

Research Article

A Fast Algorithm to Generate Feasible Solution of Production Facilities Layout Based on Plane Segmentation

Shi-wang Hou,¹ Zhibin Li,² and Hui Wang²

¹School of Business, Huaihua University, Huaihua, Hunan 400082, China

²School of Mechanical and Power Engineering, North University of China, Taiyuan, Shanxi 030051, China

Correspondence should be addressed to Shi-wang Hou; houshiwan@163.com

Received 5 November 2015; Revised 21 February 2016; Accepted 6 March 2016

Academic Editor: Xavier Delorme

Copyright © 2016 Shi-wang Hou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For facility layout problem with continuous block and unequal area, it is key to generate feasible solution of facility layout with arbitrary space form in order to find the optimal arrangement scheme under a given goal. According to the given slicing position and slicing mode, the plane for arrangement was divided into many block areas by use of plane segmentation, which was consistent with the facilities in number. The precise coordinates of the lower-left corner and the top-right corner of each facility were calculated in light of its area, width, and length. The corresponding algorithm was designed in the form of pseudocode. The procedure proposed can provide a feasible facility layout solution. The running results of facilities layout instance containing 14 facilities show that the scheme can output facilities plane layout scheme quickly and provide decision support for the facilities planning.

1. Introduction

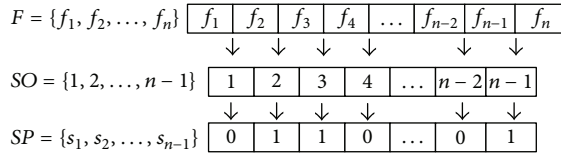
Layout design is an important content of facilities planning and design, mainly to deal with the relative position and area of various kinds of functional facilities (i.e., production or service units) as well as their relevant auxiliary facilities in order to make work flow (customers or materials) and information flow unblocked [1]. The concrete content includes the following two aspects: determination of relative position of all facilities and area of each facility. The former refers to the location relationship between different facilities and the latter refers to the size of each facility. Facilities layout solves the space allocation problem in production and operation and its purpose is to make effective combination of all facilities and obtain the maximum economic benefit. If the facility layout is unreasonable, that is, there are many invalid handling or moving or complex working sites and workshop channels, long distance and high cost handling, long time waiting, and poor production balance, the overall operating efficiency of the production system can be affected greatly [2].

Suppose all facilities' area is equal and only consider the location of the relationship between facilities; facilities layout problem (FLP) is ascribed to quadratic assignment

problem (QAP) [3, 4]. By doing so, FLP is simplified to allocate n facilities to n discrete points without considering the location coordinates for multirow and multicolumn FLP. But this supposition is hard to meet in real production system and the application of this method is limited. A lot of FLPs are for multiple continuous blocks with unequal area as pointed out in [2]. As a special case, single row FLP considers one-dimensional linear array arrangement of rectangular facilities. In this case, all facilities are arranged in the same line in spite of the consideration of different area of every facility, so the generation of feasible solution and the implementation of algorithm are relatively easy [5–11].

Double row layout is widely adapted in flexible manufacturing system [12–17] to facilitate automatic guided vehicles conveying material. The complexity on feasible solution generation was also greatly reduced by limiting the facilities to double line.

Taking facility layout objective function as criterion, the optimal facility layout approach can be found by selecting the arrangement form freely rather than by limiting its form as single, double, or multiline artificially [18]. But that will greatly expand the feasible solution space of facilities layout. So the key to search the optimal layout approach combined

FIGURE 1: Relationship between vectors F , SO , and SP .

with heuristic intelligent algorithm is to generate the feasible solution quickly.

This paper is organized as follows. Section 2 introduces the principle of plane segmentation method for facility layout. Section 3 describes the layout step of plane segmentation method and the pseudocode of the realizing algorithm. Section 4 describes an application case containing 14 facilities, verification test in Matlab environment, and some of the evaluation results of this method. Finally, Section 5 concludes with a discussion of the paper.

2. The Principle of Plane Segmentation Method for Facility Layout

Suppose vector $F = \{f_1, f_2, \dots, f_n\}$ is a collection of facilities waiting for layout; vector $SO = \{1, 2, \dots, n-1\}$ is a collection of splitting positions for each plane splitting operation; vector $SP = \{s_1, s_2, \dots, s_{n-1}\}$ is a collection of splitting modes, where n denotes the facility number and $s_i = 0$ or 1 ($i = 1, 2, \dots, n-1$).

