all values 0, divide the array into $K$ fragments, the $K$th fragment is $[(K-1) \times S$,...,$K \times S]$.

Step 2: Randomly select the $k$th batch of products from the product batches, and sequentially calculate the difference between the value in the $s\_list$ list and $Q_{FP}(k)$. If the difference is negative, add the difference to the value of the subtracted number located in the last digit of the $s\_lis$ until there is a zero or a positive number.

Step 3: Select the part batch. The part batch corresponding to the subtracted number in step 2 is the batch to be selected.

Step 4: Update $s\_list$. Set the value of the last subtracted number in Step 2 to 0 or a positive number as it appears, and all values before it to 0.

Step 5: Update array segments. The value at the array position corresponding to the selected part batch is updated to 1.

Step 6: Repeat step 2 until all possible values are taken.

Step 7: Join the array segments in sequence to form individuals.

Taking the setup case in this section as an example, assuming that the first randomly selected finished product batch is batch

1 and the second batch is batch 3, the global selection schematic is shown in Fig. 6.

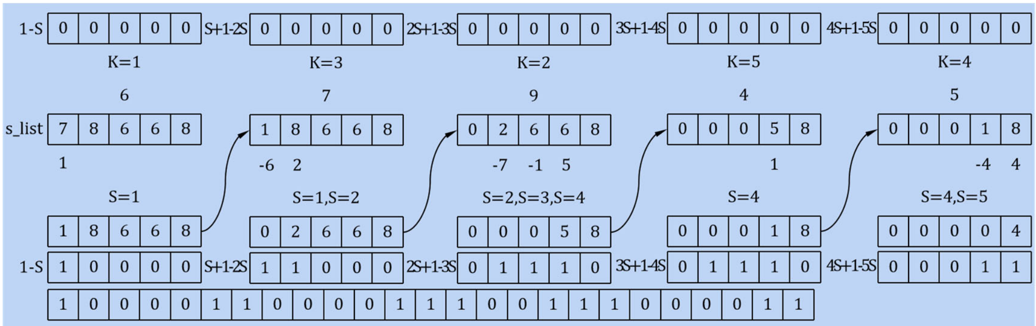


**Fig. 6.** Global selection schematic

*4.2.2.2 Local selection (LS)*

Step 1: Set up a one-dimensional integer array of length $(K \times S)$ and all values 0, divide the array into $K$ fragments, the $K$th fragment is $[(K-1) \times S$,...,$K \times S]$.
Step 2: Randomly select the $k$th batch of products from the product batches and determine whether there is a value greater than or equal to $Q_{FP}(k)$ in the list of $s\_list$. If there is, perform step 3. otherwise, perform step 5.
Step 3: Calculate the difference between the values in the $s\_list$ list and $Q_{FP}(k)$ in turn, prioritizing the first batch of parts with a difference of 0, followed by the batch of parts corresponding to the smallest positive integer in the difference.
Step 4: Calculate the difference between the values in the list of $s\_list$ and $Q_{FP}(k)$ in turn; if the difference is negative, add the difference to the value of the subtracted number in the last digit of $s\_list$ until there is a zero or a positive number.
Step 5: Update the part number list $s\_list$ and individual segments.
Step 6: Repeat step 2 until all possible values are taken.
Step 7: Join the array fragments in order to form an individual.

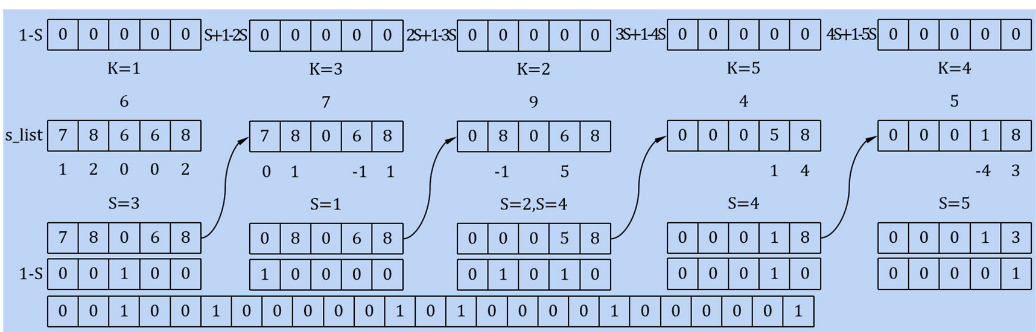


**Fig. 7.** local selection schematic

*4.2.2.3 Radom selection (RS)*

To guarantee the diversity of the initial population, the initial population should be dispersed in the solution space. The method of part batch selection for a portion of the population is RS. RS differs from GS and LS in that the number at each locus is randomly generated. The specific execution steps of random selection are as follows:
Step 1: Set up a one-dimensional integer array of length $K \times S$, where the elements of the array can be randomly valued as 0 or 1.
Step 2: Determine whether the array satisfies the 3 types of model constraints proposed in the encoding strategy, if it does, the array is an individual formed by random initialization.
Step 3: For the solutions that do not satisfy the constraints, apply the solution repair strategy.

*4.3.3 Mutation Operation Design*

Mutation Operation increases population diversity by randomly changing some genes to cause smaller perturbations in chromosomes, thereby creating new individuals. Common mutation operations are insertion mutation, reciprocal mutation and reverse-order mutation (Katoch et al., 2021b). While the above mutation manipulation methods are not very suitable for BDP, therefore, a new mutation manipulation method called multipoint mutation was developed, which is executed as follows:
Step 1: Randomly generate a random number with a value between 1 and $S$ as the first mutation point, denoted as $M_1$.
Step 2: The kth mutation point $M_k$, $M_k = M_1 + (k-1) \times S$, $k = 1, \cdots K$.
Step 3: Perform the mutation operation so that the value at the mutation point is subtracted from 1 and taken as an absolute value.
When $K = 2$, $S = 4$, the schematic of the mutation operation is shown in Fig. 8.

