



A self-adaptive estimation of distribution algorithm with differential evolution strategy for supermarket location problem

Bing-Hai Zhou¹ · Fen Tan¹

Received: 22 March 2018 / Accepted: 23 January 2019 / Published online: 8 March 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

In modern production systems, an ever-rising product variety has imposed great challenges for in-plant part supply systems used to feed mixed-model assembly lines with required parts. In recent years, many automotive manufacturers have identified the supermarket concept as an efficient part feeding strategy to enable JIT (Just-in-time) deliveries at low costs. This paper studies a discrete supermarket location problem which considers the utilization rate and capacity constraint of the supermarkets simultaneously. Firstly, a mathematical model is developed with the objective of minimizing the total system cost consisting of operating cost and transportation cost. Then, a self-adaptive estimation of distribution algorithm with differential evolution strategy, named DE/AEDA, is proposed to solve the problem. Finally, computational experiments are carried out to analyze the performance of the proposed algorithm compared with the benchmark algorithm by using a non-parametric test method. The results indicate that the proposed algorithm is valid and efficient.

Keywords In-plant material delivery · Supermarket · Location · Estimation of distribution algorithm

1 Introduction

In the past few decades, in order to satisfy the personalized and diversified demand of customers, mixed-model assembly lines (MMALs) have been widely adopted in today's automobile industry [1, 2]. In this case, several models are to be assembled by the same assembly line. And thousands of diversified parts are obliged to be delivered to multiple stations timely and accurately. Thus, part delivery scheduling at these high-variant MMALs has been a crucial challenge due to the diversified demand and the limited line-side inventory capacity [3, 4].

In most of the past cases, parts are transported from centralized warehouse directly to stations. However, in recent decades, the supermarket concept has been adopted by more and more automobile manufacturers to enable a frequent and small-lot part delivery to ensure the smooth

flow of assembling production [5]. In this case, parts are intermediately stored in the supermarket, which is a decentralized logistics area near the assembly line where parts are consumed. And the parts in the supermarket are replenished by industrial trucks from the centralized warehouse. Then, based on the predefined schedule, parts required by production are transported to stations by means of tow train on the fixed route. Due to the increase in frequency of material delivery, it is easier and more flexible to re-plan when encountering unexpected situations. Moreover, the line-side inventory and transportation distance can also be reduced effectively. In this case, the supermarket concept is increasingly prevalent in automobile production.

Four interrelated decision problems have to be settled when planning and controlling the material delivery process based on this concept [4].

- (i) Supermarket location problem (SLP): determining the number and location of these decentralized supermarkets.
- (ii) Tow train routing (TTR) problem: determining the number of required tow trains per supermarket and the route of each tow train. The route starts from

✉ Bing-Hai Zhou
bhzhou@tongji.edu.cn

Fen Tan
tanfen007@tongji.edu.cn

¹ School of Mechanical Engineering, Tongji University, Shanghai 201804, People's Republic of China

- the supermarket, visiting a sequence of stations, and eventually returns back to the supermarket.
- (iii) Tow train scheduling (TTS) problem: determining the fixed delivery schedule per tow train, including the number of deliveries and the start time of each delivery.
 - (iv) Tow train loading (TTL) problem: determining the number of part bins to be loaded to corresponding stations per tour of the tow train.

It is assumed that supermarket system is employed in this paper. However, since the space on the shop floor is scarce and expensive [6], it is necessary to make a reasonable plan to decide on the location plan of supermarkets. In other words, decision (i) is the focus of this paper. The Supermarket location problem (SLP) is the most long-term problem after deciding to implement the supermarket concept concerned with location planning. In this context, the number of supermarkets to be installed on the shop-floor and their exact locations need to be determined. Furthermore, each supermarket is to be roughly assigned to the line segments it serves. Clearly, it has to be differentiated between initial location planning in a newly erected plant and a relocation problem. While in the former case some flexibility for finding a suited location per supermarket exists, in the latter case, typically, severe constraints with regard to the size and location need to be considered. In the objective function the fixed cost caused by any additional supermarket need to be weighed against the operational gains of the supermarket-concept. Clearly, an exact quantification of these gains requires an anticipation over the complete planning hierarchy. For instance, the reduction of the travel distances to the line for any additional supermarket should be exactly quantified.

The supermarket concept is a relatively new field of research. To date, there are few literatures explicitly tackling above decision problems. Vaidyanathan et al. [7] extended the classical vehicle routing problem into a new problem named Just-in-Time Capacitated Vehicle Routing Problem (JITCVRP), to deal with decision problem (ii). It was assumed that the route, once assigned to a tow train, must be served without breaks over the planning horizon, in which decision problem (iii) was not considered. In order to ensure that deliveries are carried out according to the just-in-time principle, the objective of JITCVRP was to minimize the total trip time along all routes. Then, two heuristics were proposed to solve it. Emde and Boysen [8] solved the routing problem (ii) and scheduling problem (iii) simultaneously by the proposed nested dynamic programming procedure, with the objective of minimizing all bins lying in stock over planning horizon and the number of tow trains applied, to make sure a reliable material delivery in line with JIT principle. On this foundation, Emde et al. [9]

further studied the tow train loading problem (iv) given station demands, specific routes and schedules to determine the number of bins to be loaded to corresponding stations per tour of a capacity-limited tow train. They developed an exact algorithm with polynomial time to simultaneously minimize the maximum and the sum inventories near the line. The joint routing (ii), scheduling (iii) and loading (iv) problem of tow train was firstly tackled by Golz et al. [10] in the case of a major German OEM given a set of possible routes to choose. A heuristic procedure was presented to minimize the number of vehicle operators while avoiding stockouts at the line. Fathi et al. [11–13] aimed at minimizing the number of tours and line-side inventories to deal with the scheduling (iii) and loading (iv) problem of tow train jointly with multi-objective delivery scheduling model. The difference among them is that different algorithm was presented based on different priority heuristic rules. As for the focus of this paper, little was published regarding on decision problem (i) in in-plant environment. Emde and Boysen [6] considered the problem of locating supermarkets on the shop floor by determining the optimal number and placements of supermarkets. A dynamic programming scheme was developed because it just assumed that the supermarkets could be established everywhere around the stations without consideration of some unavailable places and capacity limit of the supermarkets. Subsequently, Alnahhal and Noche [14] took into account the above two aspects. Therefore, the time complexity of SLP was improved and the problem was NP-complete. Then, a real genetic algorithm (RGA) was constructed to solve the problem.

To the best knowledge of the authors, the aforementioned papers are the only papers explicitly solving decision problem (i). SLP bears some similarities to the well-known capacitated p-median problem (CPMP), one of the discrete facility location problems (FLP). The aim is to locate a given number of facilities from a set of sites to satisfy the demands of all customers, so that the sum of the weighted distances between facilities and assigned customers is minimized. There are two differences between CPMP and SLP. Firstly, customers served by one facility are consecutive in SLP. Secondly, unlike the CPMP, the number of facilities is not given, which means that it should be optimized to find the optimal number of supermarkets. Therefore, models and methods of CPMP are partially applicable to the SLP. As is known to all, the CPAM is an NP-hard problem and thus it is unpractical to solve it by exact algorithms [15]. Genetic algorithm (GA) [16–18], tabu search (TS) algorithm [19] and other evolutionary algorithms are most commonly used to solve CPMP [20, 21]. Correa et al. [22] proposed a modified GA with a new heuristic “hyper-mutation” operator to minimize the sum of the weighted distances. Herda [15] employed a

combined GA, combining hypermutation from G1 and dominant crossover from G2, for the CPMP. The results turned out that the authors built a high-performance parallel algorithm. As for TS, França et al. [23] developed a new adaptive tabu search approach to solve the problem. In addition, two local search heuristics, pairwise interchange and insertion, were added to generate neighborhood solution. Yaghini et al. [24] combined TS algorithm and a proposed cutting-plane neighborhood structure to construct an efficient hybrid metaheuristic algorithm. Landa-Torres [25] applied two different grouping-based algorithms to tackle the CPMP, including grouping genetic algorithm (GGA) and grouping harmony search approach (GHS). By computational experiments, the proposed grouping methods were proved to outperform other benchmark algorithms.

