applied sciences

MDPI

Article

A Facility Layout Algorithm for Logistics Scenarios Driven by Transport Lines

Fulin Jiang [1], Lin Li [1,2,*], Yiming Tang [1], Hailong Zhang [1] and Xiaoping Liu [1,2]

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China; funny@mail.hfut.edu.cn (F.J.); ymtang@hfut.edu.cn (Y.T.); liu@hfut.edu.cn (X.L.)
[2] Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei 230009, China
* Correspondence: lilin_julia@hfut.edu.cn

Abstract: The layout of facilities in a logistics scenario involves not only the working facilities responsible for processing materials but also the transport lines responsible for transporting materials. The traditional facility layout methods do not take into account the transportation facilities nor calculate the material handling cost by Manhattan distance, thus failing to fulfill the actual requirements of industrial logistics scenarios. In this paper, a facility layout algorithm framework MOSA-FD driven by transport lines is proposed. A multi-objective simulated annealing (MOSA) algorithm is designed for both material handling cost (MHC) and transport facility cost (TFC) objectives. Then, a force-directed (FD) algorithm is applied to correct the unreasonable solutions according to the material transport lines in the logistics workshop, and a better solution is quickly obtained. Finally, by comparing the results with those of other metaheuristic multi-objective algorithms, the acceleration of the force-directed algorithm in this layout problem is demonstrated in experimental instances of different scales, and our method, compared to the MOSA algorithm, can reach optimal ratios of 36% and 80%, respectively, on the multi-objective.

Keywords: logistics facilities layout problem; multi-objective optimization; force-directed algorithm; collaborative computing

check for updates

Citation: Jiang, F.; Li, L.; Tang, Y.; Zhang, H.; Liu, X. A Facility Layout Algorithm for Logistics Scenarios Driven by Transport Lines. *Appl. Sci.* **2023**, *13*, 7215. https://doi.org/10.3390/app13127215

Academic Editor: Arkadiusz Gola

Received: 9 May 2023
Revised: 8 June 2023
Accepted: 14 June 2023
Published: 16 June 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The facility layout problem refers to the design of the optimal layout of working facilities in the workshop to minimize the operating costs and maximize the operational efficiency of the industrial production system for a given material flow and logistics relationship. According to Tompkins' statistics [1], 20–50% of total operating costs and 15–70% of total product manufacturing costs are caused by MHC. In the industrial logistics scenario, the working facilities are responsible for processing, sorting, and storing. The transport lines for transporting materials are directly related to the workshop logistics process, and the MHC involves the actual length of the transport lines and the actual location of the working facilities, so the logistics facility layout problem has obvious special characteristics and is more difficult than the traditional facility layout problem. However, the traditional facility layout algorithms only use logistics information to calculate MHC, which leads to the inefficient performance of the algorithms in the actual large-scale layout. Therefore, researching reasonable facility layout algorithms for logistics scenarios has become an urgent need in the field of industrial logistics.

The logistics facility layout problem falls under the category of traditional facility layout problems by definition, and as the number of facilities in the scenario increases, it becomes increasingly difficult to find the globally optimal solution, which is classified as an NP-hard problem. Early research revealed exact algorithms for solving the problem, mainly including branch and bound algorithms [2,3] and dynamic programming

algorithms [4,5]. Although exact algorithms can obtain the optimal solution by exhaustive search in feasible solutions, they can only solve a small-scale, symmetric instance search space, and the modeling and constraints for the facility layout problem are relatively uncomplicated. The mainstream approach to studying the facility layout problem is to collaboratively compute an approximate optimal solution using metaheuristic algorithms within a reasonable time frame, mainly including the genetic algorithm, the particle swarm algorithm, and the simulated annealing algorithm.

The genetic algorithm possesses excellent global search capability, allowing it to efficiently explore a vast space of potential facility layouts and find reasonable solutions. By employing the co-evolution of multiple populations and implementing a broader search space sampling scheme, population diversity can be effectively maintained [6–8]. Refined crossover and variation operators in genetic algorithms can yield facility layout solutions with lower costs [9,10]. Integrating new solution reconstruction and decomposition strategies can further enhance the layout exploration capability of genetic algorithms [11]. For multi-objective facility layout problems, selecting solutions using genetic algorithms cannot be determined based on a single objective value alone. Each solution may have mutual advantages on different objectives, and the Pareto dominance relationship is employed to evaluate the dominant and non-dominant solution; thus, NSGA II [12–15] is widely used for facility layout optimization.

The particle swarm algorithm optimizes the objective by sharing information among particles, so each one can obtain a better layout solution based on its local and global optimal position. By mapping the discrete search space of the facility layout problem to the continuous search space through coding, high-quality layout solutions can be explored within the feasible region [16–19]. Under the multi-objective function, the algorithm records the Pareto optimal solution set through data archiving, and the algorithm optimizes the layout of each facility by archiving and updating the extreme values of each particle using a roulette wheel selection strategy [20–23].

The simulated annealing algorithm can escape local optima during the search for feasible solutions, and its execution is related to the immediate temperature, with strong search capability and robustness for facility layout problems [24–26]. The layout algorithm requires solving a mathematical model based on mixed integer programming [27] or linear programming [28] methods. Acceptance criteria for multi-objective simulated annealing algorithms are typically designed using two main approaches. The first approach involves calculating acceptance criteria for multiple objective functions in the problem modeling using weighted summation [29,30], and the second approach involves designing the acceptance probability based on the Pareto dominance relation [31,32].

The facility layout problem is similar to the layout problem of the graph structure. Both problems involve placing layout elements within a specified area while satisfying specific constraints; there is a direct correlation between certain layout elements, and the corresponding objectives need to be optimized in the layout problem. The force-directed algorithm is effective for layout problems with a graph structure and is commonly used in the field of visual layout [33–35] as it applies forces to each element, promoting the clustering or movement of these elements.

The facility layout problem can be approached through optimization strategies for layout algorithms and the design of new facility layout models that consider specific assumptions and constraints. Jolai et al. [36] introduced a dynamic facility layout problem that incorporates material entrances and exits within the facility. The positioning of material entrances and exits is determined by the actual location and orientation of the facility within the workshop, which is obtained by transforming the layout according to the facility center coordinates. Dahlbeck [37] proposed the T-row facility layout problem, which involves a set of facilities and two orthogonal rows, optimizing the layout scheme by determining the vertical entrance points between the orthogonal rows and the arranging the facilities within the interior. Besbes et al. [38,39] addressed the facility layout problem in two ways: by considering obstacles and by extending it to 3D space. Obtaining the shortest path between

facilities according to the A* algorithm is more practically valuable than the cost calculation in terms of Euclidean distance or Manhattan distance.

There are three main aspects of the logistics facility layout problem: the working facility layout, the layout constraint processing, and the transport line routing. The existing method involves arranging the working facilities initially, followed by placing the transportation lines. The orientation of the transportation routes in a logistics scenario becomes excessively reliant on the layout of the facilities. Indeed, the cost of operating a workshop is primarily determined by the transportation cost associated with each segment of the transport line. Therefore, in logistics planning, the route of the transport line holds greater significance than the layout of the facility. Moreover, in the actual industrial logistics workshop design, the transport line is firstly considered according to the material transport scheme; then, the location of the facilities within the scheme is determined by the preestablished route, and the strategy of designing facility layout around the transport line is adopted, which is quite different from the existing methods.

To overcome the limitations of existing methods in solving logistics facility layout problems, this paper proposes a multi-objective facility layout model that integrates industrial logistics characteristics. The key contributions of this model are as follows:

(1) A transport line-driven logistics facility layout framework is proposed to optimize multiple objectives based on the concept of Pareto dominance value, using the MOSA algorithm to reduce the operating cost of industrial logistics and increase the diversity of feasible solutions in layout schemes.
(2) A transport line-driven local search mechanism is proposed to correct the unreasonable solutions, using the force-directed algorithm to change the coordinates of the workshop facilities through the action of attraction and repulsion to achieve a fast iteration from unreasonable solutions to reasonable solutions.

The remaining sections of this paper are as follows. Section 2 consists of the description and modeling of the logistics facility layout problem, including objective functions, as well as problem constraints. Section 3 is the algorithmic framework for force-directed driven logistics facility layout, in which each module is described. In Section 4, the algorithm is used on three layout instances of different scales and compared with other metaheuristic optimization algorithms. In Section 5, the logistics facility layout content of this paper is concluded and discussed.

2. Problem Description and Mathematical Model

2.1. Problem Description

The logistics facility layout is a hybrid layout problem, which involves considering both the layout of logistics facilities and the routing of transport lines. However, logistics facilities and transport lines have inherently different characteristics in the actual production process, it is impractical to use a single representation structure to define both types of facilities, and a correlation exists between the two in the layout problem. Based on the functional characteristics and geometric features of different types of logistics facilities in industrial logistics scenarios, this paper posits the following assumptions regarding research content related to facility layout:

(1) Workshop areas and all working facilities are considered as rectangular structures.
(2) The orientation of the working facilities is classified into four types: up and down, left and right, which cannot be placed at an inclined angle.
(3) All working facilities are of a fixed size, and the sum of their total area is less than the area of the workshop.
(4) Each working facility has designated entrance and exit points for material transportation by transport lines.
(5) Transport lines are constrained to extend only in horizontal or vertical directions.
(6) Transport efficiency and unit costs are available for material transport routes and each individual transport line.

The realistic logistics scene mainly includes storage facilities, processing facilities, and transport lines. Within the facilities category, processing facilities can be further divided into three types: manufacturing facilities, turning facilities, and sorting facilities. Figure 1 gives the distribution of entrance and exits of various types of processing facilities. The logistics workshop exists within a two-dimensional plane coordinate system, the position of each facility within the workshop is not predetermined, and it is crucial to ensure that no overlap occurs between any of the facilities in the layout. The size of the workshop and all of the facilities are known, along with the relative locations of the entrance and exits. Transport lines exist between the exits and entrances of different facilities, with line lengths indicating the actual distance of material transport. Figure 2 shows the facility layout of the logistics scenario. The sizes of the workshop are represented by L for length and W for width. The workshop entrance and exit are designated as S_{in} and S_{out}. Facilities i and j are classified as processing facilities, while facility k is categorized as a storage facility. The center coordinates of facility i in the workshop are (x_i, y_i). The transport lines are represented by a set of directed line segments between the facilities, with h_{min} representing the minimum horizontal spacing and v_{min} representing the minimum vertical spacing required between the facilities.

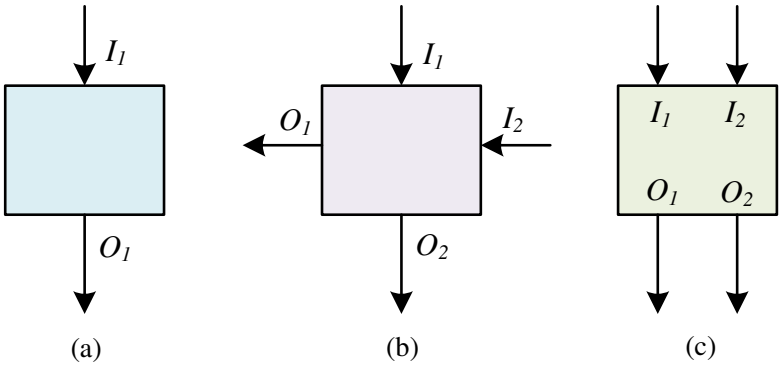


Figure 1. Examples of three types of processing facilities. (**a**) Manufacturing facility; (**b**) turning facility; and (**c**) sorting facility.