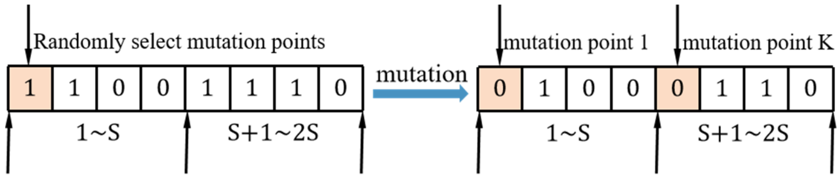



**Fig. 8.** multipoint mutation

### *4.3 Neighborhood structures*

According to the BDP-GAVNS coding characteristics, four neighborhood structures were designed based on the swap operation, reversal operation, insertion operation and mutation operation, respectively (see Fig. 9). Neighborhood structure $NS_1$: this neighborhood structure is based on the swapping operation. Under the coding strategy of Section 4.1, the solution can be viewed as consisting of $K$ segments, and different values in two of them are randomly selected to be exchanged, and this exchange probability enables the solution to still satisfy the supply and demand constraints between the number of products and the number of parts. Suppose the current solution is $X = (x_{SP}(1,1), \cdots x_{SP}(i), \cdots x_{SP}(S,1), x_{SP}(S+1,2), \cdots x_{SP}(j), \cdots x_{SP}(2 \times S, 2), \cdots x_{SP}(S \times K, K))$, If the chosen exchange positions are $i$ and $j$ （$1 \leq i \leq S$， $S \leq j \leq 2 \times S$ ）, Then the solution after exchanging the $i$ th and $j$ th positions can be expressed as: $X = (x_{SP}(1,1), \cdots x_{SP}(j), \cdots x_{SP}(S,1), x_{SP}(S+1,2), \cdots x_{SP}(i), \cdots x_{SP}(2 \times S, 2), \cdots x_{SP}(S \times K, K))$. When $K = 2$, $S = 4$, Suppose the current solution is 00110101, and if $i = 2$ and $j = 6$, the solution after the swap operation is 01110001.

Neighborhood structure $NS_2$: This neighborhood structure is based on the reversal operation, which, unlike the exchange operation that only exchanges elements in two positions, reverses all elements between two positions. Suppose the current solution is $X = (x_{SP}(1,1), \cdots x_{SP}(i), x_{SP}(i+1), \cdots x_{SP}(j), \cdots x_{SP}(S \times K, K))$, If the selected reversal positions are $i$ and $j$ （$i \neq j, 1 \leq i, j \leq S \times K$）, Then the solution after reversing all elements between the $i$th and $j$th positions can be expressed as: $X = (x_{SP}(1,1), \cdots x_{SP}(j), \cdots x_{SP}(i+1), x_{SP}(i), \cdots x_{SP}(S \times K, K))$. When $K = 2$, $S = 4$, Suppose the current solution is 00110101, and if $i = 2$ and $j = 6$, The solution after the reversal operation is then 01011001.

Neighborhood structure $NS_3$: This neighborhood structure is based on the insertion operation that randomly selects two positions and inserts the element corresponding to the first position after the element corresponding to the second position. Suppose the current solution is $X = (x_{SP}(1,1), \cdots x_{SP}(i), x_{SP}(i+1), \cdots x_{SP}(j), \cdots x_{SP}(S \times K, K))$, If the first and second selected positions are i and j, respectively, the solution after the insertion operation is $X = (x_{SP}(1,1), \cdots x_{SP}(i+1), \cdots x_{SP}(j), x_{SP}(i), \cdots x_{SP}(S \times K, K))$. When $K = 2$, $S = 4$, Suppose the current solution is 00110101, and if $i = 2$, $j = 6$, Then the solution after the insertion operation is 01101001.

Neighborhood structure $NS_4$: This neighborhood structure is based on mutation operations. First identify the key fragment in the solution. Define the key fragment as the fragment with the largest cumulative value of all elements within the fragment. Let the elements at each position within the key fragment be randomly mutated to 0 or 1. Reducing the batch dispersion of the key fragment by the mutation operation will enable the solution to be updated to a better solution. When $K = 2$, $S = 4$, Suppose the current solution is 01110101, Then the key fragment of the solution is between 1 and $S$. The solution may become 00100101 after the key fragment has been manipulated by mutation.

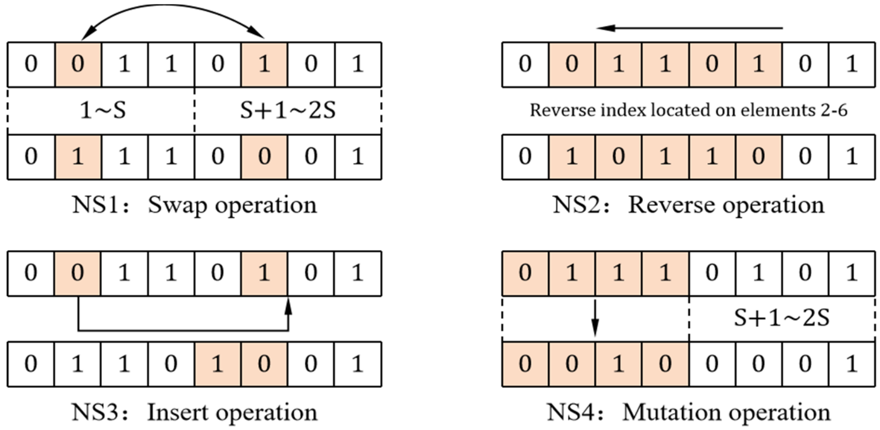



**Fig. 9.** Neighborhood structures $NS_1$, $NS_2$, $NS_3$, $NS_4$

## 5. Computational experiments

The BDP-GAVNS were coded in Matlab language with version R2018b, and they were all run on a computer with Intel Core i7-8565U CPU, 1.99GHz processor, 8G RAM, and Windows 10 system. For the research problem of this paper and the experimental content of this section, the experimental results are better when the crossover probability takes the value of 0.9,