It has been known that the metaheuristics used in aforementioned literatures are mainly derived from GA, which generates new candidates by crossover and mutation operators. Some new metaheuristics have been developed last few decades, and it indicates that new metaheuristics ought to be explored to deal with the SLP. Estimation of distribution algorithm (EDA), originally introduced by Mühlenbein and Paass [26], is a new class of group evolutionary algorithms (EA) based on statistical learning theory [27]. Unlike other EAs, instead of generating new solutions by chromosome recombination, EDA describes the distribution information of candidate solutions in search space by establishing the probability model, based on which new solutions are sampled [28]. It avoids the blindness and randomness caused by chromosome recombination, thus the search efficiency is effectively improved. Due to its great search efficiency, EDA received much attention and has been applied to solve many engineering optimization problems, including flow shop scheduling [29], project scheduling [30, 31], nurse rostering [32] etc. To the best of authors' knowledge, there has not been published paper dealing with discrete SLP problem using EDA. In this paper, EDA is chosen as the main framework of our optimization algorithm.

In view of some unavailable places on the assembly shop, this paper proposes the discrete SLP problem considering the utilization rate and capacity limit of the supermarkets simultaneously with the objective of minimizing the total system cost consisting of operating and transportation cost. Then, the authors choose EDA as the main framework of our optimization algorithm. EDA is good at global searching, but lacks the ability for refined search. Therefore, differential evolution (DE) strategy and adaptive learning rate mechanism are integrated into EDA to enhance the performance of algorithm. In this paper, the proposed algorithm is named the self-adaptive estimation

of distribution algorithm with differential evolution strategy (DE/AEDA).

The remainder of this paper is organized as follows. Section 2 describes the problem formulations of the discrete SLP problem. Section 3 is used to present the proposed DE/AEDA algorithm in detail. Computational experiments are carried out to evaluate and validate the performance of the algorithm in Sect. 4. Finally, Sect. 5 draws some conclusions.

2 Problem statement

2.1 Problem description and assumptions

In this paper, the supermarket is a decentralized logistics area near the assembly line where parts are consumed and it is replenished by industrial trucks from the centralized warehouse. Parts are intermediately stored in the supermarket, and then transported to stations by tow trains on fixed routes based on the predefined schedule. There are $|S|$ stations along the mixed-model assembly lines. Due to the limited space in the assembly shop, the shop needs to be properly planned to determine the best supermarket layout.

In order to obtain a proper mathematical model, premises and assumptions are made as follows:

1. Cycle time is selected as the basic unit time for the system.
2. Supermarket serves consecutive stations, that is to say, the supply area of supermarket is no-overlapping.
3. Materials are stored by bins of uniform size, thus, stations' demand and the capacity of the supermarket are consequently represented by the number of bins.
4. As the planning of supermarkets is a long-term decision-making issue, the demand at each station is a pre-estimate.

2.2 Notations and model formulation

In this subsection, a mathematical model is established based on aforementioned assumptions. For the convenience of description, the notations and explanations are listed in Table 1.

Concerning the goal of SLP, the classic objective of minimizing the total cost including average operating cost per shift and transportation cost of tow trains is certainly applicable, which is also widely applied in many other literatures [6, 14]. Since the capacity of tow trains is rather limited, how many and which stations can be supplied in one tour is restrained. Not too many high-demand stations can be served in one tour, or else the tow trains will be overloaded. In this paper, it is also supposed that the

Table 1 Notations and explanation

| <i>Decision variables</i> | |
|---------------------------|---|
| o_g | Last station supplied by supermarket g |
| m | The optimal number of supermarkets after decision making |
| <i>Parameters</i> | |
| S | Set of stations to be assembled, indexed by s |
| G | Set of candidate supermarkets, indexed by g |
| x_s | x -coordinate of station s |
| y_s | y -coordinate of station s |
| d_s | Estimated demand for station s per shift (the number of bins) |
| a_g | x -coordinate of candidate supermarket g |
| b_g | y -coordinate of candidate supermarket g |
| e_g | Driving distance on the supply area served by candidate supermarket g |
| η | Expected utilization rate of supermarket |
| C | Capacity of each candidate supermarket (the number of bins) |
| Γ | Average operating cost per shift in candidate supermarket g |
| u | Cost of moving one unit distance for each bin |

capacity of the supermarket is sufficient to fulfill the total demand of its responsible supply area, because overworked supermarkets will have to make use of additional vehicles, routes and safety stock, all leading to higher operating cost. According to the above assumptions and notation definitions, the SLP problem is now formulated as follows: Objective function:

$$\text{Minimize } m \cdot \Gamma + u \cdot e_g \cdot \sum_{g=1}^m \sum_{s=o_{g-1}+1}^{o_g} d_s \quad (1)$$

Subject to

$$o_0 = 1 \quad (2)$$

$$o_{g-1} + 1 \leq o_g \quad \forall g \in G \quad (3)$$

$$o_m = |S| \quad (4)$$

$$\sum_{s=o_{g-1}+1}^{o_g} d_s / C \geq \eta \quad \forall g \in G \quad (5)$$

$$\sum_{s=o_{g-1}+1}^{o_g} d_s \leq C \quad \forall g \in G \quad (6)$$

where

$$e_g = \left(|x_{o_g} - x_{o_{g-1}+1}| + |y_{o_g} - y_{o_{g-1}+1}| \right) + \left(|x_{o_g} - a_g| + |y_{o_g} - b_g| \right) + \left(|a_g - x_{o_{g-1}+1}| + |b_g - y_{o_{g-1}+1}| \right) \quad (7)$$

$$\forall g \in G$$

Objective function (1) is to minimize the total cost, including average operating cost per shift and transportation cost of tow trains. Constraints from (2) to (4) indicate each supermarket is responsible for at least one station, and the supply area is no-overlapping. Constraint (5) ensures

that the utilization rate per supermarket is not lower than the expected utilization rate. Constraint (6) makes sure that the capacity of the supermarket is sufficient to meet the total demand of its responsible supply area. Considering driving lanes in assembly workshops are generally straight lines, the driving distance of tow train is calculated according to (7) by the Manhattan metric.

Since the SLP problem is an NP-complete problem, it is almost impossible to solve real-world scale problems by exact approach. Therefore, metaheuristic algorithms are needed to deal with the SLP problem.

3 DE/AEDA

As mentioned above, the EDA has been chosen as the main framework of the proposed optimization algorithm. The EDA is a novel type of group evolutionary algorithm and has been applied to solve many engineering optimization problems successfully in many fields. Unlike other meta heuristic algorithms, instead of generating new solutions by chromosome recombination, EDA describes the distribution information of candidate solutions in search space by establishing the probability model, based on which new solutions are sampled. It avoids the blindness and randomness caused by chromosome recombination, thus the search efficiency is effectively improved.

However, although EDA is good at global searching, it lacks the ability for refined search. Therefore, differential evolution (DE) strategy and adaptive learning rate mechanism are integrated into EDA to enhance the local search performance of algorithm, which is called the self-adaptive estimation of distribution algorithm with differential evolution strategy (DE/AEDA). The combination of EDA

algorithm and DE strategy can effectively promote the search ability of the algorithm. And the detailed descriptions of the proposed algorithm are presented as follows.

3.1 Encoding schemes

Encoding is the key step for EDA as it directly determines whether the probability model is appropriate or not. For the SLP problem studied in this paper, there are two encoding schemes as shown in Fig. 1.

Encoding scheme (a) is more applicable to algorithms that generate new individuals with crossover and mutation operators. It contains two sections—the first section (shaded) represents the last station served by each supermarket and the second section indicates the corresponding supermarket.

However, taking the characteristics of EDA algorithm into account, this paper applies (b) as the encoding scheme, under which the length of encoding stands for the total number of stations and each number here represents the supermarket which is in charge of the located station.

For example, as depicted in Fig. 1, there are 12 stations in total. Both schemes imply that stations 1–5 are served by supermarket 1. Stations 6–9 are served by supermarket 3, and the rest are in the charge of supermarket 4.