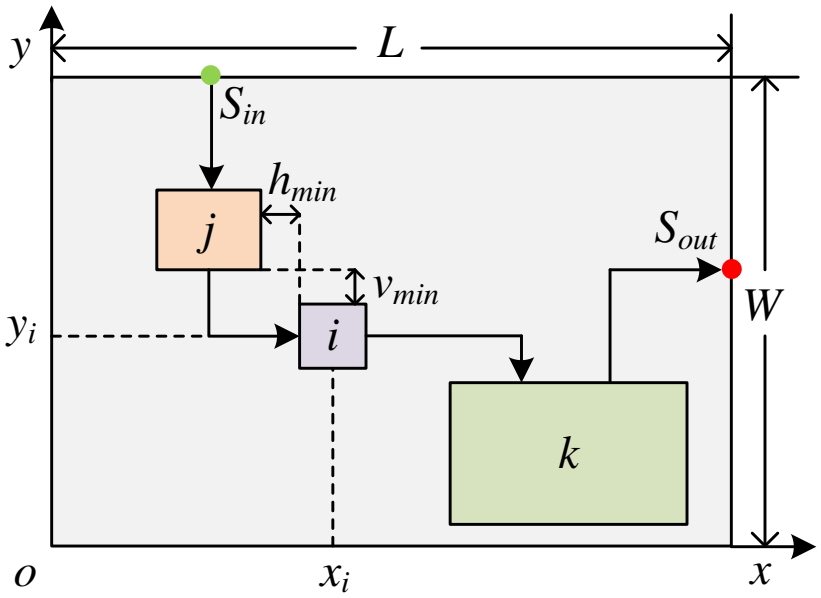


Figure 2. Logistics facility layout diagram.

2.2. Mathematical Model

To assess the impact of transport lines on operating costs in industrial logistics scenarios, two evaluation metrics have been established: material handling cost and transport facility cost. The effectiveness of layout solution X is directly proportional to its ability to minimize the target cost for both evaluation metrics.

The MHC is calculated as shown in Equation (1), which represents the total cost associated with transportation materials along a specified route. Due to the presence of multiple material entrances and exits within logistics facilities, the routing of transport lines linked to each entrance and exit may vary, so Equation (1) is formulated based on the perspective of the transport lines. It is assumed that the facility needs to process, sort, and store multiple materials, and the total number of transport lines required for material transportation is denoted as n. Any given transport line i in the facility has an entrance q and an exit p on either end. The set of exits connected to entrance q is denoted as Q_i, while the set of entrances connected to exit p is represented by P_i. Additionally, the quantity of materials being handled by line i is expressed as g_i. The transport cost per unit distance c_{pq} for material handling depends on the efficiency of each working facility, and the actual distance of line i is denoted as d_{pq}.

$$f_{MHC}(X) = \sum_{i=1}^{n} \sum_{p \in Q_i} \sum_{q \in P_i} g_i c_{pq} d_{pq} \tag{1}$$

The TFC is calculated as shown in Equation (2), which represents the total cost required for the layout of the transport facilities in the workshop. The objective is based on the actual layout of logistics facilities as the transport facilities in logistics workshop are also part of the cost of the company. The total cost of the transport facilities in the logistics workshop comprises two components, which are the total cost of linear conveyors and the total cost of turning conveyors. The cost per unit distance of linear conveyors is denoted as c_s, the cost of each individual turning conveyor is represented by c_t, and the total number of turning conveyors in the transport line is represented as n_t.

$$f_{TFC}(X) = \sum_{i=1}^{n} \sum_{p \in Q_i} \sum_{q \in P_i} c_s d_{pq} + c_t n_t \tag{2}$$

The constraints involved in the logistics facility layout problem are divided into three categories, which are the constraints in the layout of the working facilities, the constraints in the routing of the transport facilities, and the constraints existing between the transport facilities and the working facilities, which are expressed in the following equations.

The working facilities must adhere to two specific constraints, namely, the in-boundary constraint and the non-overlapping constraint. The in-boundary constraint requires that all logistics facilities be situated within the workshop during layout and that the coverage of these facilities does not surpass the area of the workshop. This is depicted mathematically in Equations (3)–(6). The non-overlapping constraint mandates that the logistics facilities cannot overlap with one another and must maintain either horizontal or vertical spacing. The fulfillment of this constraint can be achieved by satisfying one of the conditions specified in Equations (7) and (8). Assuming n denotes the number of working facilities located within the workshop, l_i and w_i are the length and width of the facility i, respectively. The coordinates (x_i, y_i) represent the center coordinates of the rectangular area for each facility.

$$x_i \geq \frac{l_i}{2} + h_{\min}, \forall i \in \{1, 2, \ldots, n\} \tag{3}$$

$$x_i \leq L - \frac{l_i}{2} - h_{\min}, \forall i \in \{1, 2, \ldots, n\} \tag{4}$$

$$y_i \geq \frac{w_i}{2} + v_{\min}, \forall i \in \{1, 2, \ldots, n\} \tag{5}$$

$$y_i \leq W - \frac{w_i}{2} - v_{\min}, \forall i \in \{1, 2, \ldots, n\} \tag{6}$$

$$|x_i - x_j| \geq \frac{l_i}{2} + \frac{l_j}{2} + h_{\min}, \forall i, j \in \{1, 2, \ldots, n\}, i \neq j \tag{7}$$

$$|y_i - y_j| \geq \frac{w_i}{2} + \frac{w_j}{2} + v_{\min}, \forall i, j \in \{1 \ldots n\}, i \neq j \tag{8}$$

The transport facility must fulfill the constraint of either horizontal or vertical extension and the requirement that the transport line between any two turning conveyors is horizontal or vertical, and one of the two satisfies the constraint, as shown in Equations (9)–(12). Assign the coordinates (x_k, y_k) and (x_g, y_g) to the turning conveyors k and g in the transport line, respectively. Designate the coordinates of the linear conveyor v between them as (x_v, y_v).

$$x_v - x_k > 0, y_v - y_k = 0 \tag{9}$$

$$x_v - x_g < 0, y_v - y_g = 0 \tag{10}$$

$$y_v - y_k > 0, x_v - x_k = 0 \tag{11}$$

$$y_v - y_g < 0, x_v - x_g = 0 \tag{12}$$

The transport facility and the working facility must satisfy the spacing constraint between them (Figure 3). To provide sufficient space for staff activities, the first and last turning conveyors in the transport line connecting the two working facilities must maintain a minimum distance from the shipping or incoming ports of the facilities, as required by the spacing constraint and indicated by Equations (13) and (14). Assume that the transport line begins or ends at facility i, and the minimum spacing is d. Then, (x_m, y_m) represent the coordinates of the first or last turning conveyor m in the conveyor line, respectively.

$$|x_m - x_i| \geq \frac{l_i}{2} + d, \forall i \in \{1, 2, \ldots, n\} \tag{13}$$

$$|y_m - y_i| \geq \frac{w_i}{2} + d, \forall i \in \{1, 2, \ldots, n\} \tag{14}$$

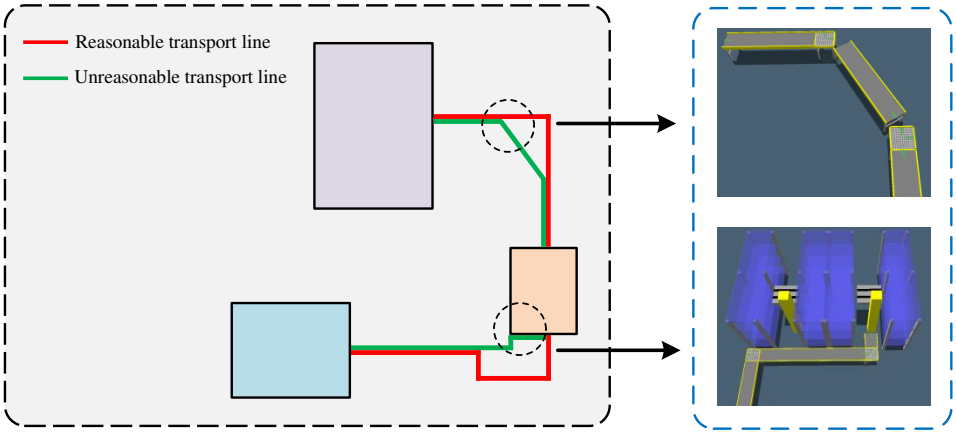


Figure 3. Constraint diagram of transport facilities.

3. Force-Directed Logistics Layout Algorithm

3.1. Algorithm Overview

This paper proposes a transportation line-driven facility layout algorithm framework, MOSA-FD, which uses a multi-objective simulated annealing algorithm combined with a force-directed algorithm local search mechanism for iterative updates. A multi-indicator evaluation path search algorithm is employed to discover the optimal routes between facilities in the logistics scenario. The objective optimization is based on the AMOSA algorithm proposed by Bandyopadhyay et al. [40] to improve its probabilistic formulation and on the local optimization based on the logistics facility layout problem applied in the force-directed algorithm [41,42]. Its main objectives are to achieve an improved Pareto front that optimizes multiple objective values, which combines the two characteristics of the simulated annealing algorithm to jump out of the local optimum and the force-directed algorithm to quickly find the optimum. Figure 4 represents the specific flow of each module of the algorithm in this paper.

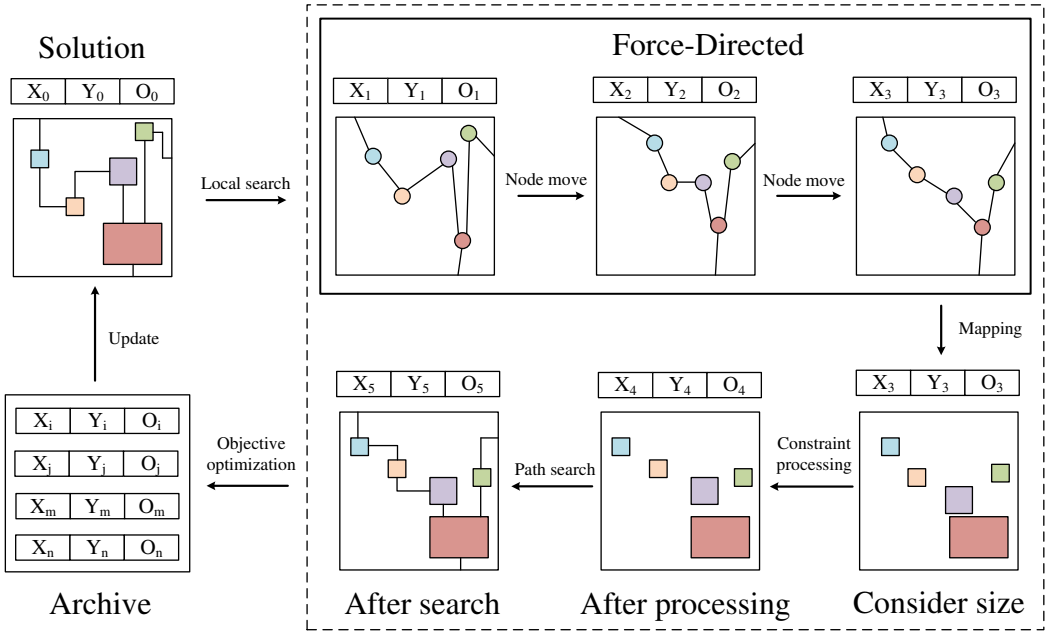


Figure 4. Algorithmic framework for logistics facility layout.