the variance probability takes the value of 0.05, the population size is 100, and the maximum number of iterations is 500.
In this section, a large number of numerical simulation experiments are designed to validate the performance and effectiveness of BDP-GAVNS, including the setting of different sizes of experimental arithmetic, the pre-tuning of experimental parameters, the validation of the effectiveness of the strategies used by the algorithm, the comparison with the Artificial Bee Colony (ABC), the Fast Evolutionary Programming (FEP), Genetic Algorithm (GA), Improved Multi-Operator Differential Evolution (IMODE) and Fast Evolutionary Programming (OFA). These five heuristic algorithms are all from the PlatEMO platform, and the parameter values are adopted from the PlatEMO platform default (Tian et al., 2017). In addition, this section uses a variety of evaluation metrics to analyze the experimental results, including relative percentage deviation and Friedman test.
BDP is a novel problem without any benchmark test instances, two scales of experimental algorithms, large and small, are generated based on the reference of relevant data from Zhejiang Geely Holding Group Co. and Chery Automobile Co. In the small-scale experimental example, the total number of product batches $K$ and the number of part batches $S$ are considered in five scenarios of 20, 30, 40, 50 and 60, respectively. In the large-scale experimental calculus, five scenarios of 70, 80, 90, 100 and 110 are considered for $K$ and $S$, respectively. In the small- and large-scale experimental arithmetic, the number of parts demanded by the product, $QD$, takes values in the range [1, 3]. Product batch $k$ containing product quantity $Q_{FP}(k)$ and part batch $s$ containing part quantity $Q_{SP}(s)$ both take the value in the range [100, 500].

*5.1 Parameters tuning and validation of the algorithm optimization strategy*

The BDP-GAVNS proposed in this paper comprises three categories of population initialization methods; the proportion of individuals generated by GS, LS, and RS to the population is $p_1$, $p_2$ and $p_3$, respectively; the values of these parameters will influence the performance of the BDP-GAVNS. In this section, the full-factorial experiment is selected to establish proportional parameters. Three levels of possible values of 0.1, 0.3, and 0.6 are considered for each parameter, and six combinations must be validated using the full-factorial experiment; additionally, each combination of parameters is tested separately on small-scale experimental arithmetic to obtain Signal-To-Noise Ratios (S/N ratio) data. A greater S/N ratio indicates a more effective combination of parameters (Kong et al., 2020). As a result, we use the S/N ratio as a performance indicator to evaluate the parameters' combination. The S/N ratio determined by the formula: $\mathrm{S/N\ ratio} = 100/\log_{10}\mathrm{obj}^2$, obj represents the objective function's value. The data for the mean S/N ratio in Table 4 were derived by calculating the S/N ratio over 200 seconds for each parameter combination. Figure 8 depicts a line graph of the mean S/N ratio corresponding to the parameters. The optimal ratio of the three varieties of BDP-GAVNS population initialization methods is $p_1$=0.3, $p_2$=0.1, and $p_3$=0.6, as shown in Fig. 10. This parameter combination is used in every subsequent experiment.

**Table 4**
The trials of 6 orthogonal scenarios.

| Trials | Parameters | | | mean S/N ratio |
|---|---|---|---|---|
| | $p_1$ | $p_2$ | $p_3$ | |
| 1 | 0.1 | 0.3 | 0.6 | 29.4766 |
| 2 | 0.1 | 0.6 | 0.3 | 29.0134 |
| 3 | 0.3 | 0.1 | 0.6 | 30.0104 |
| 4 | 0.3 | 0.6 | 0.1 | 29.9815 |
| 5 | 0.6 | 0.1 | 0.3 | 29.5126 |
| 6 | 0.6 | 0.3 | 0.1 | 29.4865 |

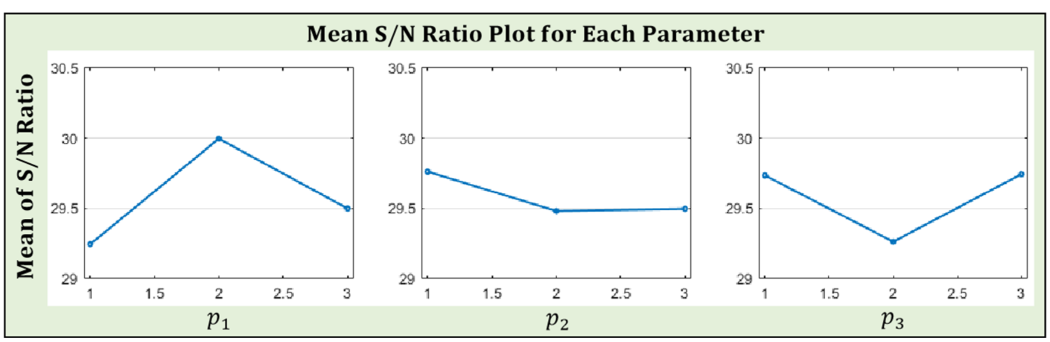


**Fig. 10.** The Mean S/N ratio plot for each level of the parameters $p_1$, $p_2$, and $p_3$.

To enrich the diversity of the sources for individuals performing crossover operations, this paper designs the MBOS for BDP-GAVNS, which are expected to accelerate the iterative convergence process. In this section, experiments are designed to verify the effectiveness of the two strategies, and three scenarios are considered in the experiments: Case1: individuals performing crossover operations are only from the population; Case 2: individuals performing crossover operations are only from the population in the early iterations of BDM-GAVNS, and the best individuals in the memory bank are used for crossover operations in the later iterations; Case 3 is the individual crossover approach proposed in this paper, which is the opposite process of case 2. The experimental results are shown in Table 5, Ave. and Tim. denote the average optimal value and average running time of the BDP-GAVNS algorithm executed 10 times in various experimental scenarios. Where a smaller average optimal value and average running time indicates a more efficient strategy. The data in Table 5 shows that the proposed strategy in this paper is efficient.