For the SLP problem of $|S|$ stations and $|G|$ candidate supermarkets, if the individual is encoded as $\pi = (\pi_1, \pi_2, \dots, \pi_{|S|})$, then it has following properties:

$$\begin{cases} \pi_s \in G, & \forall s \in S \\ \pi_{s+1} \geq \pi_s, & \forall s = 1, 2, \dots, |S| - 1 \end{cases} \quad (8)$$

3.2 Population initialization

The quality of the initial population is crucial for the convergence of algorithms and the quality of solutions. If each individual gene is randomly selected from $1, 2, \dots, |G|$, then there may be illegal individuals. Consequently, according to the characteristic formula (8) of individual encoding scheme and given population size N , the following population initialization algorithm is designed:

| | | | | | |
|---|---|----|---|---|---|
| 5 | 9 | 12 | 1 | 3 | 4 |
|---|---|----|---|---|---|

(a) Encoding scheme for traditional EAs

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 4 | 4 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|---|

(b) Encoding scheme for the EDA

Fig. 1 Encoding schemes for individuals

Step 1: Define station optional supermarket matrix as $\Omega = (\omega_{ij})_{|G| \times |S|}$, a 0–1 matrix, where $\omega_{ij} = 1$ indicates that the station j can be assigned to the supermarket i .

Step 2: Let $k = 1$.

Step 3: Make non-zero rows in the matrix Ω as $(r_1, \dots, r_b, \dots, r_B)$, and let the first value which meets $\pi_i = r_1$ be denoted as i_0 , while the k -th gene of an individual is marked as row_k . If $\sum_{s=i_0}^{k-1} d_s < \eta \cdot C$, that is, the supermarket utilization constraint has not been met, then $\text{row}_k = r_1$. While if $\sum_{s=i_0}^k d_s > C$, that is, capacities of supermarkets will be exceeded choosing r_1 , then row_k is a non-zero row randomly selected from (r_2, \dots, r_B) ; otherwise, row_k is a non-zero row randomly selected from $(r_1, \dots, r_b, \dots, r_B)$.

Step 4: Set elements in row $(\text{row}_k - 1)$ and column k to zero.

Step 5: $k \leftarrow k + 1$.

Step 6: Repeat Step 3–Step 5 until Ω comes to be a zero matrix.

Repeat Step 2–Step 6 above for N times to generate the initial population.

3.3 Probability model

The process of establishing the probability model plays a decisive role in designing the EDA. In this paper, $\mathbf{P}(\text{gen})$, a $|G| \times |S|$ -dimensional probability matrix, is applied to demonstrate the probability model for distribution of the solution space:

$$\mathbf{P}(\text{gen}) = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1|S|} \\ p_{21} & p_{22} & \cdots & p_{2|S|} \\ \cdots & \cdots & \cdots & \cdots \\ p_{|G|1} & p_{|G|2} & \cdots & p_{|G||S|} \end{pmatrix} \quad (9)$$

where $\mathbf{P}(\text{gen})$ is the probability matrix of the gen -th generation and $(p_{ij})_{|G| \times |S|}$ stands for the probability that station j is assigned to the supermarket i .

For the initial probability matrix $\mathbf{P}(\mathbf{0})$, the following uniform distribution is adopted to achieve the uniform sampling of the solution space:

$$p_{ij} = \frac{1}{|G|}, \quad i \in G, j \in S \quad (10)$$

In this paper, truncation selection method is used to select superior individuals. Select a number of SP individuals with smaller fitness values from the population, which is called SP population, where $\text{SP} = \lfloor \gamma \cdot N \rfloor$ and $\gamma \in (0, 1)$ is the selection rate.

Then, calculate the frequency matrix $\mathbf{F} = (f_{ij})_{|G| \times |S|}$ of SP population, where f_{ij} represents the frequency that station j is assigned to the supermarket i in the SP population. Moreover, the average fitness values of \bar{Z}_1 and \bar{Z}_2 of the

current population and SP population are calculated. Finally, based on the current probability matrix $\mathbf{P}(\text{gen})$ and the frequency matrix \mathbf{F} , the probability matrix is updated, in each generation.

3.4 Self-adaptive updating mechanism of probability matrix

After N new individuals are generated by sampling, a number of SP individuals are selected according to the fitness values to construct the dominant population. Then, the probability matrix is updated through the Hebb rule in machine learning incorporated with the adaptive strategy:

$$\mathbf{P}(\text{gen} + 1) = (1 - \alpha) \cdot \mathbf{P}(\text{gen}) + \alpha \cdot \mathbf{F} \tag{11}$$

$$\alpha = \alpha_0 \cdot e^{\frac{1}{\alpha_2 - \alpha_1}} \tag{12}$$

The updating process is essentially an incremental learning process, where α is the incremental learning rate, $0 < \alpha_0 < 1$ and $\alpha \leq \alpha_{\max}$, and $\alpha \cdot \mathbf{F}$ denotes the learning information obtained from the superior population. The updating formula of the probability matrix can adaptively adjust the learning rate in accordance with the fitness values in the population.

3.5 Sampling algorithm

If $\mathbf{P}(\text{gen}) = (p_{ij})_{|G| \times |S|}$ is the current probability matrix, based on Monte Carlo method, combined with the characteristics of the individual encoding in this paper, the sampling algorithm is constructed to obtain a new individual as follows:

Step 1: Let $\mathbf{W} = \mathbf{P}$. Then, select the k -th column of matrix \mathbf{W} , and let $k = 1$.

Step 2: A number of $|G|$ probability values of the k -th column of the matrix \mathbf{W} are accumulated in turn, and a number of $|G|$ cumulative probability values are thus obtained.

Step 3: $\kappa (0 < \kappa \leq 1)$ is randomly generated, and if the λ_k -th cumulative probability values is the one which is the smallest greater than κ , then the value of the k -th gene is λ_k .

Step 4: Let the elements of matrix \mathbf{W} be $w_{\lambda_k, k} = 0$ and $w_{ij} = 0 (i < \lambda_k \text{ and } j > k)$. And then the original probability is proportionally compensated for other sampling elements so as to ensure the sum of probabilities of optional supermarkets in matrix \mathbf{W} to be 1 during subsequent sampling process.

Step 5: $k \leftarrow k + 1$.

Step 6: If $k \leq |S|$, then go to Step 2, otherwise the algorithm ends and a new individual is gained through sampling.

3.6 A novel discrete DE strategy

If the optimal solution to the population is not updated for L consecutive generations in the AEDA algorithm, a novel discrete differential evolution strategy will be constructed for the sub-population with the fitness values ranking $\beta_1 \sim \beta_2$ in the current population to avoid falling into local optimum.

The discrete DE operator specifically includes three types of operators: separation, combination and mutation. Given specific differential control parameters $0 < \varphi_1 < \varphi_2 < 1$, the discrete DE strategy is operated as follows:

Step 1: Random number $\kappa (0 < \kappa \leq 1)$ is randomly generated.

Step 2: If $0 < \kappa \leq \varphi_1$, separation operator is performed as shown in Fig. 2a; if $\varphi_1 < \kappa \leq \varphi_2$, combination operator is conducted as shown in Fig. 2b; otherwise mutation operator is carried out as shown in Fig. 2c.

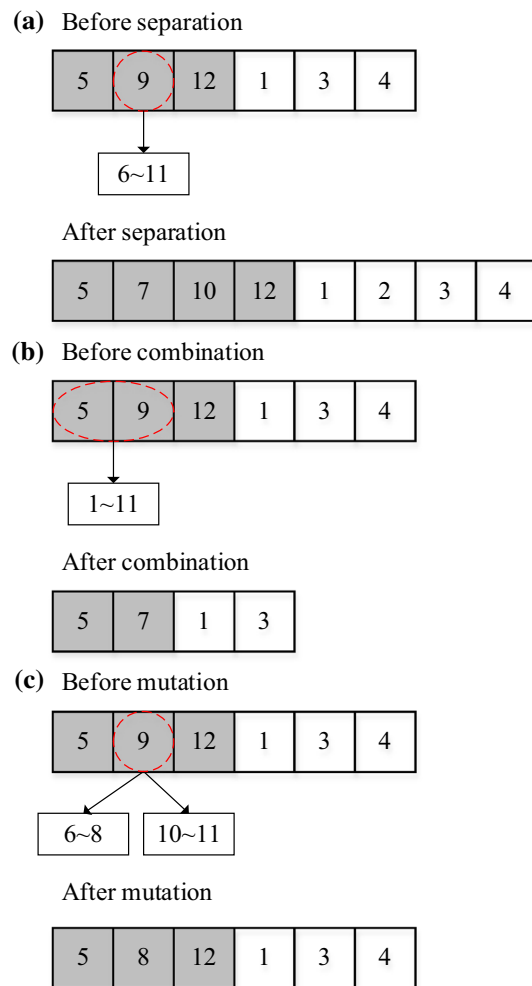


Fig. 2 Illustration of the novel discrete DE operators

Step 3: Metropolis acceptance criteria is applied to select individuals after carrying out the discrete DE strategy.