3.2. Decoding and Filtering

The decoding process utilized by the algorithm determines the layout solutions for all working facilities within the scene, and Equation (15) uses matrix encoding for the horizontal coordinates, the vertical coordinates, and the orientation of the n working facilities. In the equation, the current encoding of the algorithm at temperature t is represented by E_t, while the horizontal coordinates of working facility i in the workshop are denoted by x_i, the vertical coordinates of working facility i in the workshop are denoted by y_i, and the orientation information of working facility i in the workshop is denoted by o_i.

$$E_t = [\{x_1, \ldots, x_i, \ldots, x_n\}, \{y_1, \ldots, y_i, \ldots, y_n\}, \{o_1, \ldots, o_i, \ldots, o_n\}] \quad (15)$$

The MOSA-FD algorithm uses different ways of expanding the neighborhood solutions depending on the current temperature. At high temperatures conditions, the MOSA-FD algorithm initially explores coordinates within the workshop area randomly. However, as the temperature decreases, the exploration is limited to the neighborhoods surrounding each facility represented by the current solution. In the early stages of the algorithm, a code-swapping operation is employed to exchange the coding information between any two working facilities in the workshop, thereby facilitating the search for a better solution.

In the mid to later stages of the algorithm, a neighborhood search operation is employed to maintain the current solution representation and search the neighborhood by fine-tuning the coordinates of a working facility within a certain range.

The solution filtering is based on the multi-objective optimization strategy of Pareto dominance relation. The calculation of the acceptance probability involves comparing the new solution with the dominance value obtained from both the archive solution set and the current solution and combining this with the current temperature. In the initial period of high temperature, the new solution is readily accepted based on the dominance relation. However, as the temperature decreases, the acceptance probability gradually diminishes as well. To enhance the variety of feasible solutions in the algorithm, a non-dominated solution is chosen from the archive as the current solution after a designated number of iterations.

3.3. Force-Directed Algorithm

For a realistic industrial logistics scenario, the working facilities and the transport line in the workshop can be regarded as nodes and springs in the spring system, where other facilities and the workshop boundary have a repulsive effect on the working facilities and where the transport line has a attractive effect on the facilities connected at both ends. The solution contains the information of the center coordinates of all working facilities, the position of the nodes in the spring system is established based on the actual center coordinates of the facilities, and the attractive and repulsive forces within the system are employed to pull the facilities into the reasonable area of the workshop in order to optimize the current solution scheme.

The coordinate information of the node is derived from the current solution, and its coordinate position affects the effect of the force in the spring system. With a directional weighted graph $G = (V, E)$, the set of nodes is V, and the set of directional weighted edges is E. The length of the workshop is L, and the width is W. The natural length of the spring is d_0; the repulsive and attractive elasticity coefficients are k_r and k_t, respectively; and the coordinates of node i in the force-directed model are (x_i, y_i). Assuming that nodes a and b belong to the set V, $dist(a, b)$ denotes the Euclidean distance between two nodes in a two-dimensional plane coordinate system, $dist_x(a, b)$ denotes the difference in distance between node a and node b in the x-axis direction, and $dist_y(a, b)$ denotes the difference in distance between node a and node b in the y-axis direction.

Node a in the spring system generates repulsive forces to other nodes. In a realistic logistics scenario, facilities possess size characteristics in relation to nodes; the larger the floor area of a facility in a workshop, the more likely it is to violate the facility non-overlapping constraint, and the greater the repulsive force on other facilities should be. Therefore, when calculating the repulsive force exerted by node b on node a, the length l_b and width w_b of the corresponding facilities of node b are considered. To ensure that the placement of facilities within the workshop adheres to the boundary constraint, it is necessary to consider the repulsive force exerted by the workshop boundary on node a. Equations (16) and (17) represent the horizontal and vertical repulsive forces on node a in the system, where α and β are the weight coefficients of the combined forces of other facilities and the boundary repulsive forces, respectively, which are calculated in a way improved by Coulomb's law.

$$fr_x(a) = \alpha \sum_{b \in V} \frac{dist_x(a,b)}{dist(a,b)} \frac{k_r l_b w_b}{dist(a,b)^2} + \beta\left(\frac{k_r}{{x_a}^2} - \frac{k_r}{(L - x_a)^2}\right) \tag{16}$$

$$fr_y(a) = \alpha \sum_{b \in V} \frac{dist_y(a,b)}{dist(a,b)} \frac{k_r l_b w_b}{dist(a,b)^2} + \beta\left(\frac{k_r}{{y_a}^2} - \frac{k_r}{(W - y_a)^2}\right) \tag{17}$$

The spring system exerts an attractive force on the two end nodes towards the transportation line. In a realistic logistics scenario, different facilities have different efficiencies in handling materials, resulting in different transportation costs per unit distance c_{ab} between

different nodes a and b. Therefore, the spring in the system adjusts the attractive effect on the nodes at both ends according to the cost weight c_{ab}, and the spring with a larger weight factor has a higher attractive force on the nodes, and the material handling cost between facilities is lower. Assuming that nodes a and b are two points located along the same transport line, Equations (18) and (19) represent the horizontal and vertical attraction forces on node a in the system, which are calculated in a way that improves on Hooke's law.

$$ft_x(a) = \sum_{<a,b>\in E} -\frac{dist_x(a,b)}{dist(a,b)} k_t c_{ab} (dist(a,b) - d_0) \tag{18}$$

$$ft_y(a) = \sum_{<a,b>\in E} -\frac{dist_y(a,b)}{dist(a,b)} k_t c_{ab} (dist(a,b) - d_0) \tag{19}$$

In order to prevent the nodes from being subjected to the excessive values of the combined forces, separate thresholds are set for the horizontal and vertical combined forces in the spring system so that the node coordinate positions tend to be stable when iterating. Equations (20) and (21) describe the process of updating the coordinates of node a, including the effect of attractive and repulsive forces, where x_a^* and y_a^* are the horizontal and vertical coordinates of node a after the current iteration. After a fixed number of iterations, each node in the force-directed model will reach a convergence state, and the layout solution driven by the transport line will have a more reasonable facility coordinate position than the original solution. As depicted in Figure 5, the iterative process of the force-directed algorithm is capable of rectifying the impractical positions of facilities in the present solution. This correction leads to a substantial improvement in the efficiency of the local search performed by the algorithm.

$$x_a^* = x_a + fr_x(a) + ft_x(a) \tag{20}$$

$$y_a^* = y_a + fr_y(a) + ft_y(a) \tag{21}$$

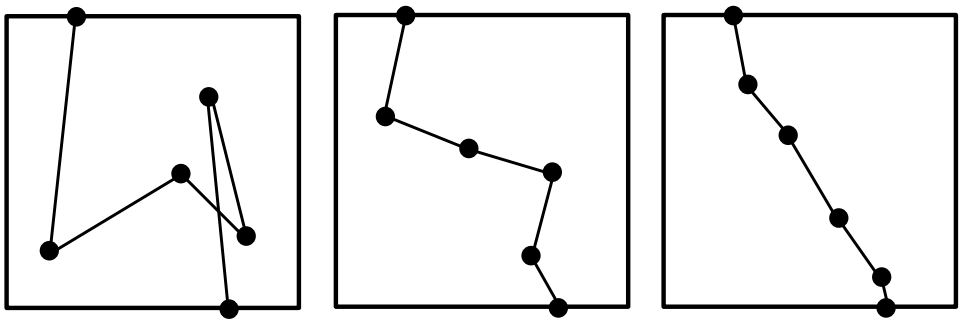


Figure 5. Example of force-directed node adjustment. (**a**) Before adjusting nodes; (**b**) in process of adjusting nodes; and (**c**) after adjusting nodes.

3.4. Constraint Detection and Process

The force-directed algorithm optimizes the layout solution by the center coordinates of the working facilities, considering the effect of attractive and repulsive forces in the spring system to determine the actual location of the facilities. As the logistics scene includes facilities that have size boundaries rather than coordinate nodes within the spring system, there might not fully comply with the prescribed constraints. Therefore, it is necessary to map these facilities onto the actual scene for constraint detection and processing. The constraints of the facilities are shown in Equations (3)–(8). The solution includes both the center coordinates and orientation information of the working facilities, and when combined with the workshop layout parameter information, it enables the determination of area

boundaries for all working facilities, and the boundary constraints and non-overlapping constraints are met by adjusting the center coordinate positions of the working facilities.

In Figure 6, the first step is to add a layer of envelope boxes for all working facilities and detect each facility according to its designated numbers. After applying the force-directed algorithm to determine the relative positions of facilities within the workshop, it is necessary to analyze the relative orientation between any facilities that violate the constraint and reasonably adjust the spacing according to the relative orientation. If a facility exists that is situated in the upper-left quadrant of another center coordinate, the facility should adjust its position in the upper-left region to refine the effect of force-directed local search. Whenever the facilities violating the constraints are adjusted, all of the working facilities constraints are retested, and if there are still facilities violating the constraints, there must be further adjusted and retesting until all of the facilities meet the non-overlapping constraints. If the total area is smaller than that of the workshop area and the boundary constraint is violated, then the envelope rectangle should be moved to the feasible coordinate area of the workshop. If a solution exists with unadjustable constraints, the objective value of this solution will be assigned a higher penalty value to avoid becoming the current solution of the algorithm.

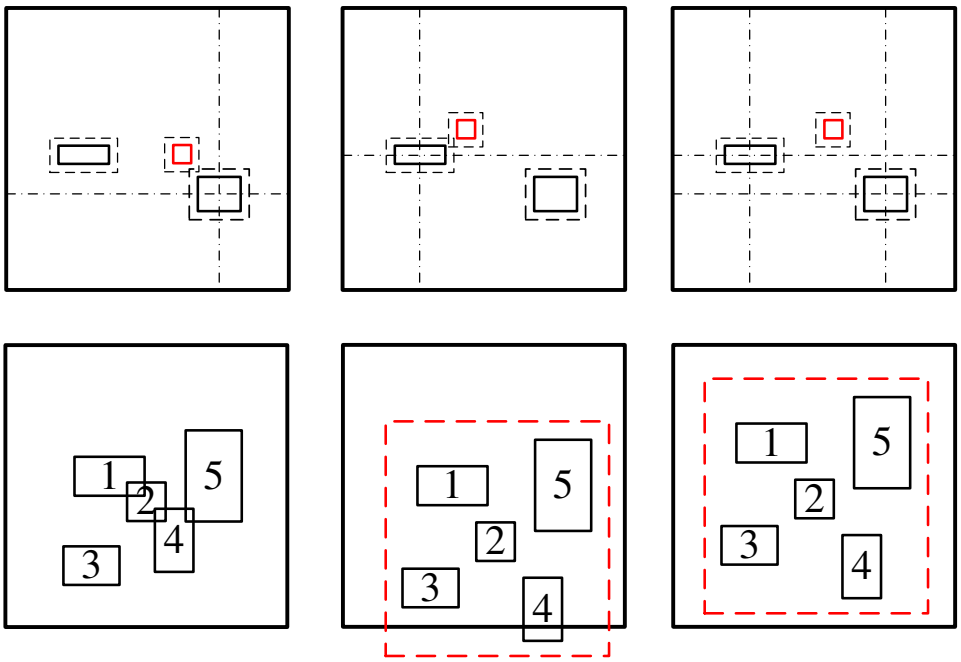


Figure 6. Non-overlapping and boundary constraint adjustment.