In this paper, the MBOS is introduced in BDP-GAVNS to enrich the individual crossover operation, which is expected to accelerate the iterative convergence process and obtain better solutions at the same time. In this section, we experimentally verify the effectiveness of the strategy by considering three scenarios: Scenario 1 is that no MBOS is used, i.e., the individuals performing crossover operations are only from the population; scenario 2 is using the memory bank optimization strategy, but the individuals performing the crossover operation are only from the population in the early optimization iteration, and the good individuals in the memory bank are used for the crossover operation only in the late optimization iteration of the BDP-GAVNS; Case 3 is the individual crossover approach proposed in this paper, which follows the opposite process of case 2. The experimental results are shown in Table 5, where Ave. and Tim. denote the average optimal value and average running time of BDP-GAVNS executed 10 times under various experimental algorithms, respectively. Where, the smaller the average optimal value and average running time indicates the more efficient strategy. The data in Table 5 shows the effectiveness of MBOS.

**Table 5**
Results of experiments on the effectiveness of MBPS

| $No.$ | $K$ | $S$ | $QD$ | Case 1 | | Case 2 | | Case 3 | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Ave. | Tim. | Ave. | Tim. | Ave. | Tim. |
| 1 | 20 | 20 | 1 | 26.60 | 4.57 | 26.60 | 4.69 | 26.30 | 4.47 |
| 2 | 20 | 30 | 1 | 26.30 | 5.21 | 26.40 | 5.37 | 26.10 | 5.13 |
| 3 | 20 | 40 | 2 | 27.20 | 6.03 | 27.10 | 6.18 | 26.90 | 6.05 |
| 4 | 30 | 30 | 1 | 51.20 | 6.92 | 50.90 | 6.98 | 46.60 | 6.70 |
| 5 | 30 | 50 | 2 | 52.40 | 10.40 | 52.10 | 11.05 | 48.10 | 10.21 |
| 6 | 30 | 60 | 3 | 56.70 | 11.55 | 55.91 | 12.13 | 47.30 | 11.29 |
| 7 | 50 | 50 | 1 | 83.60 | 15.34 | 80.70 | 16.26 | 79.90 | 14.98 |
| 8 | 50 | 60 | 1 | 81.50 | 15.08 | 78.40 | 15.79 | 73.60 | 15.16 |
| 9 | 50 | 70 | 2 | 98.60 | 21.36 | 95.30 | 22.57 | 91.80 | 20.86 |
| 10 | 60 | 60 | 1 | 102.30 | 20.28 | 100.50 | 22.18 | 92.10 | 20.27 |
| 11 | 60 | 70 | 2 | 105.60 | 23.17 | 103.30 | 24.97 | 103.60 | 23.06 |
| 12 | 70 | 70 | 1 | 130.70 | 31.25 | 127.60 | 32.89 | 111.30 | 30.06 |

*5.2 Comparison between the proposed IPPSP-VNS with other heuristic algorithms*

In this section, the proposed BDP-GAVNS is compared with five heuristic algorithms, ABC, FEP, GA, IMODE, and OFA, in 30 experimental arithmetic cases that include both small and large scales. Each algorithm is independently tested for 10 repetitions for each of the arithmetic cases, with 500 iterations in a single experiment. The experimental results under small- and large-scale arithmetic are shown in Table 6 and Table 7, respectively. Best_Obj. in the table indicates the optimal value obtained by each algorithm after the experiment. The data shows that BDP-GAVNS exhibits optimality under both small- and large-scale experimental algorithms. The validity of BDP-GAVNS was further verified by applying Relative Percentage Deviation (RPD), which was calculated as follows (Gao et al., 2016):

$$RPD(i) = \frac{Average\_Value(i) - Best\_Fitness}{Best\_Fitness} \times 100$$

$RPD(i)$ denotes the $RPD$ value of Algorithm $i$. The smaller the $RPD$ value, the better solution Algorithm $i$ can find, here $i$ takes the value range of [1, 5], $Average\_Value(i)$ denotes the average value of the algorithm obtained by repeating 10 experiments under a specific example, and *Best_Fitness* denotes the optimal value of all heuristic algorithms found by repeating 10 experiments under a specific example. The above parameters for finding the $RPD$ values are shown in Table 6 and Table 7. Based on the available data the $RPD$ values of the algorithms can be calculated for both large- and small-scale experimental algorithms, and further a box plot of the $RPD$ values can be drawn (as shown in Fig. 11).

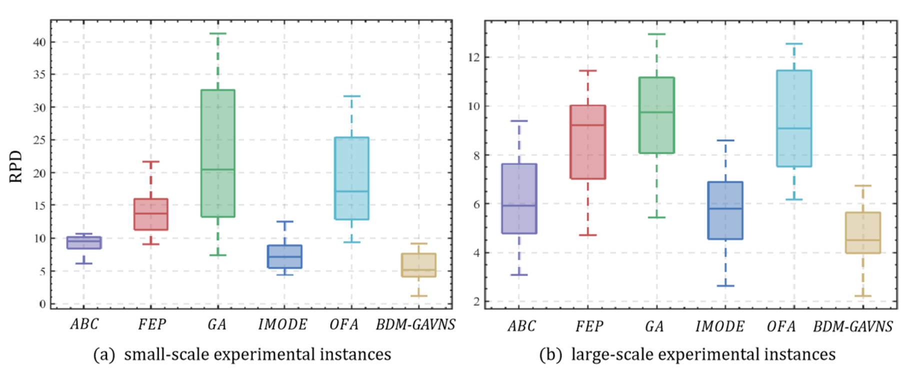
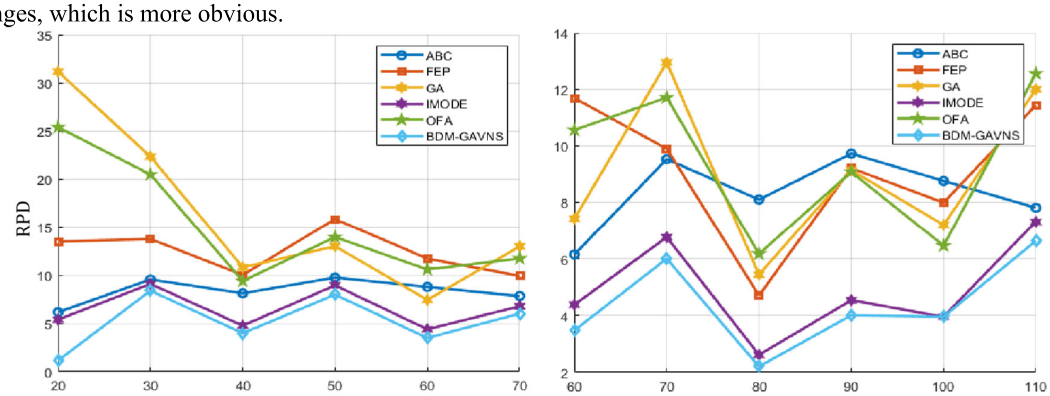