Step 4: End.

3.7 Overall procedure of DE/AEDA algorithm

Based on detailed descriptions in Sects. 3.1–3.6, the overall procedure of DE/AEDA algorithm proposed to solve the SLP is illustrated in Fig. 3.

4 Simulation and results

In this section, in order to analyze the performance of the proposed DE/AEDA, simulation experiments are carried out. All the algorithms are run on a personal PC with an Intel(R) Core(TM) i5-4200M CPU clocked at 2.50 GHz with 4 GB RAM. The codes are run on Matlab (2014a) platform.

4.1 Datasets

In order to verify the effectiveness of the DE/AEDA algorithm, the test examples $(|S|, \Gamma)$ are constructed as follows referring to Refs. [6, 14].

$|S| \in \{20, 40, 60, 100, 150, 200\}$, the number of stations, in each case corresponds to 6 different supermarket configurations, and the average operating cost per shift in the supermarket is $\Gamma \in \{50, 100, 150, 300, 500\}$. Consequently, 24 test cases are constructed in total. The y -coordinates of each station and each candidate are set to $y_s = 1$, $b_g = 5$ respectively. While the supermarkets are randomly distributed among the stations, that is, $a_g = \text{rand}(1, 6 \cdot (|S| - 1))$. And the distance between two consecutive stations is set to 6, namely $x_{s+1} - x_s = 6$, which guarantees the length of a car. The estimated demand for station s per shift follows a uniform distribution $U(40, 100)$.

4.2 Parameters setting

The main parameters of the proposed DE/AEDA algorithm include the population size N , the selection rate of SP population r and the initial learning rate α_0 . A medium-scale test case (60, 150), is selected to implement design of experiments (DOE) to explore the influence of these parameters on the performance of the DE/AEDA algorithm.

Since each factor consists of four levels, the orthogonal array $L_{16}(4^3)$ is employed. The detailed configuration of these parameters and the orthogonal array are listed in

Tables 2 and 3 respectively. For each test case, the DE/AEDA runs 20 times independently. The average result obtained by the DE/AEDA is selected as response variable, denoted as RV. The results are shown in Table 4. In addition, Fig. 4 depicts the trend of the parameters with different levels.

As shown in Table 3, the range of initial learning rate α_0 ranks first, which indicates that α_0 is the most significant parameter to the performance of the DE/AEDA, followed by the selection rate of SP population r , the last is population size N . It can be known from Fig. 4 that each parameter ought to be set reasonably. If α_0 is too small, the updating of probability matrix is slowly, resulting in weakening global searching ability of the algorithm. On the contrary, if α_0 is too big, the updating process will be rather rapid, leading to the lack of refined search process. As for the selection rate of SP population r , small r will lead to premature convergence of the population, while the large one will make the population have a slow convergence. Although the population size N has a minimal effect on the algorithm performance, it is still necessary to find a suitable value to achieve the optimal performance of the algorithm.

In summary, the configuration of parameters is set to $N = 150$, $r = 0.2$ and $\alpha_0 = 0.10$, which is a good choice to the proposed DE/AEDA.

4.3 Comparison experiment and analysis

Traditional evolutionary algorithms generate new candidates by chromosome recombination, including crossover and mutation operators. However, in the EDA algorithm, new solutions are sampled from the constructed probability model. In order to verify the performance of the proposed DE/AEDA, a benchmark algorithm named RGA (Real Genetic Algorithm) from Ref. [14] has been involved.

For each test case, each algorithm $a(a \in \{\text{DE/AEDA}, \text{RGA}\})$ will run 10 times independently. Then, statistical analysis is conducted to the results. The best value, the worst value and the calculated average value obtained in these runs by algorithm a are denoted respectively as $TZ_b(a)$, $TZ_w(a)$, $TZ_{av}(a)$. On this basis, the following variables are constructed:

$$RTZ_b(a) = TZ_b(a)/TZ_b(\text{DE/AEDA}) \quad (13)$$

$$RTZ_w(a) = TZ_w(a)/TZ_w(\text{DE/AEDA}) \quad (14)$$

$$RTZ_{av}(a) = TZ_{av}(a)/TZ_{av}(\text{DE/AEDA}) \quad (15)$$

which stands for the relative ratio of the result obtained by the algorithm a to the best value of the DE/AEDA algorithm [33]. Since the results don't necessarily obey the normal distribution, then the Wilcoxon rank-sum test, which has been increasingly applied for the intelligent

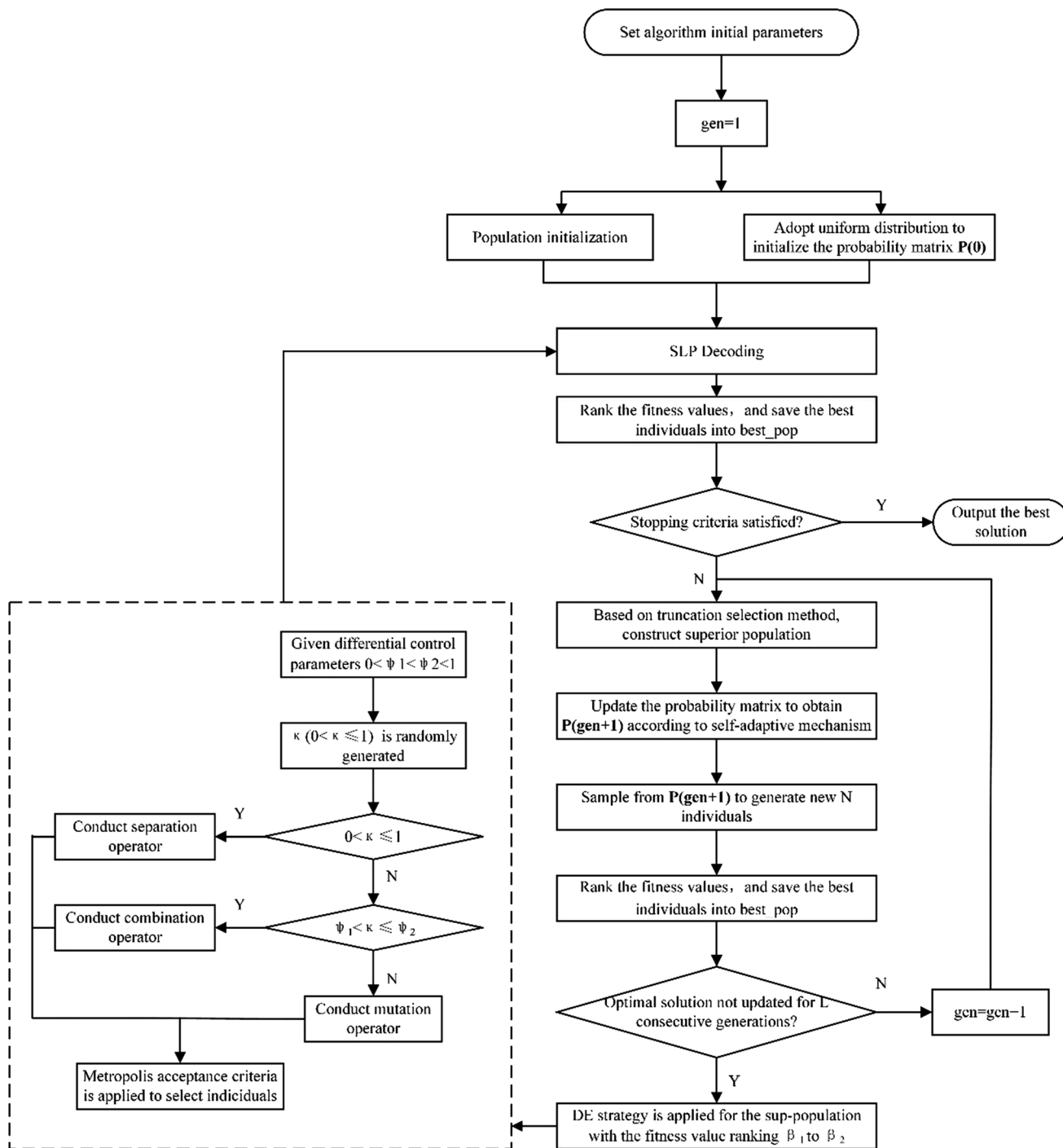


Fig. 3 Flowchart of the DE/AEDA algorithm

computing research [34], is conducted to verify whether the performance of DE/AEDA algorithm is significantly superior to other algorithms.