3.5. Path Search Algorithm

In this paper, we present a multi-metric evaluation path search algorithm to arrange the transport lines within the workshop, as shown in Figure 7. By utilizing the solution as well as facility and line information, the central coordinates of all facilities, the start and end coordinates of the lines, the boundary point coordinates of the envelope box, and the coordinates of the workshop entrances and exits are obtained. These coordinates are then used to generate sets of horizontal and vertical coordinates, and a collection of path points in the grid is generated. Due to the changing positions of the working facilities during iterations, resulting in the grid that cannot be divided into equal units, the neighbor relationship of the path points is established by sorting the values within the coordinate set. The actual coordinates of the path points are mapped to the grid coordinates to determine the starting point, end point, and obstacle point in the equal division grid. The actual coordinate path of the transport line is obtained after a path search. The steps of the multi-indicator evaluation path search Algorithm 1 (MIEA) are as follows.

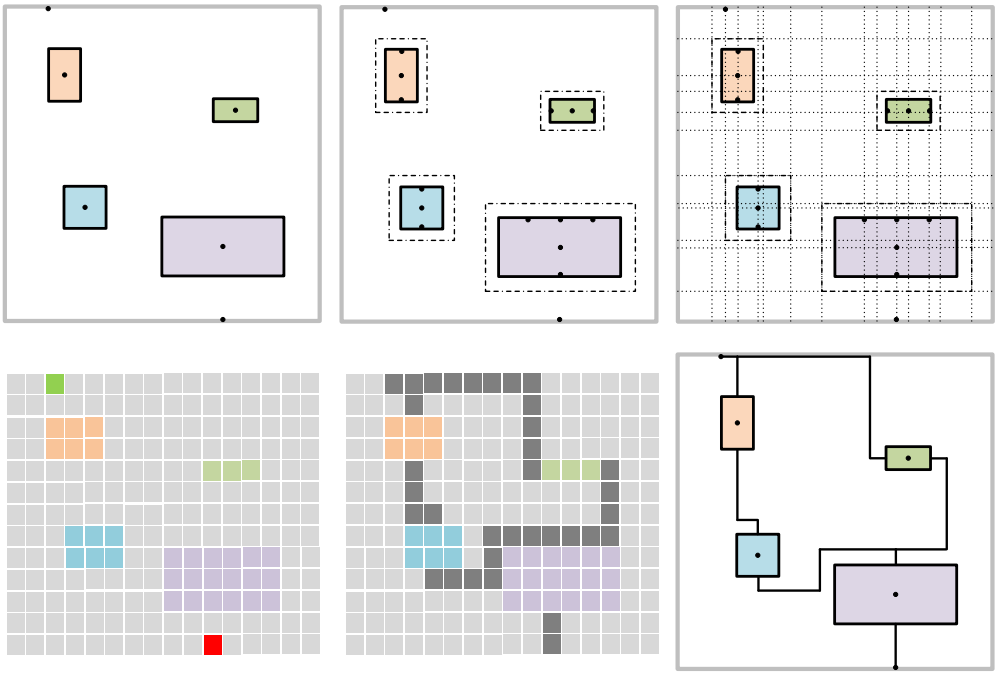


Figure 7. Transport lines search process.

Algorithm 1: MIEA

Input: starting coordinates (x_s, y_s), ending coordinate (x_e, y_e), grid point set M, open list ol, close list cl
Output: path point set P

$ol \leftarrow null$, $cl \leftarrow null$;
remove the node in M inside the facility;
add (x_s, y_s) to ol;
get the node i with the lowest cost in ol;
add (x_i, y_i) to P;
while $(x_i, y_i) \neq (x_e, y_e)$ **do**
    get the node m with the lowest cost in ol;
    add (x_m, y_m) to P;
    delect (x_m, y_m) from ol;
    add (x_m, y_m) to cl;
    get the neighbor nodes i of m;
    $cost(i) \leftarrow G(i) + H(i) + E(i)$;
    add (x_i, y_i) to ol;
end

In realistic logistics scenarios, the cost of a turning conveyor is generally higher than that of a linear conveyor. Therefore, when searching for path nodes along a transportation line, the turning point must be considered, given the cost per unit distance of a linear conveyor is c_s and the cost of a single turn conveyor is c_t, and assuming that the coordinates of the starting point are (x_s, y_s), the coordinates of the ending point are (x_e, y_e), the coordinates of point i are (x_i, y_i), the parent point of a given point i is denoted by p, and the set of turn points is represented as T. In the multi-metric evaluation path search algorithm, the calculation of line cost, denoted as $cost(i)$, involves three primary metric components, where $g(i)$ denotes the actual cost of node i to the starting point, $h(i)$ denotes the cost of a straight line from point i to the ending point, and $e(i)$ denotes the cost of whether point i

is a turner, and the three part cost calculation formulas are shown in Equations (22)–(24), respectively.

$$g(i) = g(p) + (|x_i - x_p| + |y_i - y_p|) \times c_s \tag{22}$$

$$h(i) = (|x_i - x_e| + |y_i - y_e|) \times c_s \tag{23}$$

$$e(i) = \begin{cases} 0 & i \notin T \\ c_t & i \in T \end{cases} \tag{24}$$

Based on the above path search algorithm, each iteration involves traversing all of the transport lines to extract the coordinate set information of all of the lines in the scene. Where the set of transport line path points is represented as $[(x_s, y_s), \ldots, (x_i, y_i), \ldots, (x_e, y_e)]$, the length of each segment of the transport line is determined by calculating the sum of the distances between its front and back coordinates, the MHC objective value is derived by aggregating the cost per unit transport distance of each line, and the TFC objective value is obtained by computing the conveyor cost.

3.6. Objective Optimization Mechanism

In this paper, the two objectives of MHC and TFC are defined to evaluate the quality of feasible solutions for industrial logistics scenarios. After obtaining the target solution metrics through layout and wiring, the MOSA-FD algorithm evaluates the likelihood of accepting the target solution by utilizing both its dominance value and the present temperature. This assessment determines whether the new solution will replace the current one and whether it will be added to the archive. The concept of dominance value is defined in the literature [40], which is represented by Equation (25) and indicates whether solution b is dominated by solution a. In the scenario of dual objectives, the dominance value signifies the proportion of the rectangular region where the diagonal is formed by solutions a and b in relation to the entire objective rectangular area. Figure 8 illustrates that a smaller dominance value implies that the objective values of both solutions are more similar.

$$\Delta dom_{a,b} = \frac{|f_{MHC}(a) - f_{MHC}(b)|}{R_{MHC}} \times \frac{|f_{TFC}(a) - f_{TFC}(b)|}{R_{TFC}} \tag{25}$$

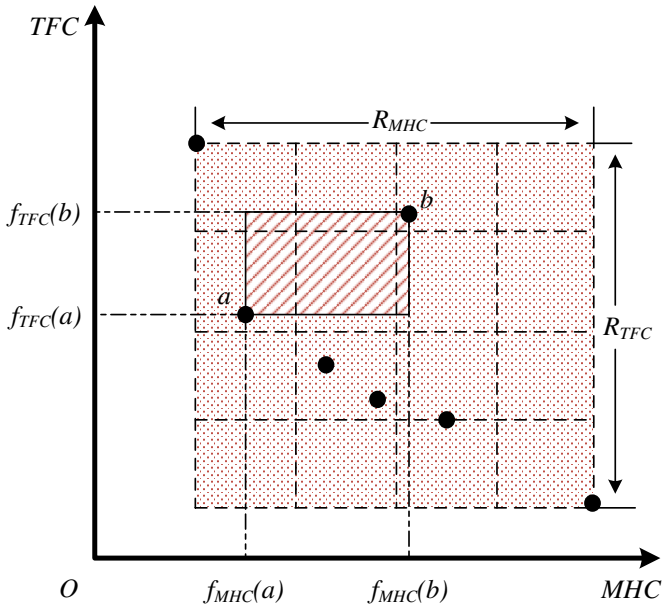


Figure 8. Dominance value and crowding degree.

In Equation (25), the function $f_{MHC}(a)$ represents the MHC objective value of solution a, whereas $f_{TFC}(a)$ denotes the TFC objective value of solution a, and the range of values for the MHC and TFC objective functions in the new solution and archive are represented by R_{MHC} and R_{TFC}, respectively. To compare the current solution with the target solution, Pareto dominance relationship is employed. The three possible cases are: the current solution dominates the target solution, the target solution dominates the current solution, or the two solutions do not dominate each other. The likelihood of accepting the target solution can be determined by analyzing the dominant value and the current temperature, which are discussed in detail below:

(1) If the current solution dominates the target solution, it means that the current solution outperforms the target solution on all objective values. If the target solution is dominated by k solutions in the archive, then the acceptance probability can be calculated using Equations (26) and (27) to replace the current solution, but the target solution will not be added to the archive.

$$\Delta dom_{avg} = \frac{\sum_{i=1}^{k} \Delta dom_{i,new} + \Delta dom_{current,new}}{k+1} \tag{26}$$

$$prob = e^{-(\Delta dom_{avg}/T)} \tag{27}$$

(2) If the target solution and the current solution are non-dominated, it means that both have advantages and disadvantages on the objective values. If the target solution is not dominated in the archive, it will be set as the current solution and added to the archive, while all feasible solutions dominated by the target solution will be removed from the archive. If the target solution is dominated by k solutions in the archive, the acceptance probability can be calculated using Equations (27) and (28) to replace the current solution, but the target solution will not be added to the archive.

$$\Delta dom_{avg} = \frac{\sum_{i=1}^{k} \Delta dom_{i,new}}{k} \tag{28}$$

(3) If the target solution dominates the current solution, it means that the target solution is superior to the current solution on all objective values. If the target solution is not dominated in the archive, set it as the current solution and add it to the archive data while removing all feasible solutions dominated by the target solution in the archive. If the target solution is dominated by k solutions in the archive, use the non-dominated solution corresponding to Δdom_{min} to calculate the acceptance probability for replacing the current solution; otherwise, directly replace the current solution with the target solution. The formula for calculating the probability is detailed in Equation (29), where Δdom_{min} represents the smallest domination value distance from the target solution in the archive.

$$prob = \frac{1}{1 + \exp(-\Delta dom_{\min})} \tag{29}$$

When the quantity of non-dominated solutions in archive exceeds the threshold, the crowding degree is calculated by counting the number of non-dominated solutions in the regional solution space grid. The higher the crowding degree, the higher the probability of removing the non-dominated solutions, and the more uniform the distribution of non-dominated solutions, as shown in Figure 8. Through the utilization of the MOSA-FD algorithm, the non-dominated solutions in archive will gradually tend to be better, and the Pareto front will be closer to both axes.

4. Experimental Results and Analysis

4.1. Experimental Design

In order to verify the effect of force-directed on the layout of logistics scenes and determine the optimal combination of MOSA and force-directed, this paper applies force-directed to MOPSO, NSGAII, and mixed integer programming (MIP) respectively, forming MOPSO-FD, NSGA-FD and MIP-FD algorithms. The four multi-objective algorithms are tested separately by comparison experiments, in which each algorithm is run independently for 10 times, and the parameters of the algorithms are kept consistent for each run. Therefore, the number of iterations for each algorithm is determined based on the scale, with 100, 80, and 50 iterations used for different sizes while ensuring that the running time of each algorithm is kept in check. The known information of the workshops, facilities, and lines of each instance is shown in Tables 1–7. The transport lines are represented as shown in Table 5. The line 1 indicates that the material passes through entrance 1 of the workshop, entrance 1 and exit 1 of facility 1, entrance 1 and exit 1 of facility 4, entrance 1 and exit 1 of facility 5, and entrance 1 of facility 6. The unit cost represents the unit transportation cost per path segment in the line.

Table 1. Logistics workshop data.

Instance	Workshop Length	Workshop Width	Entrance Coordinates	Exit Coordinates
1	35	35	(3, 0)	(27, 35)
2	50	50	(0, 8), (0, 42)	(50, 25)
3	60	60	(0, 10), (15, 0),(30, 0), (45, 0), (60, 10)	(0, 50), (15, 60), (30, 60), (45, 60), (60, 50)

Table 2. Instance 1 facilities data.