**Fig. 11.** The boxplots of $RPD$ of the compared algorithms in small-scale and large-scale instances.

Under small-scale arithmetic example, the *RPD* box plots of the BDP-GAVNS take values in the range of [1, 10], and those of the ABC, FEP, GA, IMODE and OFA take values in the range of [6, 11], [9, 22], [7, 42], [4, 13], [9, 32], respectively. Under large-scale arithmetic example, the *RPD* box plots of the BDP-GAVNS take values in the range of [2, 7], and the *RPD* box plots of the ABC, FEP, GA, IMODE, and OFA take values in the range of [3, 10], [4, 12], [5, 13], [2, 9], and [6, 13], respectively. From the box plots of *RPD* values, the *RPD* mean, minimum, maximum, median, upper quartile and lower quartile of BDP-GAVNS algorithm are smaller than other algorithms, therefore, BDP-GAVNS is more stable and efficient.

The a and b graphs in Fig. 12 show the variation of RPD of each algorithm with the number of product batches and part batches, the fold line corresponding to the RPD value of the BDP-GAVNS is located at the lowest place in both cases, the fluctuation amplitude is small and has a certain degree of regularity, which shows that BDP-GAVNS take into account both high efficiency and stability at the same time. The trend of the RPD value of BDP-GAVNS is closer to IMODE, while the other algorithms are farther away from the RPD value of the BDP-GAVNS, especially when the number of batches of the parts changes, which is more obvious.




(a) Number of product batches (b) Number of part batches

**Fig. 12.** Variation of RPD value with number of product batches and part batches

**Table 6**
The comparison results between the BDP-GAVNS and the other five heuristic algorithms in small-scale instances.

| *No.* | *K* | *S* | *QD* | Best_ Obj. | ABC | | FEP | | GA | | IMODE | | OFA | | BDP-GAVNS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Ave. | *RPD* | Ave. | *RPD* | Ave. | *RPD* | Ave. | *RPD* | Ave. | *RPD* | Ave. | *RPD* |
| 1 | 20 | 20 | 1 | 26 | 27.60 | 6.15 | 29.50 | 13.46 | 34.10 | 31.15 | 27.40 | 5.38 | 32.60 | 25.38 | 26.30 | 1.15 |
| 2 | 20 | 30 | 1 | 25 | 27.40 | 9.60 | 29.40 | 17.60 | 33.80 | 35.20 | 27.10 | 8.40 | 31.30 | 25.20 | 26.10 | 4.40 |
| 3 | 20 | 40 | 2 | 26 | 28.20 | 8.46 | 29.90 | 15.00 | 34.60 | 33.08 | 27.50 | 5.77 | 34.10 | 31.15 | 26.90 | 3.46 |
| 4 | 20 | 50 | 2 | 25 | 27.10 | 8.40 | 29.40 | 17.60 | 34.70 | 38.80 | 27.20 | 8.80 | 32.10 | 28.40 | 26.30 | 5.20 |
| 5 | 20 | 60 | 3 | 24 | 26.50 | 10.42 | 29.20 | 21.67 | 33.90 | 41.25 | 27.00 | 12.50 | 31.60 | 31.67 | 26.20 | 9.17 |
| 6 | 30 | 30 | 1 | 43 | 47.10 | 9.53 | 48.90 | 13.72 | 52.60 | 22.33 | 46.90 | 9.07 | 51.80 | 20.47 | 46.60 | 8.37 |
| 7 | 30 | 40 | 1 | 43 | 47.30 | 10.00 | 46.90 | 9.07 | 51.80 | 20.47 | 46.50 | 8.14 | 50.30 | 16.98 | 46.50 | 8.14 |
| 8 | 30 | 50 | 2 | 46 | 50.20 | 9.13 | 50.60 | 10.00 | 55.20 | 20.00 | 49.30 | 7.17 | 53.90 | 17.17 | 48.10 | 4.57 |
| 9 | 30 | 60 | 3 | 45 | 49.60 | 10.22 | 50.00 | 11.11 | 54.60 | 21.33 | 48.20 | 7.11 | 53.40 | 18.67 | 47.30 | 5.11 |
| 10 | 40 | 40 | 1 | 63 | 68.10 | 8.10 | 69.30 | 10.00 | 69.80 | 10.79 | 66.00 | 4.76 | 68.90 | 9.37 | 65.50 | 3.97 |
| 11 | 40 | 50 | 1 | 61 | 67.50 | 10.66 | 69.80 | 14.43 | 69.50 | 13.93 | 65.10 | 6.72 | 68.60 | 12.46 | 64.60 | 5.90 |
| 12 | 40 | 60 | 2 | 65 | 69.90 | 7.54 | 73.30 | 12.77 | 70.90 | 9.08 | 68.20 | 4.92 | 71.70 | 10.31 | 68.90 | 6.00 |
| 13 | 50 | 50 | 1 | 74 | 81.20 | 9.73 | 85.70 | 15.81 | 83.60 | 12.97 | 80.60 | 8.92 | 84.30 | 13.92 | 79.90 | 7.97 |
| 14 | 50 | 60 | 1 | 69 | 76.30 | 10.58 | 80.10 | 16.09 | 80.10 | 16.09 | 76.10 | 10.29 | 80.70 | 16.96 | 73.60 | 6.67 |
| 15 | 60 | 60 | 1 | 89 | 96.80 | 8.76 | 99.40 | 11.69 | 95.60 | 7.42 | 92.90 | 4.38 | 98.40 | 10.56 | 92.10 | 3.48 |