Rank values R are divided into positive values R^+ and negative values R^- . Let T^+ be the sum of ranks for positive values and T^- the sum of ranks for the opposite, namely $T^+ = \sum R^+$, $T^- = \sum R^-$. Then, $W = \min(T^+, T^-)$, the smaller of the sums is calculated, defined as the ranks-

related variable. The null hypothesis is that there is no statistical difference between DE/AEDA and RGA. Due to 10 independent runs for each algorithm, then $n = 10$. When the null hypothesis can be rejected at the 0.05 level, the critical value under two-tailed test is $U = 8$. Therefore, if $W < U$, it indicates that the performance of DE/AEDA is significantly superior to the other one. The difference between the worst value and the best value obtained in each

Table 2 Configuration of parameter values

| Factors | Level | | | |
|------------|-------|------|------|------|
| | 1 | 2 | 3 | 4 |
| N | 50 | 100 | 150 | 200 |
| r | 0.1 | 0.2 | 0.3 | 0.4 |
| α_0 | 0.02 | 0.06 | 0.10 | 0.14 |

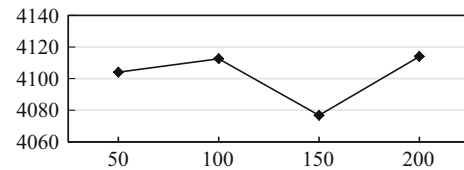
Table 3 Orthogonal table and RV values

| No. | Factors | | | RV |
|-----|---------|-----|------------|---------|
| | N | r | α_0 | |
| 1 | 50 | 0.1 | 0.02 | 4197.47 |
| 2 | 50 | 0.2 | 0.06 | 4007.46 |
| 3 | 50 | 0.3 | 0.10 | 4091.53 |
| 4 | 50 | 0.4 | 0.14 | 4119.87 |
| 5 | 100 | 0.1 | 0.06 | 4065.4 |
| 6 | 100 | 0.2 | 0.02 | 4148.63 |
| 7 | 100 | 0.3 | 0.14 | 4212.73 |
| 8 | 100 | 0.4 | 0.10 | 4023.47 |
| 9 | 150 | 0.1 | 0.10 | 4058.28 |
| 10 | 150 | 0.2 | 0.14 | 4045.61 |
| 11 | 150 | 0.3 | 0.02 | 3999.21 |
| 12 | 150 | 0.4 | 0.06 | 4203.87 |
| 13 | 200 | 0.1 | 0.14 | 4121.41 |
| 14 | 200 | 0.2 | 0.10 | 4096.51 |
| 15 | 200 | 0.3 | 0.06 | 4199.04 |
| 16 | 200 | 0.4 | 0.02 | 4039.09 |

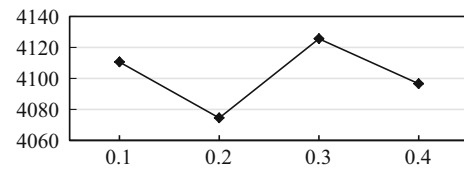
Table 4 Statistical analysis of experimental results

| Level | N | r | α_0 |
|-----------------------|-----------|-----------|------------|
| 1 | 4104.0825 | 4110.64 | 4096.1 |
| 2 | 4112.5575 | 4074.5525 | 4118.9425 |
| 3 | 4076.7425 | 4125.6275 | 4067.4475 |
| 4 | 4114.0125 | 4096.575 | 4124.905 |
| Range R | 37.27 | 51.075 | 57.4575 |
| Rank of major factors | 3 | 2 | 1 |
| Optimal level | 3 | 2 | 3 |
| Optimal configuration | 150 | 0.2 | 0.10 |

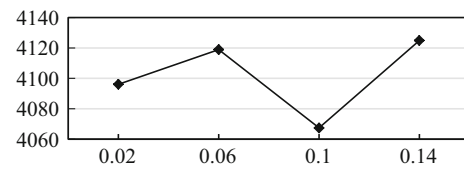
test case denoted as Δ_{wb} , which is calculated by $\Delta_{wb} = TZ_w - TZ_b$, can be regarded as an indicator to measure the stability of the algorithm.



(a) Population size



(b) Selection rate of SP population



(c) Initial learning rate

Fig. 4 Factor level trend of the parameters

According to the statistical analysis results in Table 5, among the 24 test cases, there are 21 cases which demonstrate that the results obtained by the DE/AEDA are significantly better than those of RGA. Besides, it can be observed from Fig. 5 that the Δ_{wb} values of DE/AEDA are mainly distributed within the interval (6%, 10%), while the Δ_{wb} values of RGA are mainly distributed within the interval (7%, 17%). Therefore, it can be concluded that the DE/AEDA algorithm is more stable and robust than the RGA algorithm.

Consequently, the experimental results show that the performance of DE/AEDA algorithm is superior to that of RGA, resulting in solving SLP problem more effectively. The superiority of DE/AEDA algorithm mainly originates from two aspects: one is the application of the probability model which avoids the randomness and blindness caused by chromosome recombination, including crossover and mutation. The other is the random searching ability of the probability model and local optimization ability from the hybrid of the discrete DE operators.

4.4 Cost function and parameter analysis

Apart from the comparison experiments which are carried out in Sect. 4.3, the proposed DE/AEDA is compared with RGA by the objective function of SLP in this part. Table 6 shows the detailed results for all station counts, averaged over all 30 runs per station count S . All instances were tested with four different fixed cost values Γ , hence giving

Table 5 Comparison between DE/AEDA and RGA

| (S , F) | DE/AEDA | | | RGA | | | W | Whether significantly |
|------------|---------|---------|------------|---------|---------|------------|------|-----------------------|
| | RTZ_b | RTZ_w | RTZ_{av} | RTZ_b | RTZ_w | RTZ_{av} | | |
| (20, 100) | 1.0000 | 1.0020 | 1.0007 | 1.0000 | 1.0120 | 1.0090 | 22.5 | N |
| (20, 150) | 1.0000 | 1.0103 | 1.0083 | 1.0000 | 1.0351 | 1.0174 | 17 | N |
| (20, 200) | 1.0000 | 1.0258 | 1.0070 | 1.0000 | 1.0724 | 1.0301 | 7 | Y |
| (20, 300) | 1.0000 | 1.0312 | 1.0128 | 0.9996 | 1.0584 | 1.0279 | 21 | N |
| (40, 100) | 1.0000 | 1.0519 | 1.0279 | 1.0133 | 1.1377 | 1.0514 | 5 | Y |
| (40, 150) | 1.0000 | 1.0622 | 1.0301 | 1.0376 | 1.1380 | 1.0729 | 6 | Y |
| (40, 200) | 1.0000 | 1.0587 | 1.0393 | 1.0504 | 1.1550 | 1.1044 | 5 | Y |
| (40, 300) | 1.0000 | 1.0657 | 1.0467 | 1.0475 | 1.1442 | 1.1166 | 7 | Y |
| (60, 100) | 1.0000 | 1.1026 | 1.0526 | 1.0512 | 1.1696 | 1.1015 | 7 | Y |
| (60, 150) | 1.0000 | 1.1051 | 1.0534 | 1.0751 | 1.1893 | 1.0982 | 5 | Y |
| (60, 200) | 1.0000 | 1.0973 | 1.0488 | 1.0716 | 1.2073 | 1.1086 | 4 | Y |
| (60, 300) | 1.0000 | 1.1055 | 1.0535 | 1.0613 | 1.1930 | 1.1124 | 5 | Y |
| (100, 100) | 1.0000 | 1.1002 | 1.0618 | 1.0655 | 1.1707 | 1.1118 | 3 | Y |
| (100, 150) | 1.0000 | 1.0951 | 1.0562 | 1.0765 | 1.1755 | 1.1403 | 1 | Y |
| (100, 200) | 1.0000 | 1.0868 | 1.0514 | 1.0748 | 1.1915 | 1.1174 | 3 | Y |
| (100, 300) | 1.0000 | 1.0926 | 1.0448 | 1.0700 | 1.1683 | 1.1034 | 3 | Y |
| (150, 100) | 1.0000 | 1.0970 | 1.0540 | 1.0607 | 1.1756 | 1.1207 | 6 | Y |
| (150, 150) | 1.0000 | 1.0948 | 1.0521 | 1.0636 | 1.1926 | 1.1269 | 3 | Y |
| (150, 200) | 1.0000 | 1.0883 | 1.0530 | 1.0671 | 1.2360 | 1.1302 | 3 | Y |
| (150, 300) | 1.0000 | 1.0917 | 1.0511 | 1.0707 | 1.2127 | 1.1224 | 2 | Y |
| (200, 100) | 1.0000 | 1.0867 | 1.0483 | 1.0657 | 1.2297 | 1.1242 | 3 | Y |
| (200, 150) | 1.0000 | 1.0851 | 1.0478 | 1.0659 | 1.2075 | 1.1313 | 6 | Y |
| (200, 200) | 1.0000 | 1.0912 | 1.0571 | 1.0694 | 1.2225 | 1.1342 | 4 | Y |
| (200, 300) | 1.0000 | 1.0831 | 1.0435 | 1.0723 | 1.2209 | 1.1407 | 3 | Y |