Number	Type	Length	Width	Entrance Coordinates	Exit Coordinates	RGB Values
1	manufacture	2	1	(0, −0.5)	(0, 0.5)	(138, 201, 227)
2	manufacture	2	4	(0, −2)	(0, 2)	(212, 106, 109)
3	manufacture	2.5	2	(0, −1)	(0, 1)	(79, 106, 80)
4	manufacture	3	1.8	(0, −0.9)	(0, 0.9)	(62, 25, 1)
5	sort	6	4	(−1.5, −2), (1.5, −2)	(−1.5, 2), (1.5, 2)	(75, 77, 185)
6	storage	10	6	(−3, 3), (0, 3)	(3, 3)	(26, 239, 161)

Table 3. Instance 2 facilities data.

Number	Type	Length	Width	Entrance Coordinates	Exit Coordinates	RGB Values
1	manufacture	2.5	2.5	(−1.25, 0)	(1.25, 0)	(138, 201, 227)
2	manufacture	4	2	(−2, 0)	(2, 0)	(212, 106, 109)
3	manufacture	2.5	2	(−1.25, 0)	(1.25, 0)	(79, 106, 80)
4	manufacture	4	2	(−2, 0)	(2, 0)	(62, 25, 1)
5	sort	6	4	(−3, −1), (−3, 1)	(3, 0)	(75, 77, 185)
6	manufacture	4	4	(0, −2)	(0, 2)	(26, 239, 161)
7	manufacture	6	3	(−3, 0)	(3, 0)	(236, 110, 68)
8	manufacture	4	6	(0, −3)	(0, 3)	(161, 8, 130)
9	manufacture	3	4	(0, 2)	(0, −2)	(204, 212, 219)
10	manufacture	2	4	(0, −2)	(0, 2)	(31, 194, 133)
11	storage	10	6	(−3, 3)	(3, 3)	(125, 154, 15)
12	storage	8	8	(−2, 4), (0, 4)	(2, 4)	(217, 86, 114)

Table 4. Instance 3 facilities data.

Number	Type	Length	Width	Entrance Coordinates	Exit Coordinates	RGB Values
1	manufacture	2	2	(0, −1)	(0, 1)	(138, 201, 227)
2	manufacture	1	3	(0, −1.5)	(0, 1.5)	(212, 106, 109)
3	manufacture	2	2	(0, −1)	(0, 1)	(79, 106, 80)
4	manufacture	1	3	(0, −1.5)	(0, 1.5)	(62, 25, 1)
5	manufacture	2	2	(0, −1)	(0, 1)	(75, 77, 185)
6	manufacture	2	2	(0, −1)	(0, 1)	(26, 239, 161)
7	manufacture	1	3	(0, −1.5)	(0, 1.5)	(236, 110, 68)
8	manufacture	2	2	(0, −1)	(0, 1)	(161, 8, 130)
9	manufacture	1	3	(0, −1.5)	(0, 1.5)	(204, 212, 219)
10	manufacture	2	2	(0, −1)	(0, 1)	(31, 194, 133)
11	manufacture	1	3	(0, −1.5)	(0, 1.5)	(125, 154, 15)
12	manufacture	2	2	(0, −1)	(0, 1)	(217, 86, 114)
13	manufacture	1	3	(0, −1.5)	(0, 1.5)	(171, 197, 229)
14	manufacture	2	2	(0, −1)	(0, 1)	(177, 132, 229)
15	manufacture	1	3	(0, −1.5)	(0, 1.5)	(62, 80, 194)
16	manufacture	2	2	(0, −1)	(0, 1)	(127, 66, 63)
17	manufacture	1	3	(0, −1.5)	(0, 1.5)	(147, 85, 124)
18	manufacture	2	2	(0, −1)	(0, 1)	(28, 43, 57)
19	manufacture	1	3	(0, −1.5)	(0, 1.5)	(133, 50, 26)
20	manufacture	2	2	(0, −1)	(0, 1)	(78, 119, 40)
21	manufacture	1	3	(0, −1.5)	(0, 1.5)	(197, 70, 234)
22	manufacture	2	2	(0, −1)	(0, 1)	(218, 63, 10)
23	storage	6	6	(−1, 3)	(1, 3)	(67, 85, 139)
24	storage	6	6	(−1, 3)	(1, 3)	(129, 82, 163)
25	storage	6	6	(−1, 3)	(1, 3)	(150, 41, 123)

Table 5. Instance 1 line data.

Number	Transport Line	Unit Cost
1	workshop entrance 1—facility 1 (1/1)—facility 4 (1/1)—facility 5 (1/1)—facility 6 entrance 1	3, 1, 2, 1
2	workshop entrance 1—facility 2 (1/1) - facility 3 (1/1)—facility 5 (2/2)—facility 6 entrance 2	2, 2, 1, 3
3	workshop entrance 1—facility 1 (1/1)—facility 3 (1/1)—facility 5 (1/1)—facility 6 entrance 1	3, 1, 2, 1
4	facility 6 exit 1—workshop exit 1	2

Table 6. Instance 2 line data.

Number	Transport Line	Unit Cost
1	workshop entrance 1—facility 1 (1/1)—facility 2 (1/1)—facility 3 (1/1)—facility 5 (1/1)—facility 11 entrance 1	2, 3, 5, 2, 4
2	workshop entrance 1 - facility 1 (1/1)—facility 2 (1/1)—facility 4 (1/1)—facility 5 (2/1)—facility 11 entrance 1	2, 3, 4, 2, 4

Table 6. *Cont.*

Number	Transport Line	Unit Cost
3	facility 4 (1/1)—facility 6 (1/1)—facility 12 entrance 2	3, 4
4	workshop entrance 2—facility 7 (1/1)—facility 8 (1/1)—facility 9 (1/1)—facility 12 entrance 1	1, 4, 2, 2
5	facility 11 entrance 1—facility 10 (1/1)—workshop exit 1	2, 5
6	facility 12 exit 1—workshop exit 1	2

Table 7. Instance 3 line data.

Number	Transport Line	Unit Cost
1	workshop entrance 1—facility 1 (1/1)—facility 2 (1/1)—facility 3 (1/1)—facility 4 (1/1)—facility 5 (1/1)—workshop exit 1	3, 2, 2, 1, 2, 3
2	workshop entrance 2—facility 6 (1/1)—facility 7 (1/1)—facility 8 (1/1)—facility 9 (1/1)—facility 23 entrance 1	3, 2, 2, 1, 2
3	workshop entrance 3—facility 10 (1/1)—facility 11 (1/1)—facility 12 (1/1)—facility 13 (1/1)—facility 24 entrance 1	3, 2, 2, 1, 2
4	workshop entrance 4—facility 14 (1/1)—facility 15 (1/1)—facility 16 (1/1)—facility 17 (1/1)—facility 25 entrance 1	3, 2, 2, 1, 2
5	workshop entrance 5—facility 18 (1/1)—facility 19 (1/1)—facility 20 (1/1)—facility 21 (1/1)—facility 22 (1/1)—workshop exit 5	3, 2, 2, 1, 2, 3
6	facility 23 exit 1—workshop exit 2	2
7	facility 24 exit 1—workshop exit 3	2
8	facility 25 exit 1—workshop exit 4	2

There four algorithms proposed in this paper is validated for its performance through layout instances of different scales. The layout instance of a small-scale consists of six facilities, one workshop entrance, and one workshop exit. The layout instance of a medium-scale consists of 12 facilities, 2 workshop entrances, and 1 workshop exit. The layout instance of large-scale consists of 25 facilities, 5 workshop entrances, and 5 workshop exits. Experiments run on Intel(R) Xeon(R) CPU E5-2609 v4 1.70 GHz CPU, 16 GB RAM. Each algorithm is implemented using Python 3.6.

The horizontal distance h_{min} and vertical distance v_{min} of the workshop are 2; the distance d between the facility and the transport line is 2; and the unit cost of the linear conveyor c_s and the unit cost of the turning conveyor c_t are 100 and 2000, respectively. The MOSA-FD algorithm uses 80 iterations for neighbor finding, with the initial temperature set to 10,000 and a cooling rate of 0.9. In contrast, the parameters of the MOPSO-FD algorithm include a particle population size of 100, C_1 and C_2 values of 0.5, and a W value of 0.5. The NSGAII-FD algorithm is configured with a population size of 100, a crossover rate of 0.8, and a mutation rate of 0.2. Among them, the force-oriented algorithm is iterated a total of 16 times, with a facility repulsion weight α of 0.6 and a boundary repulsion weight β of 0.4. Additionally, the attractive force coefficient k_t is set to 1, while the repulsive force

coefficient k_r is set to 2. The result values of each algorithm for different instances are shown in Tables 8–11.

Table 8. Results of NSGAII-FD and NSGAII under each instance. Bold values are better.

Instance	Algorithm	MHC			TFC			TIME		
		Worst	Best	Average	Worst	Best	Average	Worst	Best	Average
1	NSGAII-FD	**270.12**	225.33	**250.24**	**126.32**	**98.69**	**105.07**	**521.14**	**399.82**	**458.95**
	NSGAII	437.56	**224.98**	336.86	171.39	102.19	131.00	607.91	492.87	544.35
2	NSGAII-FD	**569.41**	**473.41**	**534.96**	**428.80**	**319.78**	**384.44**	**1484.58**	**1332.43**	**1411.37**
	NSGAII	689.65	560.73	606.74	712.34	518.06	605.06	2662.71	2125.97	2415.38
3	NSGAII-FD	**1163.72**	**1072.23**	**1105.13**	**1018.10**	**826.96**	**936.44**	**4299.41**	**3841.69**	**4097.30**
	NSGAII	1656.63	1524.48	1582.29	4379.90	3643.47	3875.08	6733.82	5994.14	6505.01

Table 9. Results of MOPSO-FD and MOPSO under each instance. Bold values are better.

Instance	Algorithm	MHC			TFC			TIME		
		Worst	Best	Average	Worst	Best	Average	Worst	Best	Average
1	MOPSO-FD	**280.68**	**219.35**	**250.66**	**110.55**	**99.68**	**103.90**	**484.05**	**471.07**	**479.05**
	MOPSO	304.45	219.43	265.24	124.41	101.74	111.06	553.26	502.57	515.03
2	MOPSO-FD	**676.25**	**499.21**	**602.11**	**742.15**	**423.69**	**520.97**	**1735.24**	**1574.53**	**1646.30**
	MOPSO	1026.73	830.83	904.50	808.47	618.61	735.02	2587.26	1981.39	2317.53
3	MOPSO-FD	**1181.48**	**1081.24**	**1128.65**	**1361.22**	**1172.46**	**1260.78**	**4210.14**	**3632.34**	**3989.21**
	MOPSO	2128.14	1923.34	2031.30	7006.57	6592.06	6790.41	6625.03	5700.33	6419.60

Table 10. Results of MOSA-FD and MOSA under each instance. Bold values are better.

Instance	Algorithm	MHC			TFC			TIME		
		Worst	Best	Average	Worst	Best	Average	Worst	Best	Average
1	MOSA-FD	**286.64**	**219.13**	**243.19**	**117.08**	**97.95**	**103.42**	**427.23**	**403.29**	**420.01**
	MOSA	361.66	248.16	302.90	151.31	110.61	128.58	537.33	501.70	516.80
2	MOSA-FD	**541.23**	**425.32**	**476.73**	**400.77**	**281.32**	**321.60**	**1453.64**	**1340.59**	**1398.02**
	MOSA	739.04	668.93	712.82	699.37	595.84	636.49	2468.91	2142.55	2347.53
3	MOSA-FD	**1134.21**	**1062.81**	**1105.24**	**1198.36**	**911.99**	**1026.85**	**3379.75**	**2819.38**	**3074.41**
	MOSA	1762.41	1661.23	1718.37	5731.62	4940.75	5403.53	6383.47	5502.90	5830.48

Table 11. Results of MIP-FD and MIP under each instance. Bold values are better.