In each plane splitting operation, the relationship between vectors F , SO , and SP is shown in Figure 1.

In actual splitting operation, the element in vectors F , SO , and SP can be any permutation within their respective values that an element may take. For a vector F with given element order, there is a splitting position between every two adjacent elements. From left to right, the splitting position randomly ranges from 1 to $(n-1)$, respectively, and forms the splitting position vector SO . For a vector SO with given element order, there is a splitting mode for every element. From left to right, the splitting mode takes the value 1 or 0, respectively, and forms the splitting position vector SP . According to the splitting position value of each element in SP , vector F is partitioned into two parts, left and right. The value in vector SP corresponding to the splitting position value determines the plane splitting mode, and 0 and 1 denote splitting vertically and horizontally, respectively.

Taking a FLP with 6 facilities as an example, the above approach can be described in Figures 2 and 3. Here, $F = \{f_1, f_2, \dots, f_6\}$, $SO = \{3, 2, 4, 1, 5\}$, and $SP = \{1, 0, 1, 1, 0\}$.

3. The Realization of Plane Partition Method of Facility Layout

3.1. Determination of Left Set $setL$ and Right Set $setR$ for Each Partitioning Operation

Step 1. Find the maximum of existing splitting position SO_{max} .

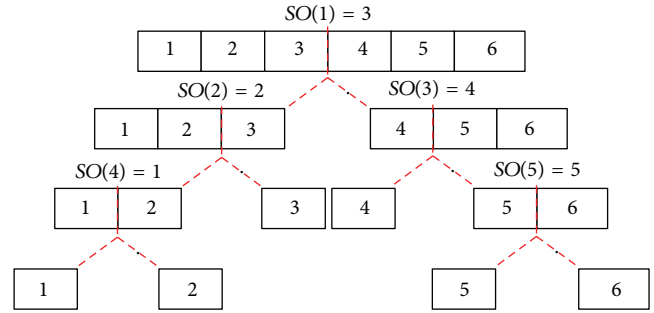


FIGURE 2: Set partitioning diagram.

Step 2. Find the minimum of existing splitting position SO_{min} .

Step 3. In accordance with the relationship between the current splitting position SO_{cur} , SO_{max} , and SO_{min} , new left set $setL$ and right set $setR$ are determined as follows:

- (i) If $SO_{cur} < SO_{min}$, in this case, the current splitting position SO_{cur} lies in existing $setL$ where SO_{min} was found before. This operation partitions this $setL$ into two parts, that is, new $setL$ and $setR$.
- (ii) If $SO_{cur} > SO_{max}$, the current splitting position SO_{cur} lies in the existing $setR$ where SO_{max} was found before. This operation partitions this $setR$ into two parts, that is, new $setL$ and $setR$.
- (iii) If $SO(i) < SO_{cur} < SO(j)$, here, $SO(i)$ denotes the maximum of existing splitting position less than SO_{cur} and $SO(j)$ denotes the minimum of existing splitting position greater than SO_{cur} . The completion order of splitting operation corresponding to the splitting positions $SO(i)$ and $SO(j)$ affects the object of this partition. If $SO(i)$ splitting was carried out later than $SO(j)$, that is, $i > j$, the right set resulting from $SO(i)$ splitting was going to be partitioned. Similarly, if $SO(i)$ splitting was carried out earlier than $SO(j)$, that is, $i < j$, the left set resulting from $SO(j)$ splitting was going to be partitioned.

This paper defines two structures to record the information of splitting position and its results, sets $setL$ and $setR$; that is, $setL = \text{struct}(\text{"splitting position"}, \{\}, \text{"left set"}, \{\})$ and $setR = \text{struct}(\text{"splitting position"}, \{\}, \text{"right set"}, \{\})$.

The process to determine the sets $setL$ and $setR$ of every splitting operation is shown in Algorithm 1.

The partition task in Figure 2 was completed using the above approach and the results are shown in Table 1. After repeating the operation 5 times, the facility set was partitioned into 5 sets, each of which contained only one element. At the same time, the plane was also partitioned into 5 blocks, each of which was for arranging a facility.