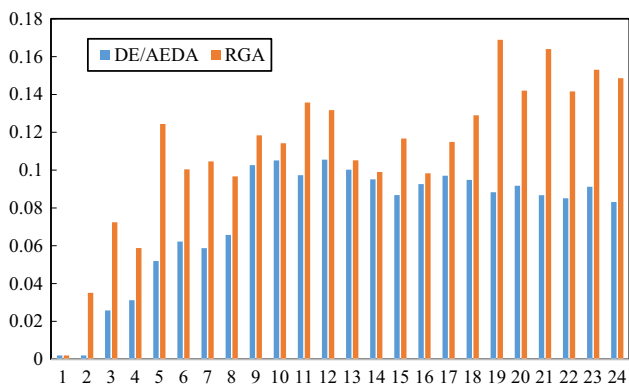


Fig. 5 Stability comparison between DE/AEDA and RGA

monetary value to the basic trade-off of fixed cost and transportation cost. “ m ” denotes the number of supermarkets in the resulting solution for the given station count and fixed cost, averaged over the 30 instances. “ F^* ” denotes the total transportation cost, as calculated by the latter item in the objective function [Eq. (1)]. “ F ” stands for the optimal SLP objective value, which exactly equals the total operation cost. And “CPU time” is the average time in seconds taken by the algorithm per instance.

Expectedly, the lower the fixed cost F per supermarket, the more supermarkets are installed and the lower the corresponding SLP objective value F . This is, however, not true when fixed costs are very low and hence many supermarkets erected. This can be explained by the fact that every supermarket needs at least one vehicle stationed at it, which also entails fixed cost. At some point, the cost of adding another supermarket to an already large supermarket set is too high to be offset by the only marginally more transportation cost reduction it enables. This leads to a conclusion that even a small number of supermarkets is very efficient in supplying an assembly line, while too many logistics areas can quickly become counterproductive.

On a performance note, the propose DE/AEDA algorithm appears to run quite fast and seems adequate for problems of realistic size, as it could solve all instances in acceptable CPU time. Also, DE/AEDA has achieved a better performance on the objective function of SLP. In Table 6, optimal solutions are bolded and it can be seen that the propose DE/AEDA algorithm outperforms RGA in most cases (18 of 24 cases). Therefore, it can be concluded

Table 6 Cost function comparison between DE/AEDA and RGA

| S | Γ | DE/AEDA | | | | RGA | | | |
|-----|--------|----------|------------|-------------------|--------|----------|------------|-------------------|--------|
| | | <i>m</i> | <i>F*</i> | <i>F</i> | CPU(s) | <i>m</i> | <i>F*</i> | <i>F</i> | CPU(s) |
| 10 | 500 | 4.75 | 3038.91 | 5412.17 | 0.12 | 4.80 | 2994.35 | 5394.24 | 0.11 |
| 10 | 1000 | 4.05 | 4762.50 | 8815.28 | 0.07 | 4.10 | 4735.86 | 8832.66 | 0.07 |
| 10 | 5000 | 2.32 | 12,212.07 | 23,804.83 | 0.19 | 2.32 | 12,148.96 | 23,769.36 | 0.18 |
| 10 | 10,000 | 2.25 | 17,620.54 | 40,162.91 | 0.06 | 2.21 | 18,249.41 | 40,307.45 | 0.06 |
| 30 | 500 | 11.79 | 8774.25 | 14,666.82 | 0.59 | 11.79 | 9257.50 | 15,151.24 | 0.58 |
| 30 | 1000 | 8.98 | 13,692.72 | 22,673.56 | 0.36 | 8.97 | 14,369.51 | 23,341.82 | 0.38 |
| 30 | 5000 | 5.07 | 36,161.87 | 61,501.59 | 0.27 | 5.14 | 34,667.67 | 60,381.40 | 0.27 |
| 30 | 10,000 | 3.72 | 53,550.66 | 90,774.16 | 0.19 | 3.79 | 54,602.38 | 92,502.31 | 0.18 |
| 60 | 500 | 16.96 | 17,592.33 | 26,073.57 | 2.50 | 16.97 | 18,130.66 | 26,615.66 | 2.46 |
| 60 | 1000 | 8.59 | 27,524.36 | 36,111.93 | 2.08 | 8.62 | 27,706.93 | 36,326.23 | 2.16 |
| 60 | 5000 | 6.49 | 71,515.39 | 103,948.28 | 1.90 | 6.55 | 71,059.52 | 103,813.43 | 1.90 |
| 60 | 10,000 | 4.63 | 105,368.22 | 151,633.29 | 1.51 | 4.65 | 110,403.72 | 156,885.87 | 1.42 |
| 100 | 500 | 26.59 | 28,846.75 | 42,140.30 | 34.23 | 26.64 | 29,908.30 | 43,225.94 | 36.90 |
| 100 | 1000 | 13.31 | 44,070.51 | 57,381.72 | 29.23 | 13.30 | 44,915.98 | 58,213.34 | 27.62 |
| 100 | 5000 | 9.88 | 117,199.40 | 166,581.77 | 22.29 | 9.95 | 122,549.33 | 172,283.51 | 21.70 |
| 100 | 10,000 | 6.46 | 172,636.78 | 237,272.44 | 21.69 | 6.46 | 170,529.68 | 235,167.22 | 20.92 |
| 150 | 500 | 19.97 | 41,091.42 | 51,074.09 | 39.09 | 19.96 | 43,904.83 | 53,885.49 | 38.59 |
| 150 | 1000 | 14.52 | 68,886.41 | 83,401.90 | 24.93 | 14.54 | 71,762.79 | 86,301.44 | 25.46 |
| 150 | 5000 | 9.57 | 180,124.52 | 227,977.59 | 22.39 | 9.62 | 177,657.56 | 225,773.25 | 21.56 |
| 150 | 10,000 | 7.17 | 235,527.23 | 307,248.17 | 14.15 | 7.26 | 264,255.99 | 336,847.48 | 14.71 |
| 200 | 500 | 25.77 | 58,679.87 | 71,563.00 | 98.33 | 25.80 | 60,100.18 | 73,002.40 | 98.81 |
| 200 | 1000 | 18.87 | 91,445.19 | 110,318.02 | 72.31 | 19.02 | 94,016.64 | 113,037.40 | 69.94 |
| 200 | 5000 | 12.28 | 238,685.50 | 300,076.33 | 65.98 | 12.35 | 249,275.29 | 311,047.17 | 69.37 |
| 200 | 10,000 | 8.98 | 352,193.24 | 441,960.01 | 65.34 | 8.91 | 365,783.20 | 454,897.14 | 63.28 |

|S|: number of stations; Γ: average operating cost per shift; *m*: number of supermarkets; *F**: total transportation cost; *F*: total cost (Objective function); CPU: average CPU time

that DE/AEDA is superior to RGA in obtaining better results.