Instance	Algorithm	MHC			TFC			TIME		
		Worst	Best	Average	Worst	Best	Average	Worst	Best	Average
1	MIP-FD	329.78	**236.59**	251.67	133.75	**97.07**	124.20	81.79	51.58	63.12
	MIP		469.42			206.15			**32.43**	

Table 11. *Cont.*

Instance	Algorithm	MHC			TFC			TIME		
		Worst	**Best**	**Average**	**Worst**	**Best**	**Average**	**Worst**	**Best**	**Average**
2	MIP-FD	596.29	**537.53**	586.75	447.19	**395.23**	432.63	152.95	126.84	130.36
	MIP		802.11			790.14			**80.7**	
3	MIP-FD	1294.67	**1043.34**	1124.57	1159.41	**960.45**	1073.7	383.22	332.09	354.43
	MIP		1927.62			6648.15			**273.57**	

In terms of time performance, it is evident that the transport line-driven layout algorithm has a significantly lower runtime; since the attractive and repulsive forces are implemented on the facilities by the force-directed algorithm, it becomes simpler for the algorithm to obtain solutions that adhere to boundary constraints and non-overlapping facility constraints and leads to a decrease in the number of constraint processing required The runtime of the MIP-FD and MIP methods is significantly lower than that of the metaheuristic algorithm in this paper. This can be attributed to the utilization of the Gurobi optimizer, which computes the mathematical model and obtains optimal results quickly. Additionally, there is no extensive iterative optimization process required after the wiring stage. In terms of objective performance, the cost of the layout demonstrates a correlation with both the location of the facility and the orientation of the transport lines, based on the provided objective Equations (1) and (2). This correlation remains consistent across the same experimental data for a specific instance because the workshop area factors are held constant. The transport line-driven layout algorithm exhibits a lower average objective value because the force-directed algorithm makes efficient use of the available transport line data to search for the optimal location for each facility within a reasonable range. Therefore, this enables the algorithm to achieve a lower number of transport line lengths and turning points, resulting in an improved objective value.

Figure 9 shows a comparison of the Pareto optimal frontier plots for all of the algorithms in each instance, where the optimal frontier based on the transport line drive algorithm is obviously closer to the coordinate system on both sides. Figure 10 shows the Pareto optimal frontier plots. These plots were obtained through experiments by modifying the main parameters of each metaheuristic algorithm under the same layout conditions. For the MOSA-FD algorithm, the cooling rate is changed to 0.95; for the MOPSO-FD algorithm, the values of C1 and C2 are changed to 0.8, and the value of W is changed to 0.2; and for the NSGAII-FD algorithm, the crossover rate is changed to 0.9, and the mutation rate is changed to 0.1. Through the comparison of the frontier plots, similar to Figure 9, the transport line-driven layout algorithms (MOSA-FD, NSGAII-FD, and MOPSO-FD) exhibit lower MHC and TFC objective costs compared to the non-transport line-driven layout algorithms (MOSA, NSGAII, and MOPSO). This significant cost difference can be attributed to the traction effect of the force-directed algorithm. However, due to modifications in some metaheuristic parameters, the MOSA-FD, NSGAII-FD, and MOPSO-FD algorithms demonstrate larger cost fluctuations and overall higher layout costs compared to the results in Figure 9.

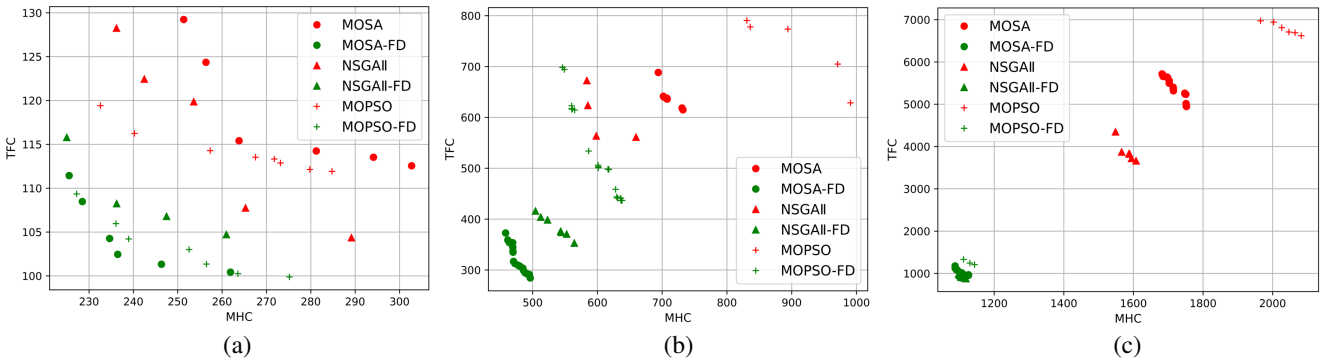


Figure 9. The Pareto optimal frontier plots. (**a**) Instance 1; (**b**) instance 2; and (**c**) instance 3.

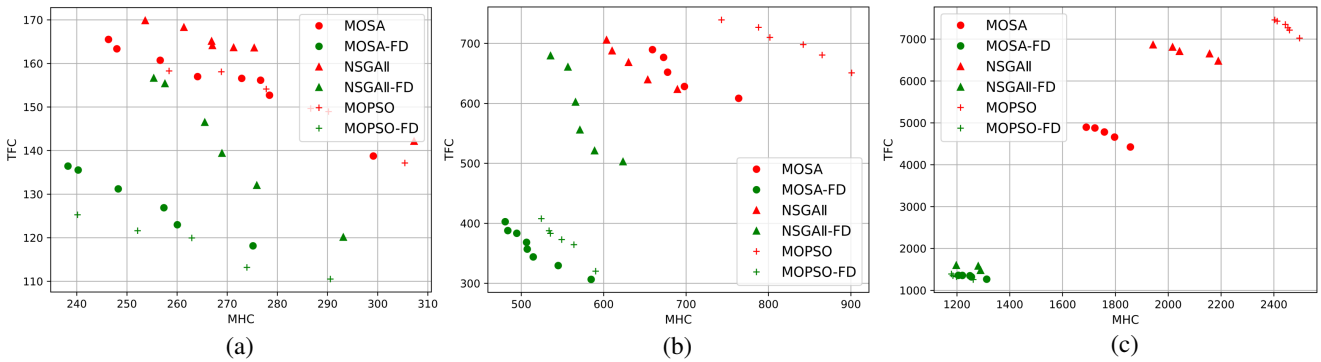


Figure 10. The Pareto optimal frontier plots for contrast parameters. (**a**) Instance 1; (**b**) instance 2; and (**c**) instance 3.

Figure 11a–c shows the histograms of the results of the three transport line-driven based layout algorithms for different scale instances. For the time performance, MOSA-FD exhibits a better average running time than the other algorithms since NSGAII-FD and MOPSO-FD are population-based optimized algorithms, which entail conducting force-directed local search and constraint detection and processing operations for each solution in the population. However, the MOSA-FD algorithm is based on the current solution only, and the target solution is obtained through the neighbor finding operation and force-directed algorithm processing. For the objective performance, the MOSA-FD algorithm demonstrates superior performance in terms of average MHC and TFC objective values, which further validates the effectiveness of the multi-objective optimization mechanism proposed in this paper. Figure 11d–f shows the histograms of the optimization ratios of three transport line-driven layout algorithms at different scale instances. Taking the results of layout instance 3 as an example, based on the average metrics data of MHC, TFC, and TIME, it can be found that the metrics optimization ratios of NSGAII-FD compared to NSGAII are 30%, 76%, and 37%, respectively; the metrics optimization ratios of MOPSO-FD compared to MOPSO The proportions are 44%, 81%, and 37%, respectively; and the metrics optimization proportions of MOSA-FD compared with MOSA are 36%, 80%, and 47%, respectively, indicating that the force-directed algorithm can significantly improve the performance of logistics facility layout algorithms.

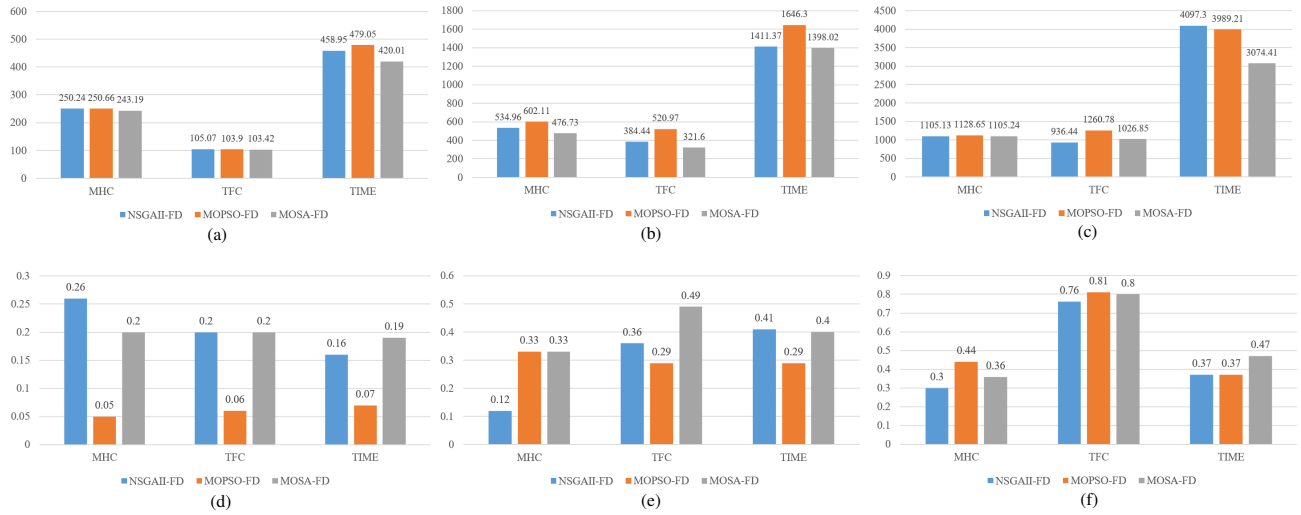


Figure 11. Histogram of data for each algorithm at each instance. (**a**) Instance 1 of each algorithm metric; (**b**) instance 2 of each algorithm metric; (**c**) instance 3 of each algorithm metric; (**d**) instance 1 optimization ratio of transport line-driven algorithm; (**e**) instance 2 optimization ratio of transport line-driven algorithm; and (**f**) instance 3 optimization ratio of transport line-driven algorithm.