3.2. Coordinates Determination of the First Splitting Line.

Before the first splitting operation, the layout area was empty and no facility was arranged. The coordinates of the lower-left corner and upper-right corner of the layout plane were

```

Procedure (determination of setL and setR) (N is the cardinality of facility set F, cso is position set that have been split)
(1) setL(1) ← struct("splitting position", {0}, "left set", {}); % Initialize the first left set with empty set
(2) setR(1) ← struct("splitting position", {0}, "right set", {F}); % Initialize the first right set with set F
(3) for i ← 1 To N
(4)   [maxcso, ix_max] ← max(cso);
(5)   [mincso, ix_min] ← min(cso);
(6)   temp1 ← max(index(SO(i) > cso));
(7)   temp2 ← min(index(SO(i) < cso));
(8)   if SO(i) > cso(ix_max)
(9)     setL(i) ← struct("splitting position", {SO(i)}, "left set", {setR(ix_max).right set(1: SO(i)-maxcso)});
(10)    setR(i) ← struct("splitting position", {SO(i)}, "right set", {setR(ix_max).right set(SO(i-maxcso): end)});
(11)  elseif SO(i) < cso(ix_min)
(12)    setL(i) ← struct("splitting position", {SO(i)}, "left set", {setL(ix_min).left set(1: SO(i))});
(13)    setR(i) ← struct("splitting position", {SO(i)}, "right set", {setL(ix_min).left set((SO(i) + 1): end)});
(14)  elseif cso(temp1) < s(i) < cso(temp2)
(15)    if temp1 > temp2
(16)      setL(i) ← struct("splitting position", {SO(i)}, "left set", {setR(temp1).right set(1: SO(i)-cso(temp1))});
(17)      setR(i) ← struct("splitting position", {SO(i)}, "left set", {setR(temp1).right set((s(i)-cso(temp1) + 1): end)});
(18)    elseif temp1 < temp2
(19)      ix ← length(setL(temp2).left set);
(20)      setL(i) ← struct("splitting position", {SO(i)}, "left set", {setL(temp2).left set(1: ix-cso(temp2) + SO(i))});
(21)      setR(i) ← struct("splitting position", {SO(i)}, "right set", {setL(temp2).left set(ix-cso(temp2) + SO(i) + 1): end)});
(22)    end if
(23)  end if
(24) end for
    
```

ALGORITHM 1: Process to determine the left and right set for given splitting position.

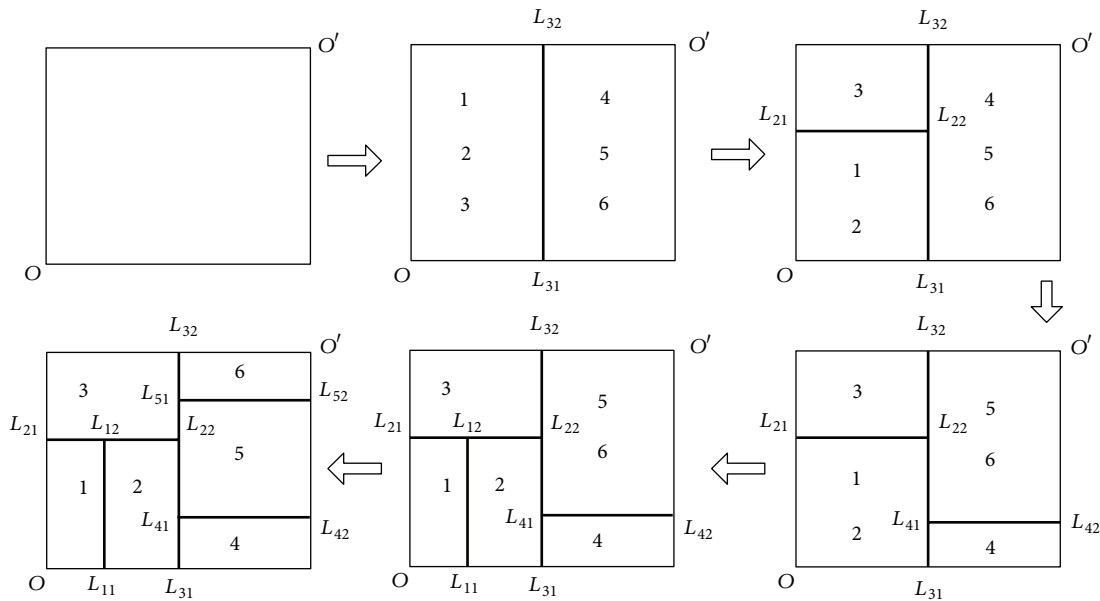


FIGURE 3: Plane segmenting process including 6 facilities.

denoted by $O(0, 0)$ and $O'(\sum h_i, \sum v_i)$, where h_i and v_i were the horizontal width and the vertical height of facility i , respectively. The coordinates of the upper-right corner were the extreme value when all the facilities were arranged side by side horizontally or vertically. The plane was firstly split according to the value of $SP(1)$. The splitting line was denoted by $L_{i1}L_{i2}$, where the first subscript $i = 1, 2, \dots, n - 1$ shows

the splitting position and the second subscript taking value of 1 or 2 means the start point or end point of splitting line. The line direction is from left to right for horizontal line and from bottom to top for vertical line.