**Table 7**
The comparison results between the BDP-GAVNS and the other five heuristic algorithms in large-scale instances.

| *No.* | *K* | *S* | *QD* | Best_Obj. | ABC | | FEP | | GA | | IMODE | | OFA | | BDP-GAVNS | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Ave. | *RPD* | Ave. | *RPD* | Ave. | *RPD* | Ave. | *RPD* | Ave. | *RPD* | Ave. | *RPD* |
| 1 | 70 | 70 | 1 | 105 | 113.20 | 7.81 | 115.40 | 9.90 | 118.60 | 12.95 | 112.10 | 6.76 | 117.30 | 11.71 | 111.30 | 6.00 |
| 2 | 70 | 80 | 1 | 105 | 112.90 | 7.52 | 114.80 | 9.33 | 118.30 | 12.67 | 111.70 | 6.38 | 117.50 | 11.90 | 111.00 | 5.71 |
| 3 | 70 | 90 | 2 | 114 | 118.70 | 4.12 | 120.60 | 5.79 | 123.50 | 8.33 | 118.90 | 4.30 | 122.10 | 7.11 | 117.60 | 3.16 |
| 4 | 70 | 100 | 2 | 112 | 118.50 | 5.80 | 119.80 | 6.96 | 122.90 | 9.73 | 118.50 | 5.80 | 123.00 | 9.82 | 116.80 | 4.29 |
| 5 | 70 | 110 | 3 | 120 | 127.10 | 5.92 | 128.30 | 6.92 | 129.60 | 8.00 | 127.20 | 6.00 | 130.90 | 9.08 | 125.40 | 4.50 |
| 6 | 80 | 80 | 1 | 149 | 153.60 | 3.09 | 156.00 | 4.70 | 157.10 | 5.44 | 152.90 | 2.62 | 158.20 | 6.17 | 152.30 | 2.21 |
| 7 | 80 | 90 | 1 | 140 | 150.30 | 7.36 | 155.10 | 10.79 | 155.80 | 11.29 | 146.70 | 4.79 | 156.70 | 11.93 | 145.90 | 4.21 |
| 8 | 80 | 100 | 2 | 152 | 159.10 | 4.67 | 162.90 | 7.17 | 163.00 | 7.24 | 158.90 | 4.54 | 162.60 | 6.97 | 157.70 | 3.75 |
| 9 | 80 | 110 | 3 | 154 | 162.80 | 5.71 | 167.20 | 8.57 | 166.80 | 8.31 | 161.50 | 4.87 | 167.40 | 8.70 | 161.20 | 4.68 |
| 10 | 90 | 90 | 1 | 152 | 159.70 | 5.07 | 166.00 | 9.21 | 165.90 | 9.14 | 158.90 | 4.54 | 165.80 | 9.08 | 158.10 | 4.01 |
| 11 | 90 | 100 | 1 | 150 | 161.50 | 7.67 | 164.30 | 9.53 | 165.40 | 10.27 | 162.40 | 8.27 | 165.10 | 10.07 | 157.50 | 5.00 |
| 12 | 90 | 110 | 1 | 149 | 161.10 | 8.12 | 164.00 | 10.07 | 165.10 | 10.81 | 161.80 | 8.59 | 164.90 | 10.67 | 157.10 | 5.44 |
| 13 | 100 | 100 | 1 | 164 | 171.00 | 4.27 | 177.10 | 7.99 | 175.80 | 7.20 | 170.50 | 3.96 | 174.60 | 6.46 | 170.50 | 3.96 |
| 14 | 100 | 110 | 1 | 159 | 169.80 | 6.79 | 176.50 | 11.01 | 175.40 | 10.31 | 170.00 | 6.92 | 173.40 | 9.06 | 169.70 | 6.73 |
| 15 | 110 | 110 | 1 | 160 | 175.00 | 9.38 | 178.30 | 11.44 | 179.20 | 12.00 | 171.70 | 7.31 | 180.10 | 12.56 | 170.60 | 6.63 |

The average optimal values of the six algorithms under small and large scale arithmetic cases are statistically analyzed using Friedman's test (Friedman, 1940), and if the average optimal value of a certain heuristic algorithm is the smallest by repeating the experiment for 10 times in a particular case, the ranking of the algorithm is set to 6, i.e., the smaller the average optimal value of the algorithm, the lower its ranking is. According to this ranking method, the average ranking of each heuristic algorithm can be calculated (see Table 8). The average ranking of BDP-GAVNS is 5.90, which is greater than that of all other heuristic algorithms.

The original hypothesis $H_0$ of Friedman's test is that there is no significant difference between the six algorithms ABC, FEP, GA, IMODE, OFA, BDP-GAVNS and the significance level is 0.05. The alternative hypothesis $H_1$ is that there is a significant difference between the above six algorithms. The Chi-Square value $\chi^2$ is 27.531, and from the chi-square distribution table, ${\chi^2}_{0.05}(6-1) = 11.07$, while 11.07<27.531, and p-value 0.00004<<0.05, so the alternative hypothesis is accepted and the original hypothesis is rejected, i.e., there is a significant difference in the six algorithms, and BDP-GAVNS is more superior.