4.2. Layout Analysis

Figures 12–14 represent the layout result plots of each algorithm for different scale instances, respectively. The solution space of the facility layout problem increases with the number of facilities. Since the non-transport line-driven layout algorithms (NSGA II, MOPSO, MOSA, and MIP) fail to effectively use the information of known transport lines, resulting in a scenario where the facility locations can only be generated randomly, it is impossible to find a reasonable area for the layout of facilities in the workshop with a large solution space. Moreover, the layout outcomes often contain excessive turning points and the workshop space is divided into multiple small areas due to the presence of transport lines. This can result in limited mobility for the staff within the facility. In contrast, the layout of facilities and transport lines based on the transport line-driven layout algorithms (NSGA II-FD, MOPSO-FD, MOSA-FD, and MIP-FD) is more reasonable, the activity space in the workshop is more open, and this characteristic is more obvious as the scale of the layout instance increases. The transport line drive algorithm exhibits significantly shorter transport lines than the non-transport line-driven method, and there is no need for bypassing. As a result, this approach performs better in terms of cost metrics. By comparing the layout solutions of MOSA-FD algorithm and MIP method, it is obvious to find that MOSA-FD algorithm belongs to free-space layout, which has fewer constraints on the coordinate position of the facilities, and the logistics relationship of the facility layout will be more closely driven by force direction. While the MIP method belongs to the multi-row facility layout, the same row of facilities will be in the same coordinate area, and the lines of the layout solution are greatly influenced by the location of the facilities, which makes it difficult to discover the logistic relationship of the layout solution.

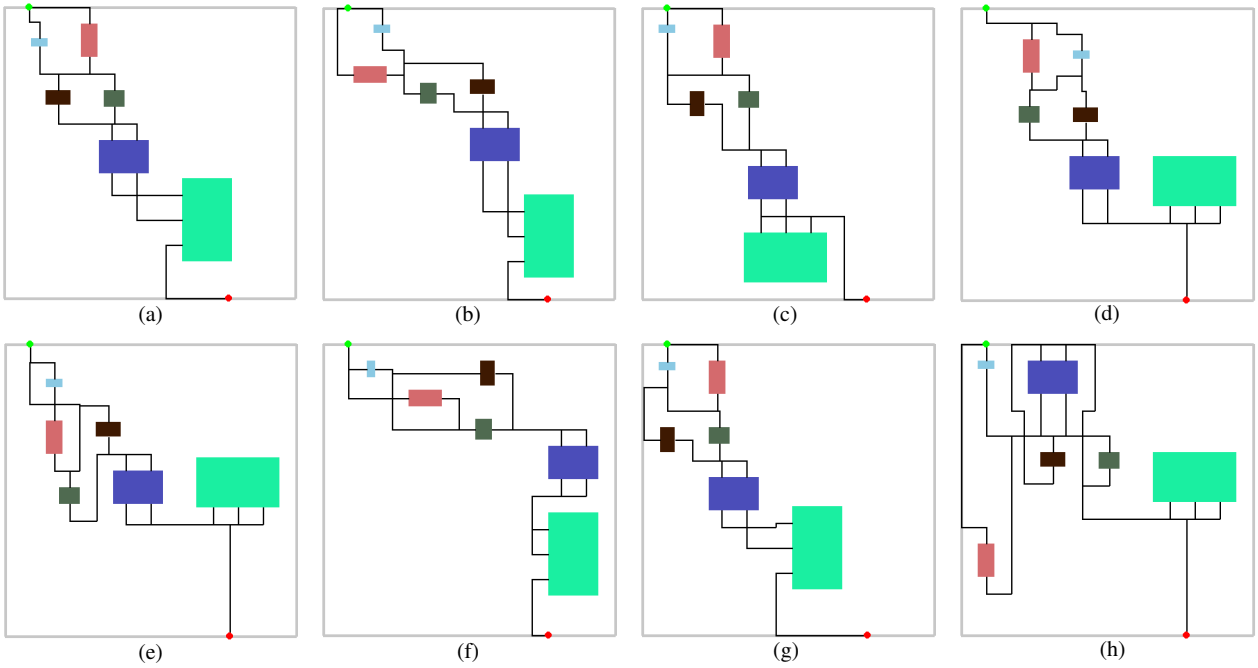


Figure 12. Layout plots of each algorithm at instance 1. (**a**) NSGA II-FD; (**b**) MOPSO-FD; (**c**) MOSA-FD; (**d**) MIP-FD; (**e**) NSGA II; (**f**) MOPSO; (**g**) MOSA; and (**h**) MIP.

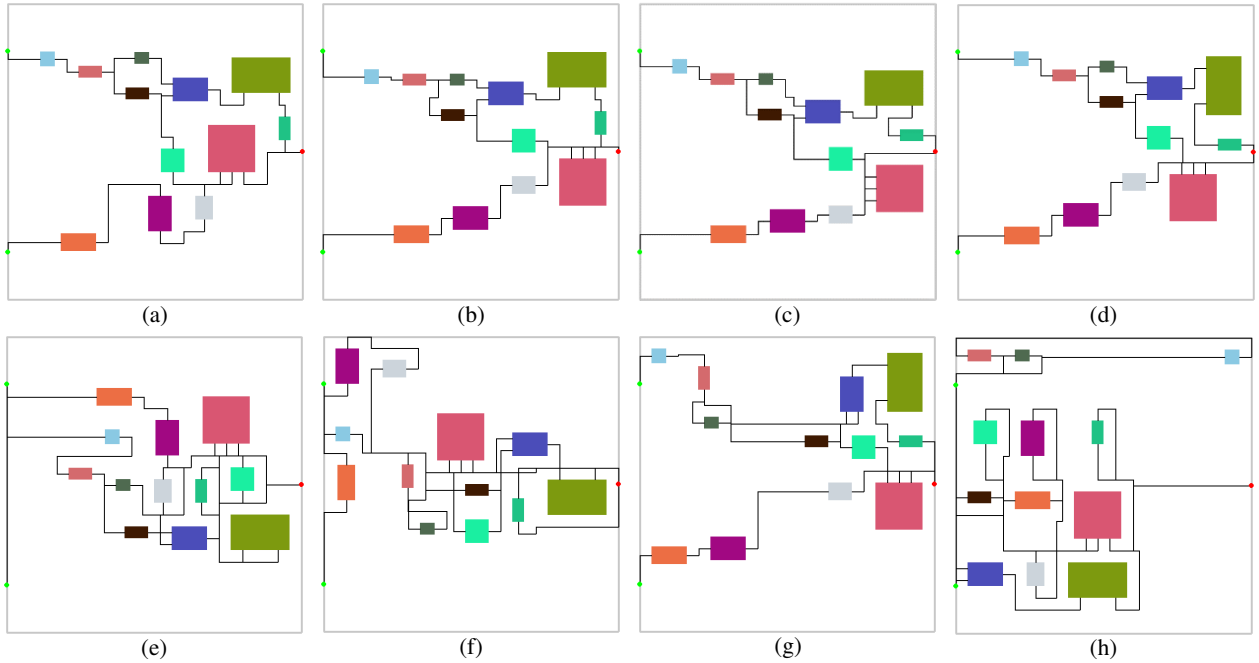


Figure 13. Layout plots of each algorithm at instance 2. (**a**) NSGA II-FD; (**b**) MOPSO-FD. (**c**) MOSA-FD; (**d**) MIP-FD; (**e**) NSGA II; (**f**) MOPSO; (**g**) MOSA; and (**h**) MIP.

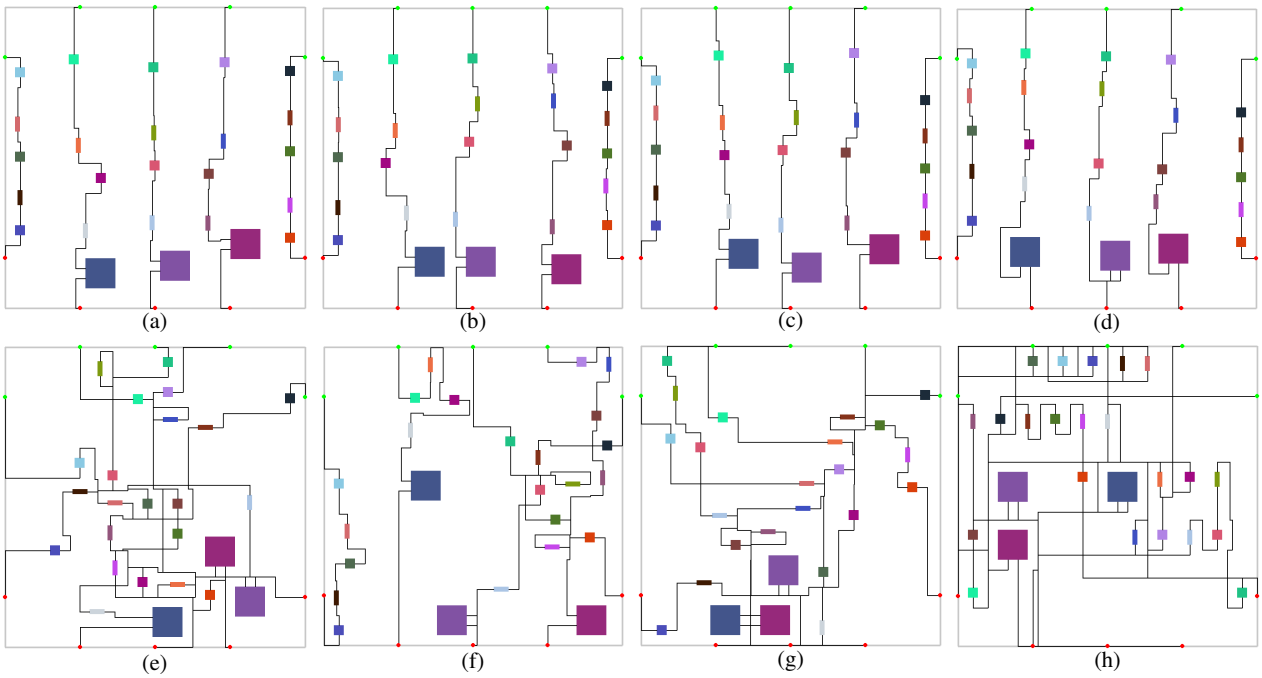


Figure 14. Layout plots of each algorithm at instance 2. (**a**) NSGA II-FD; (**b**) MOPSO-FD; (**c**) MOSA-FD; (**d**) MIP-FD; (**e**) NSGA II; (**f**) MOPSO; (**g**) MOSA; and (**h**) MIP.

Figure 15 shows the simulation diagrams corresponding to the layout scheme of logistics facilities using MOSA-FD algorithm on three instances respectively, according to which it can be seen that a strong facility correlation exists between the transport line-driven layout diagrams. The facilities with a logistics relationship are closely connected, the material transportation distance is shorter, and the target cost of the solution can be optimized. Therefore, the MOSA-FD model-driven logistics facility layout algorithm can be highly valuable for practical application in workshop layouts.

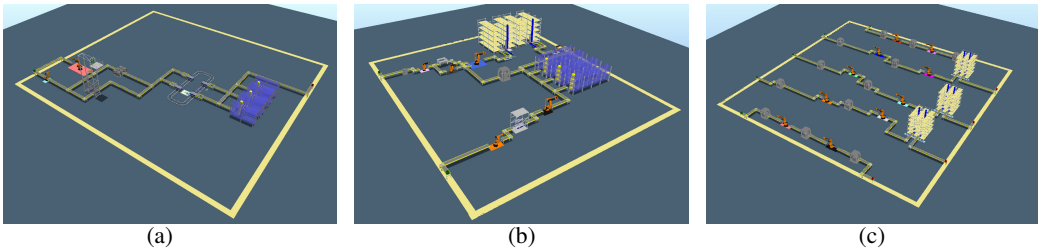


Figure 15. MOSA-FD simulation results for each instance. (**a**) Instance 1; (**b**) instance 2; and (**c**) instance 3.

5. Conclusions

Based on the existence of transport lines in logistics workshops, this paper proposes a transport line-driven logistics facility layout algorithm. The innovative approach is to incorporate the force-directed into the layout of facilities and transport lines, separating the layout of transport lines from the arrangement of facilities and adjusting the positions of facilities through elastic forces. Compared to non-transport line-driven algorithms, this approach can effectively reduce operational costs and improve algorithmic time efficiency.