(i) $SP(1) = 0$. In this case, the first line is a horizontal one partitioning the layout area into two parts, upper and

```

Procedure (Drawing the first splitting line) ( $F, SP, setL, setR, LW, LH$ )
(1)  $[tf, index] \leftarrow setL(1).elements$  and their index in  $F$ ;
(2) if  $SP(1) = 0$ 
(3)    $hlin\_start(1) \leftarrow 0$ ; %  $x$ -coordinate of horizontal line start point
(4)    $hlin\_end(1) \leftarrow sum(LW)$ ; %  $x$ -coordinate of horizontal line end point
(5)    $hlin\_y(1) \leftarrow sum(LH(F(index)))$ ; %  $y$ -coordinate of horizontal line
(6) elseif  $SP(1) = 1$ 
(7)    $vlin\_start(1) \leftarrow 0$ ;
(8)    $vlin\_end(1) \leftarrow sum(LH)$ ;
(9)    $vlin\_x(1) \leftarrow sum(LW(F(index)))$ ;
(10) end if
(11) Connect point  $(hlin\_start(1), hlin\_y(1))$  and  $(hlin\_end(1), hlin\_y(1))$ 

```

ALGORITHM 2: Coordinate determination of the first line.

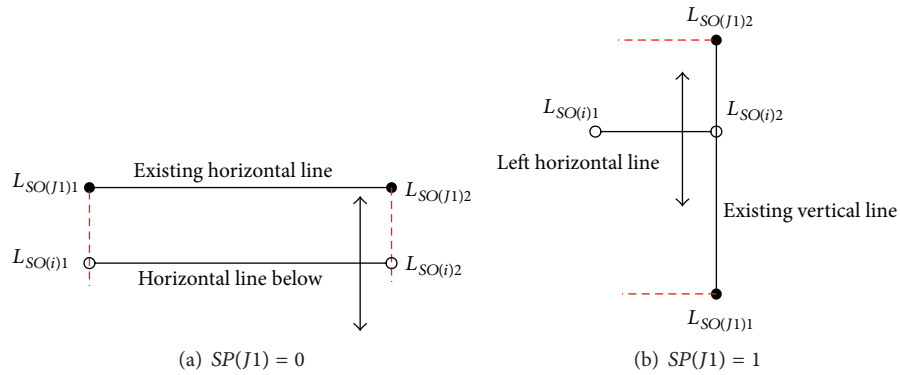


FIGURE 4: Case of current left set only belonging to some existing left set.

TABLE 1: Results of set partitioning.

Splitting position	3	2	4	1	5
Left set	$\{f_1, f_2, f_3\}$	$\{f_1, f_2\}$	$\{f_4\}$	$\{f_1\}$	$\{f_5\}$
Right set	$\{f_4, f_5, f_6\}$	$\{f_3\}$	$\{f_5, f_6\}$	$\{f_2\}$	$\{f_6\}$

lower. The facilities of vector F on the left of splitting point $SO(1)$, that is, $setL(1).leftset$, were included in the lower part. The facilities of vector F on the right of $SO(1)$, that is, $setR(1).rightset$, were included in the upper part. The first line was denoted by $L_{SO(1)1}(0, \sum_{i \in setL(1).leftset} v_i)$, $L_{SO(1)2}(\sum h_i, \sum_{i \in setL(1).leftset} v_i)$.

(ii) $SP(1) = 1$. When $SP(1) = 1$, the first line $L_{SO(1)1}L_{SO(1)2}$ is vertical and can be determined by $L_{SO(1)1}(\sum_{i \in setL(1).leftset} h_i, 0)$, $L_{SO(1)2}(\sum_{i \in setL(1).leftset} h_i, \sum v_i)$.

As shown in Algorithm 2, the first line can be drawn by connecting the point $(hlin_start(1), hlin_y(1))$ and point $(hlin_end(1), hlin_y(1))$. As a result, the layout area was partitioned into two parts.