**Table 8**
The statistical results obtained by Friedman test of the compared algorithms.

| Algorithms | Average rank |
|---|---|
| ABC | 4.20 |
| FEP | 2.47 |
| GA | 1.63 |
| IMODE | 4.83 |
| OFA | 1.97 |
| BDM-GAVNS | 5.90 |
| p-value | 0.00004 |
| Chi-Square | 27.531 |

## 6. Conclusions

This paper discusses the batch dispersion problem (BDP) under the assembled bill of materials (BOM). When the batch dispersion is used to measure the degree of dispersed use of part batches and the number of recalls is used to evaluate the traceability performance, it is found that the BDP is caused by the random matching combinations of parts batches and product batches, which is not conducive to the reduction of the number of recalls and the improvement of the traceability. Therefore, this paper optimizes the matching combinations between part batches and product batches at the production planning stage, builds a mixed-integer planning model with the objective of minimizing the total batch dispersion of part batches, and demonstrates that the BDP under the assembled BOM is an NPC problem. Since BDP is an NPC problem, the BDP-GAVNS hybrid heuristic algorithm is proposed to tackle it. In the algorithm optimization strategy, BDP-GAVNS chooses to embed the VNS into the improved GA in order to improve the local search capability, and also develops the memory bank preservation strategy (MBOS) to enhance the crossover operation. Considering the significance of population initialization in evolutionary algorithms, a GLR part batch selection method is developed for BDP-GAVNS to improve genetic algorithm performance. Four types of neighborhood structures are created based on the switch operation, the reversal operation, the insertion operation, and the mutation operation, respectively. The feasibility and effectiveness of BDP-GAVNS in solving BDP are verified by numerical experiments, which show that (1) the optimal combination is when the proportion of individuals generated by the three population initialization methods, GS, LS and RS, to the population takes the values of 0.3, 0.1, and 0.6, respectively; (2) the MBOS enriches the source of individuals required for crossover operations and improves the algorithm's optimization capability and efficiency; (3) BDP-GAVNS is more effective than the other five heuristic algorithms including genetic algorithm in seeking the optimal solution of BDP.

The research in this paper only explores the BDP under single-level assembled BOM, and the actual industrial application also involves disassembled BOM, so how to use the research in this paper as a basis to solve the BDP under multi-level BOM is a problem worthy of further research. Despite the large number of experiments in this paper, it is still difficult to avoid the existence of randomness, and in the future, it may be possible to use big data analytics, artificial intelligence technology and blockchain technology, etc. to predict and solve the BDP.

**Acknowledgments**

This work was supported by the National Key R&D Program of China under Grant number 2021YFB3300900 and the National Natural Science Foundation of China under Grant number 72371087, 71531008, 72188101.

## References

Astill, J., Dara, R. A., Campbell, M., Farber, J. M., Fraser, E. D., Sharif, S., & Yada, R. Y. (2019). Transparency in food supply chains: A review of enabling technology solutions. *Trends in Food Science & Technology*, *91*, 240-247. https://doi.org/10.1016/j.tifs.2019.07.024

Aung, M. M., & Chang, Y. S. (2014). Traceability in a food supply chain: Safety and quality perspectives. *Food Control, 39,* 172–184. https://doi.org/10.1016/j.foodcont.2013.11.007

Badia-Melis, R., Mishra, P., & Ruiz-García, L. (2015). Food traceability: New trends and recent advances. A review. *Food Control, 57*, 393–401. https://doi.org/10.1016/j.foodcont.2015.05.005

Chen, T., Ding, K., & Hao, S. (2019). Batch-based traceability for pork: A mobile solution with 2D barcode technology. *Food Control, 107*. https://doi.org/10.1016/j.foodcont.2019.106770

Comba, L., Belforte, G., Dabbene, F., & Gay, P. (2013). Methods for traceability in food production processes involving bulk products. *Biosystems Engineering, 116*(1), 51–63. https://doi.org/10.1016/j.biosystemseng.2013.06.006

Dabbene, F., & Gay, P. (2011). Food traceability systems: Performance evaluation and optimization. *Computers and Electronics in Agriculture, 75*(1), 139–146. https://doi.org/10.1016/j.biosystemseng.2013.09.006

Dabbene, F., Gay, P., & Tortia, C. (2014). Traceability issues in food supply chain management: A review. *Biosystems Engineering, 120*, 65–80. https://doi.org/10.1016/j.biosystemseng.2013.09.006

Deng, W., Zhao, H., Zou, L., Li, G., Yang, X., & Wu, D. (2017). A novel collaborative optimization algorithm in solving complex optimization problems. *Soft Computing, 21*(15), 4387–4398. https://doi.org/10.1007/s00500-016-2071-8

Dupuy, C., Botta-Genoulaz, V., & Guinet, A. (2005). Batch dispersion model to optimise traceability in food industry. *Journal of Food Engineering, 70*(3), 333–339. https://doi.org/10.1016/j.jfoodeng.2004.05.074

Fan, B., Qian, J., Wu, X., Du, X., Li, W., Ji, Z., & Xin, X. (2019). Improving continuous traceability of food stuff by using barcode-RFID bidirectional transformation equipment: Two field experiments. *Food Control, 98*, 449–456. https://doi.org/10.1016/j.foodcont.2018.12.002

Friedman, M. (1940). A Comparison of Alternative Tests of Significance for the Problem of $m$ Rankings. *The Annals of Mathematical Statistics, 11*(1), 86–92. https://doi.org/10.1214/aoms/1177731944

Gao, K. Z., Suganthan, P. N., Pan, Q. K., Chua, T. J., Chong, C. S., & Cai, T. X. (2016). An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time. *Expert Systems with Applications, 65,* 52–67. https://doi.org/10.1016/j.eswa.2016.07.046

Garg, H. (2016). A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation, 274*, 292–305. https://doi.org/10.1016/j.amc.2015.11.001

ISO (EN ISO 8402.1995, Point 3.16). (1995). [ISO European Standard (1995)].