We designed an algorithmic framework for transport line-driven logistics facility layout. The framework utilizes the MOSA algorithm as an optimization method, which optimizes the objective function values for a feasible solution. In addition, the framework utilizes the local search mechanism of the force-directed algorithm to corrects the layout positions of facilities by attractive and repulsive effects, and obtains a better layout solution. The experiments show that the MOSA-FD algorithm has time performance and cost ad-

vantages over the population-based optimization algorithms (MOPSO-FD, NSGA2-FD). In some instances, the method is able to reduce the objective cost by 36% and 80%, respectively, and the runtime by 47% compared to the MOSA algorithm.

By treating each facility and transport line in the workshop as a node and spring in a spring system, respectively, and pulling the facilities into the reasonable area of the workshop according to the force in the system to adjust the current solution of the facility layout, the local search of each logistics facility in the reasonable layout area of the workshop is realized. According to the experimental metrics and simulation results, it can be concluded that the solution adjusted by the force-directed algorithm avoids multiple constraint adjustments and provides more contiguous activity space for the workshop staff in the logistics workshop.

The MOSA-FD algorithm is designed to work with fixed transport line information, but in reality, material handling lines may change. Therefore, the next step in research would be to consider introducing fuzzy intelligence technology [43–45] to help optimize the layout of facilities.

Author Contributions: Conceptualization, F.J.; methodology, F.J. and L.L.; validation, F.J. and H.Z.; formal analysis, F.J., L.L. and X.L.; investigation, F.J.; resources, L.L., X.L. and Y.T.; data curation, F.J.; writing—original draft preparation, F.J.; writing—review and editing, F.J., L.L. and Y.T.; visualization, F.J.; supervision, L.L. and Y.T.; and funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China grant number 62277014 and the Key Research and Development Project in Anhui Province of China grant number 2022f04020006.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tompkins, J.A.; White, J.A.; Bozer, Y.A.; Tanchoco, J.M.A. *Facilities Planning*; John Wiley & Sons: Hoboken, NJ, USA, 2010.
2. Kim, J.; Kim, Y.D. A branch and bound algorithm for locating input and output points of departments on the block layout. *J. Oper. Res. Soc.* **1999**, *50*, 517–525. [CrossRef]
3. Xie, W.; Sahinidis, N.V. A branch-and-bound algorithm for the continuous facility layout problem. *Comput. Chem. Eng.* **2008**, *32*, 1016–1028. [CrossRef]
4. Rosenblatt, M.J. The dynamics of plant layout. *Manag. Sci.* **1986**, *32*, 76–86. [CrossRef]
5. Oncan, T.; Altinel, I.K. Exact solution procedures for the balanced unidirectional cyclic layout problem. *Eur. J. Oper. Res.* **2008**, *189*, 609–623. [CrossRef]
6. Palomo-Romero, J.M.; Salas-Morera, L.; García-Hernández, L. An island model genetic algorithm for unequal area facility layout problems. *Expert Syst. Appl.* **2017**, *68*, 151–162. [CrossRef]
7. Sun, X.; Lai, L.F.; Chou, P.; Chen, L.R.; Wu, C.C. On GPU implementation of the island model genetic algorithm for solving the unequal area facility layout problem. *Appl. Sci.* **2018**, *8*, 1604. [CrossRef]
8. Dunker, T.; Radons, G.; Westkämper, E. A coevolutionary algorithm for a facility layout problem. *Int. J. Prod. Res.* **2003**, *41*, 3479–3500. [CrossRef]
9. El-Baz, M.A. A genetic algorithm for facility layout problems of different manufacturing environments. *Comput. Ind. Eng.* **2004**, *47*, 233–246. [CrossRef]
10. Sadrzadeh, A. A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. *Comput. Ind. Eng.* **2012**, *62*, 1055–1064. [CrossRef]
11. Paes, F.G.; Pessoa, A.A.; Vidal, T. A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. *Eur. J. Oper. Res.* **2017**, *256*, 742–756. [CrossRef]
12. Pourhassan, M.R.; Raissi, S. An integrated simulation-based optimization technique for multi-objective dynamic facility layout problem. *J. Ind. Inf. Integr.* **2017**, *8*, 49–58. [CrossRef]
13. Huo, J.; Liu, J.; Gao, H. An nsga-ii algorithm with adaptive local search for a new double-row model solution to a multi-floor hospital facility layout problem. *Appl. Sci.* **2021**, *11*, 1758. [CrossRef]
14. Esmikhani, S.; Kazemipoor, H.; Sobhani, F.M.; Molana, S.M.H. Solving fuzzy robust facility layout problem equipped with cranes using MPS algorithm and modified NSGA-II. *Expert Syst. Appl.* **2022**, *210*, 118402. [CrossRef]

15. Aiello, G.; La Scalia, G.; Enea, M. A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. *Expert Syst. Appl.* **2012**, *39*, 10352–10358. [CrossRef]
16. Derakhshan Asl, A.; Wong, K.Y. Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization. *J. Intell. Manuf.* **2017**, *28*, 1317–1336. [CrossRef]
17. Kulturel-Konak, S.; Konak, A. A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem. *Eng. Optim.* **2011**, *43*, 1263–1287. [CrossRef]
18. Samarghandi, H.; Taabayan, P.; Jahantigh, F.F. A particle swarm optimization for the single row facility layout problem. *Comput. Ind. Eng.* **2010**, *58*, 529–534. [CrossRef]
19. Hu, B.; Yang, B. A particle swarm optimization algorithm for multi-row facility layout problem in semiconductor fabrication. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3201–3210. [CrossRef]
20. Liu, J.; Zhang, H.; He, K.; Jiang, S. Multi-objective particle swarm optimization algorithm based on objective space division for the unequal-area facility layout problem. *Expert Syst. Appl.* **2018**, *102*, 179–192. [CrossRef]
21. Zhang, Z.; Wu, L.; Wu, Z.; Zhang, W.; Jia, S.; Peng, T. Energy-Saving Oriented Manufacturing Workshop Facility Layout: A Solution Approach Using Multi-Objective Particle Swarm Optimization. *Sustainability* **2022**, *14*, 2788. [CrossRef]
22. Tang, H.; Ren, S.; Jiang, W.; Chen, Q. Optimization of Multi-Objective Unequal Area Facility Layout. *IEEE Access* **2022**, *10*, 38870–38884. [CrossRef]
23. Guan, C.; Zhang, Z.; Liu, S.; Gong, J. Multi-objective particle swarm optimization for multi-workshop facility layout problem. *J. Manuf. Syst.* **2019**, *53*, 32–48. [CrossRef]
24. Hunagund, I.; Pillai, V.; Kempaiah, U. A simulated annealing algorithm for unequal area dynamic facility layout problems with flexible bay structure. *Int. J. Ind. Eng. Comput.* **2018**, *9*, 307–330. [CrossRef]
25. Bozer, Y.A.; Wang, C.T. A graph-pair representation and MIP-model-based heuristic for the unequal-area facility layout problem. *Eur. J. Oper. Res.* **2012**, *218*, 382–391. [CrossRef]
26. Grobelny, J.; Michalski, R. A novel version of simulated annealing based on linguistic patterns for solving facility layout problems. *Knowl.-Based Syst.* **2017**, *124*, 55–69. [CrossRef]
27. Kulturel-Konak, S.; Konak, A. A large-scale hybrid simulated annealing algorithm for cyclic facility layout problems. *Eng. Optim.* **2015**, *47*, 963–978. [CrossRef]
28. Xiao, Y.; Zheng, Y.; Zhang, L.; Kuo, Y. A combined zone-LP and simulated annealing algorithm for unequal-area facility layout problem. *Adv. Prod. Eng. Manag.* **2016**, *11*, 259–270. [CrossRef]
29. Matai, R. Solving multi objective facility layout problem by modified simulated annealing. *Appl. Math. Comput.* **2015**, *261*, 302–311. [CrossRef]
30. Şahin, R. A simulated annealing algorithm for solving the bi-objective facility layout problem. *Expert Syst. Appl.* **2011**, *38*, 4460–4465. [CrossRef]
31. Şahin, R.; Türkbey, O. A simulated annealing algorithm to find approximate Pareto optimal solutions for the multi-objective facility layout problem. *Int. J. Adv. Manuf. Technol.* **2009**, *41*, 1003–1018. [CrossRef]
32. Pourvaziri, H.; Pierreval, H. Dynamic facility layout problem based on open queuing network theory. *Eur. J. Oper. Res.* **2017**, *259*, 538–553. [CrossRef]
33. Aris, K.A.W.M.; Ramasamy, C.; Aris, T.N.M.; Zolkepli, M. Dynamic Force-directed Graph with Weighted Nodes for Scholar Network Visualization. *Int. J. Adv. Comput. Sci. Appl.* **2022**, *13*, 756–770. [CrossRef]
34. Jenny, B.; Stephen, D.M.; Muehlenhaus, I.; Marston, B.E.; Sharma, R.; Zhang, E.; Jenny, H. Force-directed layout of origin-destination flow maps. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1521–1540. [CrossRef]
35. Streit, M.; Gratzl, S.; Gillhofer, M.; Mayr, A.; Mitterecker, A.; Hochreiter, S. Furby: Fuzzy force-directed bicluster visualization. *BMC Bioinform.* **2014**, *15*, 1–13. [CrossRef]
36. Jolai, F.; Tavakkoli-Moghaddam, R.; Taghipour, M. A multi-objective particle swarm optimisation algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations. *Int. J. Prod. Res.* **2012**, *50*, 4279–4293. [CrossRef]
37. Dahlbeck, M. A mixed-integer linear programming approach for the T-row and the multi-bay facility layout problem. *Eur. J. Oper. Res.* **2021**, *295*, 443–462. [CrossRef]
38. Besbes, M.; Zolghadri, M.; Costa Affonso, R.; Masmoudi, F.; Haddar, M. A methodology for solving facility layout problem considering barriers: genetic algorithm coupled with A* search. *J. Intell. Manuf.* **2020**, *31*, 615–640. [CrossRef]
39. Besbes, M.; Zolghadri, M.; Costa Affonso, R.; Masmoudi, F.; Haddar, M. 3D facility layout problem. *J. Intell. Manuf.* **2021**, *32*, 1065–1090. [CrossRef]
40. Bandyopadhyay, S.; Saha, S.; Maulik, U.; Deb, K. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Trans. Evol. Comput.* **2008**, *12*, 269–283. [CrossRef]
41. Kamada, T.; Kawai, S. An algorithm for drawing general undirected graphs. *Inf. Process. Lett.* **1989**, *31*, 7–15. [CrossRef]
42. Fruchterman, T.M.; Reingold, E.M. Graph drawing by force-directed placement. *Softw. Pract. Exp.* **1991**, *21*, 1129–1164. [CrossRef]
43. Tang, Y.; Zhang, L.; Bao, G.; Ren, F.; Pedrycz, W. Symmetric implicational algorithm derived from intuitionistic fuzzy entropy. *Iran. J. Fuzzy Syst.* **2022**, *19*, 27–44.

44. Tang, Y.; Pan, Z.; Pedrycz, W.; Ren, F.; Song, X. Viewpoint-based kernel fuzzy clustering with weight information granules. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 342–356. [CrossRef]
45. Tang, Y.; Huang, J.; Pedrycz, W.; Li, B.; Ren, F. A Fuzzy Clustering Validity Index Induced by Triple Center Relation. *IEEE Trans. Cybern.* **2023**, 1–13. DIO:10.1109/TCYB.2023.3263215. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.