4.5 Validation of the operators

In order to verify the validity of self-adaptive learning rate and DE operators, based on the control variable method, DE/AEDA is compared to AEDA and DE/EDA respectively. For each test case, each algorithm $a(a \in \{DE/AEDA, DE/EDA, AEDA\})$ runs 10 times independently. The average computation time for obtaining the optimal solution is selected as setting time. In the end, the results are statistically analysed. The aforementioned non-parametric test method is also applied to analyze the results of each test case. Table 7 shows the statistical analysis values of the DE/EDA algorithm and the AEDA algorithm. It is obvious that among the 24 test cases, there are 22 cases showing that the results obtained by the DE/AEDA are significantly better than that of DE/EDA. And for the AEDA algorithm, there are 21 cases in total. Thus, it illustrates the impact of self-adaptive learning rate and

DE operators on the performance of the algorithm and its effectiveness respectively.

From Figs. 6, 7 and 8, the distribution of the statistical results of 24 test cases in different metrics can be illustrated directly. As depicted in Fig. 6, the corresponding RTZ_b value of the AEDA algorithm is larger than that of the other two algorithms, which indicates that DE operator has a stronger influence on refined search ability of the algorithm. Figure 7 depicts that the impact of self-adaptive learning rate is even greater in terms of the average performance. As can be seen from Fig. 8, there is no significant difference in the stability of the 3 algorithms, which can be referred to probability model of the EDA algorithm for statistical learning. As a result, self-adaptive learning rate and DE operators can help improve the performance of the EDA algorithm in different aspects.

Table 7 Statistical analysis of comparison among DE/EDA, AEDA and DE/AEDA algorithms

| (S , T) | DE/EDA | | | | | AEDA | | | | |
|------------|---------|---------|------------|-----|---|---------|---------|------------|-----|---|
| | RTZ_b | RTZ_w | RTZ_{av} | W | # | RTZ_b | RTZ_w | RTZ_{av} | W | # |
| (20, 100) | 1.0000 | 1.0104 | 1.0081 | 6 | Y | 1.0000 | 1.0091 | 1.0043 | 9 | N |
| (20, 150) | 1.0039 | 1.0188 | 1.0158 | 6 | Y | 1.0054 | 1.0156 | 1.0112 | 7 | Y |
| (20, 200) | 1.0094 | 1.0348 | 1.0261 | 7 | Y | 1.0124 | 1.0358 | 1.0187 | 5 | Y |
| (20, 300) | 1.0137 | 1.0517 | 1.0255 | 4 | Y | 1.0096 | 1.0552 | 1.0196 | 7 | Y |
| (40, 100) | 1.0275 | 1.0738 | 1.0490 | 5 | Y | 1.0174 | 1.0891 | 1.0375 | 6 | Y |
| (40, 150) | 1.0374 | 1.0979 | 1.0617 | 5 | Y | 1.0376 | 1.0953 | 1.0398 | 5 | Y |
| (40, 200) | 1.0339 | 1.1152 | 1.0696 | 6 | Y | 1.0493 | 1.1049 | 1.0342 | 15 | N |
| (40, 300) | 1.0386 | 1.1188 | 1.0705 | 10 | N | 1.0557 | 1.1191 | 1.0567 | 4 | Y |
| (60, 100) | 1.0441 | 1.1491 | 1.0784 | 7 | Y | 1.0588 | 1.1503 | 1.0648 | 6 | Y |
| (60, 150) | 1.0388 | 1.1464 | 1.0759 | 4 | Y | 1.0543 | 1.1458 | 1.0597 | 4 | Y |
| (60, 200) | 1.0454 | 1.1405 | 1.0851 | 3 | Y | 1.0636 | 1.1464 | 1.0634 | 3 | Y |
| (60, 300) | 1.0407 | 1.1437 | 1.0946 | 5 | Y | 1.0623 | 1.1615 | 1.0581 | 9 | N |
| (100, 100) | 1.0472 | 1.1308 | 1.0885 | 5 | Y | 1.0597 | 1.1373 | 1.0485 | 2 | Y |
| (100, 150) | 1.0537 | 1.1671 | 1.1063 | 6 | Y | 1.0549 | 1.1526 | 1.0724 | 7 | Y |
| (100, 200) | 1.0473 | 1.1594 | 1.0977 | 4 | Y | 1.0571 | 1.1542 | 1.0686 | 5 | Y |
| (100, 300) | 1.0454 | 1.1385 | 1.0842 | 6 | Y | 1.0607 | 1.1293 | 1.0521 | 4 | Y |
| (150, 100) | 1.0484 | 1.1687 | 1.0933 | 5 | Y | 1.0622 | 1.1574 | 1.0675 | 4 | Y |
| (150, 150) | 1.0395 | 1.1574 | 1.0921 | 5 | Y | 1.0583 | 1.1471 | 1.0693 | 3 | Y |
| (150, 200) | 1.0512 | 1.1518 | 1.0996 | 2 | Y | 1.0643 | 1.1506 | 1.0574 | 6 | Y |
| (150, 300) | 1.0477 | 1.1466 | 1.0848 | 3 | Y | 1.0721 | 1.1459 | 1.0732 | 2 | Y |
| (200, 100) | 1.0581 | 1.1493 | 1.0790 | 6 | Y | 1.0544 | 1.1389 | 1.0698 | 3 | Y |
| (200, 150) | 1.0469 | 1.1377 | 1.0748 | 5 | Y | 1.0651 | 1.1457 | 1.0581 | 6 | Y |
| (200, 200) | 1.0441 | 1.1645 | 1.1053 | 15 | N | 1.0703 | 1.1585 | 1.0724 | 3 | Y |
| (200, 300) | 1.0527 | 1.1768 | 1.0967 | 6 | Y | 1.0669 | 1.1581 | 1.0677 | 4 | Y |

“#” means “whether is significantly”

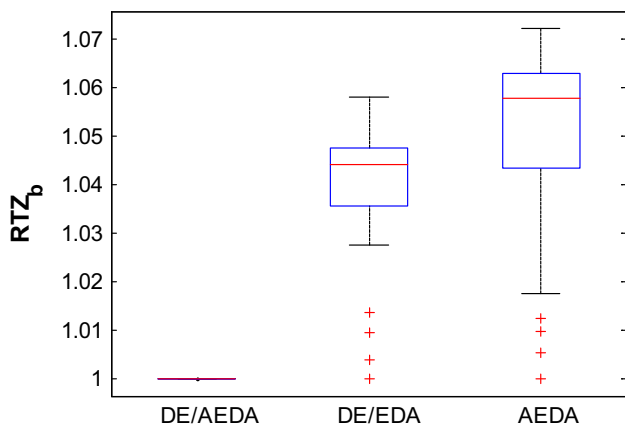


Fig. 6 Box plots of 24 test cases in RTZ_b metric

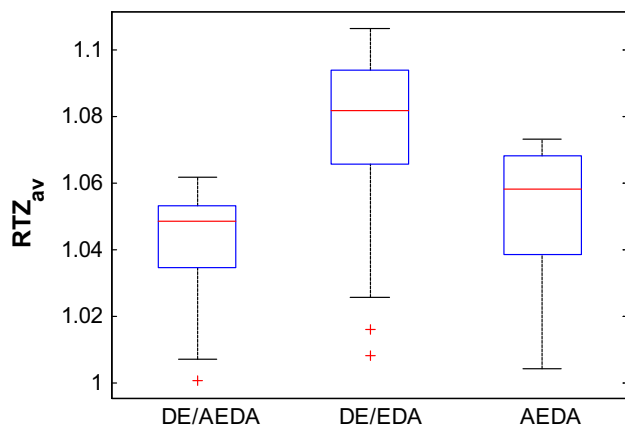


Fig. 7 Box plots of 24 test cases in RTZ_{av} metric

5 Conclusions

In this paper, a novel EDA algorithm, named self-adaptive estimation of distribution algorithm with differential evolution strategy (DE/AEDA) is proposed to solve the discrete SLP problem with the consideration of the utilization rate and capacity constraint of the supermarkets

simultaneously. Unlike traditional evolutionary algorithms, instead of generating new solutions by chromosome recombination, in the EDA algorithm, new solutions are sampled from the constructed probability model, so EDA is good at global searching. Considering EDA lacks the ability for refined search, the DE strategy and adaptive learning rate mechanism are integrated into EDA. The

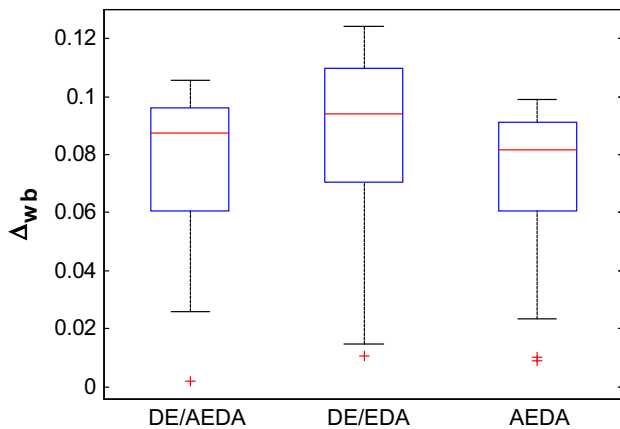


Fig. 8 Box plots of 24 test cases in Δ_{wb} metric

Wilcoxon rank-sum test method is carried out to verify the performance of the proposed algorithm. From the comparisons of the results, it can be concluded that the performance of DE/AEDA is superior to that of other algorithms, resulting in solving SLP problem more effectively.