3.3. Coordinates Determination of the Second and Subsequent Splitting Line. From the second splitting operation, inclusion relations between the left set obtained in current operation and the left set and right set obtained before were determined.

For the set including the current left set, its maximum of the structure attribute SO is recorded. Let $J1$ be the index of maximum SO in left set and let $J2$ be the index of maximum SO in right set.

(I) If the current splitting mode is horizontal, that is, $SP(i) = 0$, $i \geq 2$, the splitting line of current operation has the following cases according to the value of $J1$ and $J2$.

(i) Consider $J1 \neq 0, J2 = 0$. In this case, the left set obtained in current operation was included by only one existing left set. The determination of splitting line according to the value of $SP(J1)$ was as shown in Figure 4.

As shown in Figure 4(a), when $SP(J1) = 0$, the corresponding existing splitting line was a horizontal line $L_{SO(J1)1}L_{SO(J1)2}$, below which the current line $L_{SO(i)1}L_{SO(i)2}$ was drawn horizontally, where $L_{SO(i)1}(x_{L_{SO(J1)1}}, \sum_{j \in setL(i).leftset} v_j)$, $L_{SO(i)2}(x_{L_{SO(i)2}}, y_{L_{SO(i)1}})$; Figure 4(b) shows the other case of $SP(J1) = 1$. The corresponding existing splitting line $L_{SO(J1)1}L_{SO(J1)2}$ was vertical, on the left side of which the current line $L_{SO(i)1}L_{SO(i)2}$ was drawn, where $L_{SO(i)1}(0, \sum_{j \in setL(i).leftset} v_j)$, $L_{SO(i)2}(x_{L_{SO(J1)1}}, y_{L_{SO(i)1}})$.

(ii) Consider $J1 = 0, J2 \neq 0$. In this case, the left set obtained in current operation was included by only one existing right set. The determination of splitting line according to the value of $SP(J2)$ was as shown in Figure 5.

As shown in Figure 5(a), when $SP(J2) = 0$, the corresponding existing splitting line was a horizontal line

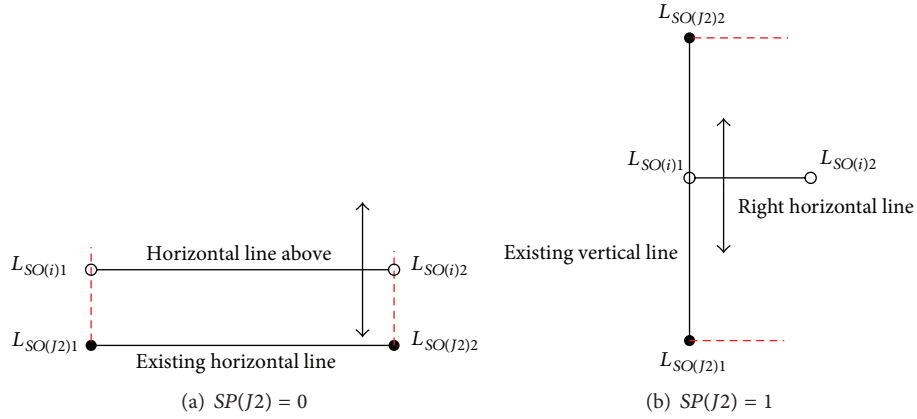


FIGURE 5: Case of current left set only belonging to some existing right set.