Katoch, S., Chauhan, S. S., & Kumar, V. (2021a). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications, 80*(5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6

Katoch, S., Chauhan, S. S., & Kumar, V. (2021b). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications, 80*(5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6

Khalilpourazari, S., & Khalilpourazary, S. (2019). An efficient hybrid algorithm based on Water Cycle and Moth-Flame Optimization algorithms for solving numerical and constrained engineering optimization problems. *Soft Computing, 23*(5), 1699–1722. https://doi.org/10.1007/s00500-017-2894-y

Kong, M., Pei, J., Liu, X., Lai, P.-C., & Pardalos, P. M. (2020). Green manufacturing: Order acceptance and scheduling subject to the budgets of energy consumption and machine launch. *Journal of Cleaner Production, 248*, 119300. https://doi.org/10.1016/j.jclepro.2019.119300

Li, M., Qian, J.-P., Yang, X.-T., Sun, C.-H., & Ji, Z.-T. (2010). A PDA-based record-keeping and decision-support system for traceability in cucumber production. *Computers and Electronics in Agriculture, 70*(1), 69–77. https://doi.org/10.1016/j.compag.2009.09.009

Liang, K., Chen, X., He, R., Li, J., Okinda, C., Han, D., & Shen, M. (2019). Development and parameter optimization of automatic separation and identification equipment for grain tracing systems based on grain tracers with QR codes. *Computers and Electronics in Agriculture, 162*, 709–718. https://doi.org/10.1016/j.compag.2019.04.039

Liang, K., Thomasson, J. A., Lee, K.-M., Shen, M., Ge, Y., & Herrman, T. J. (2012). Printing data matrix code on food-grade tracers for grain traceability. *Biosystems Engineering, 113*(4), 395–401. https://doi.org/10.1016/j.biosystemseng.2012.09.012

Luvisi, A., Panattoni, A., Bandinelli, R., Rinaldelli, E., Pagano, M., & Triolo, E. (2012). Ultra-High Frequency transponders in grapevine: A tool for traceability of plants and treatments in viticulture. *Biosystems Engineering, 113*(2), 129–139. https://doi.org/10.1016/j.biosystemseng.2012.06.015

Maity, M., Tolooie, A., Sinha, A. K., & Tiwari, M. K. (2021). Stochastic batch dispersion model to optimize traceability and enhance transparency using Blockchain. *Computers & Industrial Engineering, 154*, 107134. https://doi.org/10.1016/j.cie.2021.107134

Moe, T. (1998). Perspectives on traceability in food manufacture—ScienceDirect. *Trends in Food Science & Technology, 9*(5), 211–214. https://doi.org/10.1016/S0924-2244(98)00037-5

Mokarram, M. J., Niknam, T., Aghaei, J., Shafie-khah, M., & Catalão, J. P. S. (2019). Hybrid Optimization Algorithm to Solve the Nonconvex Multiarea Economic Dispatch Problem. *IEEE Systems Journal, 13*(3), 3400–3409. https://doi.org/10.1109/JSYST.2018.2889988

Pierini, G. D., Fernandes, D. D. S., Diniz, P. H. G. D., de Araújo, M. C. U., Di Nezio, M. S., & Centurión, M. E. (2016). A digital image-based traceability tool of the geographical origins of Argentine propolis. *Microchemical Journal, 128,* 62–67. https://doi.org/10.1016/j.microc.2016.04.015

Qian, J., Dai, B., Wang, B., Zha, Y., & Song, Q. (2022). Traceability in food processing: Problems, methods, and performance evaluations—a review. *Critical Reviews in Food Science and Nutrition, 62*(3), 679–692. https://doi.org/10.1080/10408398.2020.1825925

Qian, J., Fan, B., & Wu, X. (2017). Cbmprehensive and quantifiable granularity: A novel model to measure agro-food traceability. *Food Control, 74*, 98–106. https://doi.org/10.1016/j.foodcont.2016.11.034

Qian, J., Ruiz-Garcia, L., Fan, B., Robla Villalba, J. I., McCarthy, U., Zhang, B., Yu, Q., & Wu, W. (2020). Food traceability system from governmental, corporate, and consumer perspectives in the European Union and China: A comparative review. *Trends in Food Science & Technology, 99*, 402–412. https://doi.org/10.1016/j.tifs.2020.03.025

Sardina, M. T., Tortorici, L., Mastrangelo, S., Di Gerlando, R., Tolone, M., & Portolano, B. (2015). Application of microsatellite markers as potential tools for traceability of Girgentana goat breed dairy products. *Food Research International, 74,* 115–122. https://doi.org/10.1016/j.foodres.2015.04.038

Slowik, A., & Kwasnicka, H. (2020). Evolutionary algorithms and their applications to engineering problems. *Neural Computing and Applications, 32*(16), 12363–12379. https://doi.org/10.1007/s00521-020-04832-8

Tian, Y., Cheng, R., Zhang, X., & Jin, Y. (2017). PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *IEEE Computational Intelligence Magazine, 12*, 73–87. https://doi.org/10.1109/MCI.2017.2742868

Van der Spiegel, M., Sterrenburg, P., Haasnoot, W., & van der Fels-Klerx, H. J. (2013). Towards a decision support system for control of multiple food safety hazards in raw milk production. *Trends in Food Science & Technology, 34*(2), 137–145. https://doi.org/10.1016/j.tifs.2013.10.001

Wang, W., Tian, G., Zhang, H., Li, Z., & Zhang, L. (2023). A hybrid genetic algorithm with multiple decoding methods for energy-aware remanufacturing system scheduling problem. *Robotics and Computer-Integrated Manufacturing, 81*, 102509. https://doi.org/10.1016/j.rcim.2022.102509

Yang, X., Qian, J., Li, J., Ji, Z., Fan, B., Xing, B., & Li, W. (2016). A real-time agro-food authentication and supervision system on a novel code for improving traceability credibility. *Food Control, 66,* 17–26. https://doi.org/10.1016/j.foodcont.2016.01.032

Ye, J. (2014). Similarity measures between interval neutrosophic sets and their applications in multicriteria decision-making. *Journal of Intelligent & Fuzzy Systems, 26*(1), 165–172. https://doi.org/10.3233/IFS-120724

Zhang, Z., Ding, S., & Jia, W. (2019). A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems. *Engineering Applications of Artificial Intelligence, 85*, 254–268. https://doi.org/10.1016/j.engappai.2019.06.017

Zhao, H., Wang, F., & Yang, Q. (2020). Origin traceability of peanut kernels based on multi-element fingerprinting combined with multivariate data analysis. *Journal of the Science of Food and Agriculture, 100*(10), 4040–4048. https://doi.org/10.1002/jsfa.10449