Finally, the supermarket location problem is of great practical significance and deserves deeper investigation in the future. First, it is supposed in this paper that supermarkets can be located anywhere on the assembly floor. However, if only very specific areas on the shop floor are available for construction of a supermarket, then discrete location planning approaches should be involved to solve the problem. Second, other objectives can be considered and factors to be optimized in multi-objective scenarios are to be studied in the location decision problem. For example, the effect of number of supermarkets on the safety stock can be useful in many lean manufacturing situations. Moreover, it is also of great managerial significance to involve the total weighted distance of the routes applied by respective supermarkets through all the stations and back. In case the same parts are used at multiple stations, it may be efficient to have these stations served by one supermarket. Considering their interdependencies with other corporate objectives, SLP problems will remain a fruitful field of research in the foreseeable future.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant No. 71471135.

Funding The authors appreciate the supports to this research from the National Natural Science Foundation of China under Grant No. 71471135.

Compliance with ethical standards

Conflict of interest No potential conflict of interest was reported by the authors.

References

- Shahabudeen P, Gopinath R, Krishnaiah K (2002) Design of bi-criteria kanban system using simulated annealing technique. *Comput Ind Eng* 41(4):355–370
- Wang J, Chang Q, Xiao G, Wang N, Li S (2011) Data driven production modeling and simulation of complex automobile general assembly plant. *Comput Ind* 62(7):765–775
- Battini D, Faccio M, Persona A, Sgarbossa F (2009) Framework to optimise the inventory centralisation/decentralisation degree and feeding policy in assembly systems. *Int J Serv Oper Manag* 6(2):184–205
- Battini D, Boysen N, Emde S (2013) Just-in-Time supermarkets for part supply in the automobile industry. *J Manag Control* 24(2):209–217
- Emde S, Gendreau M (2017) Scheduling in-house transport vehicles to feed parts to automotive assembly lines. *Eur J Oper Res* 260(1):255–267
- Emde S, Boysen N (2012) Optimally locating in-house logistics areas to facilitate JIT-supply of mixed-model assembly lines. *Int J Prod Econ* 135(1):393–402
- Vaidyanathan BS, Matson JO, Miller DM, Matson JE (1999) A capacitated vehicle routing problem for just-in-time delivery. *IIE Trans* 31(11):1083–1092
- Emde S, Boysen N (2012) Optimally routing and scheduling tow trains for JIT-supply of mixed-model assembly lines. *Eur J Oper Res* 217(2):287–299
- Emde S, Fliedner M, Boysen N (2012) Optimally loading tow trains for just-in-time supply of mixed-model assembly lines. *IIE Trans* 44(2):121–135
- Golz J, Gujjula R, Günther HO, Rinderer S, Ziegler M (2012) Part feeding at high-variant mixed-model assembly lines. *Flex Serv Manuf J* 24(2):119–141
- Fathi M, Alvarez MJ, Hassanimehraban F, Rodríguez V (2014) A multiobjective optimization algorithm to solve the part feeding problem in mixed-model assembly lines. *Math Probl Eng* 2014(1):1–12
- Fathi M, Rodríguez V, Alvarez MJ (2014) A novel memetic ant colony optimization-based heuristic algorithm for solving the assembly line part feeding problem. *Int J Adv Manuf Technol* 75(1–4):629–643
- Fathi M, Rodríguez V, Fontes DB, Alvarez MJ (2016) A modified particle swarm optimisation algorithm to solve the part feeding problem at assembly lines. *Int J Prod Res* 54(3):878–893
- Alnahhal M, Noche B (2015) A genetic algorithm for supermarket location problem. *Assem Autom* 35(1):122–127
- Herda M (2015) Combined genetic algorithm for capacitated p-median problem. In: *Computational intelligence and informatics (CINTI), 2015 16th IEEE international symposium on*, IEEE, pp 151–154
- Arqub OA, Abo-Hammour Z (2014) Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. *Inf Sci* 279:396–415
- Abo-Hammour Z, Abu Arqub O, Momani S, Shawagfeh N (2014) Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic algorithm. *Discrete Dyn Nat Soc* 2014:1–15
- Abo-Hammour Z, Arqub OA, Alsmadi O, Momani S, Alsaedi A (2014) An optimization algorithm for solving systems of singular boundary value problems. *Appl Math Inf Sci* 8(6):2809–2821
- Zhou BH, Shen CY (2018) Multi-objective optimization of material delivery for mixed model assembly lines with energy consideration. *J Clean Prod* 192:293–305
- Reese J (2006) Solution methods for the p-median problem: an annotated bibliography. *Networks* 48(3):125–142

21. Mladenović N, Brimberg J, Hansen P, Moreno-Pérez JA (2007) The p-median problem: a survey of metaheuristic approaches. *Eur J Oper Res* 179(3):927–939
22. Correa ES, Steiner MTA, Freitas AA, Carnieri C (2004) A genetic algorithm for solving a capacitated p-median problem. *Numer Algorithms* 35(2–4):373–388
23. França PM, Sosa NM, Pureza V (1999) An adaptive tabu search algorithm for the capacitated clustering problem. *Int Trans Oper Res* 6(6):665–678
24. Yaghini M, Karimi M, Rahbar M (2013) A hybrid metaheuristic approach for the capacitated p-median problem. *Appl Soft Comput* 13(9):3922–3930
25. Landa-Torres I, Del Ser J, Salcedo-Sanz S, Gil-Lopez S, Portilla-Figueroa JA, Alonso-Garrido O (2012) A comparative study of two hybrid grouping evolutionary techniques for the capacitated P-median problem. *Comput Oper Res* 39(9):2214–2222
26. Mühlenbein H, Paass G (1996) From recombination of genes to the estimation of distributions I. Binary parameters. In: *International conference on parallel problem solving from nature*, Springer, Berlin, Heidelberg, pp 178–187
27. Zhou BH, Hu LM, Zhong ZY (2018) A hybrid differential evolution algorithm with estimation of distribution algorithm for reentrant hybrid flow shop scheduling problem. *Neural Comput Appl* 30(1):193–209
28. Zhang Q, Sun J, Tsang E, Ford J (2006) Estimation of distribution algorithm with 2-opt local search for the quadratic assignment problem. In: *Towards a new evolutionary computation*, Springer, Berlin, Heidelberg, pp 281–292
29. Tiwari A, Chang PC, Tiwari MK, Kollanoor NJ (2015) A Pareto block-based estimation and distribution algorithm for multi-objective permutation flow shop scheduling problem. *Int J Prod Res* 53(3):793–834
30. Zhou BH, Tan F (2018) Electric vehicle handling routing and battery swap station location optimization for automotive assembly lines. *Int J Comput Integr Manuf* 31(10):978–991
31. Wang L, Fang C (2012) An effective estimation of distribution algorithm for the multi-mode resource-constrained project scheduling problem. *Comput Oper Res* 39(2):449–460
32. Aickelin U, Burke EK, Li J (2007) An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *J Oper Res Soc* 58(12):1574–1585
33. Zhang R, Song S, Wu C (2013) A hybrid artificial bee colony algorithm for the job shop scheduling problem. *Int J Prod Econ* 141(1):167–178
34. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.