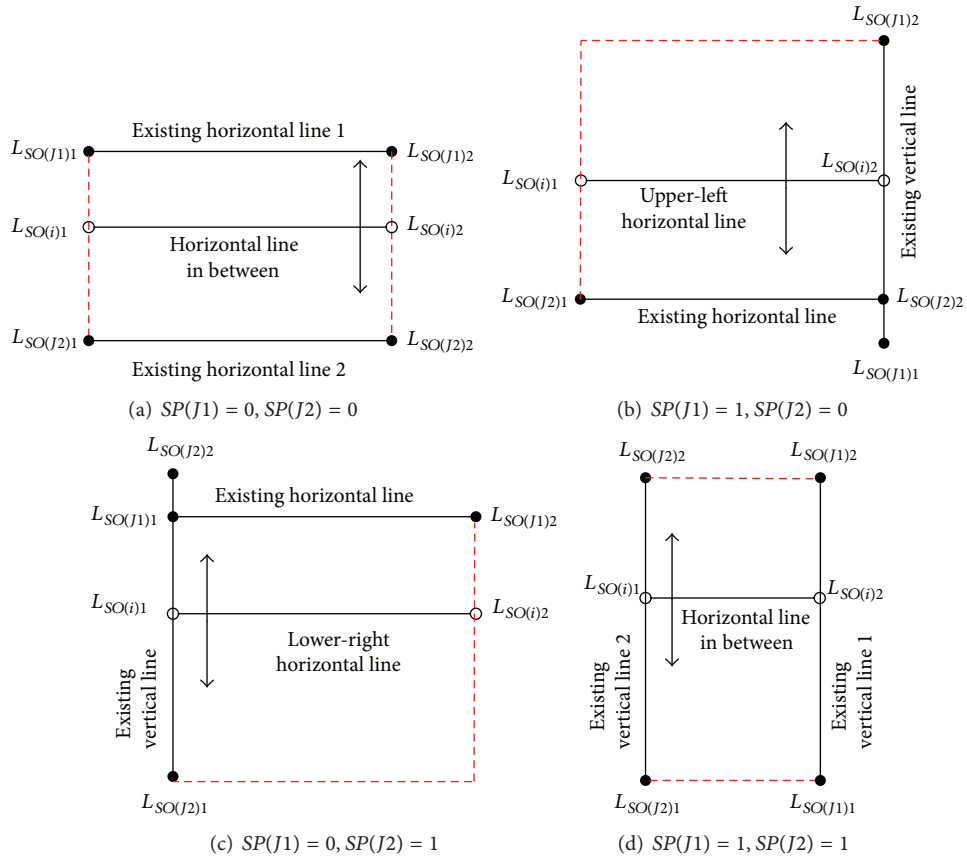


FIGURE 6: Case of current left set belonging to both some existing right set and some existing left set.

$L_{SO(J2)1}L_{SO(J2)2}$, above which the current line $L_{SO(i)1}L_{SO(i)2}$ was drawn, where $L_{SO(i)1}(x-L_{SO(J2)1}, y-L_{SO(J2)1} + \sum_{j \in \text{set}L(i), \text{leftset}} v_j)$, $L_{SO(i)2}(x-L_{SO(J2)2}, y-L_{SO(i)1})$; Figure 5(b) shows the other case of $SP(J2) = 1$. The corresponding existing splitting line $L_{SO(J1)1}L_{SO(J1)2}$ was vertical, on the right side of which the current line $L_{SO(i)1}L_{SO(i)2}$ was drawn, where $L_{SO(i)1}(x-L_{SO(J2)1}, \sum_{j \in \text{set}L(i), \text{leftset}} v_j)$, $L_{SO(i)2}(\sum h_i, y-L_{SO(i)1})$.

(iii) Consider $J1 \neq 0, J2 \neq 0$. In this case, the left set obtained in current operation belonged to both some existing right set and some existing left set. The determination of splitting line according to the value of $SP(J2)$ and $SP(J1)$ was as shown in Figure 6.

As shown in Figure 6(a), when $SP(J1) = 0$ and $SP(J2) = 0$, the corresponding two existing splitting lines were $L_{SO(J1)1}L_{SO(J1)2}$ and $L_{SO(J2)1}L_{SO(J2)2}$, and they were

Procedure (Drawing the second and subsequent splitting line) ($F, SP, setL, setR, LW, LH$)

- (1) for $i \leftarrow 2$ TO $N - 1$
- (2) $J1 \leftarrow 0; J2 \leftarrow 0;$
- (3) for $j = 1$ TO $i - 1$
- (4) if $setL(i).left\ set \subset setL(j).left\ set \ \& \ j > J1$
- (5) $J1 \leftarrow j$
- (6) elseif $setL(i).left\ set \subset setL(j).right\ set \ \& \ j > J2$
- (7) $J2 \leftarrow j$
- (8) end if
- (9) end for
- (10) [tf, index] $\leftarrow setL(i).elements$ and their index in F ;
- (11) if $J1 \sim= 0 \ \& \ J2 = 0$
- (12) if $SP(J1) = 0$
- (13) $hlin_start(i) \leftarrow hlin_start(J1); hlin_end(i) \leftarrow hlin_end(J1);$
- (14) $hlin_y(i) \leftarrow sum(LH(F(index)));$
- (15) elseif $SP(J1) = 1$
- (16) $hlin_start(i) \leftarrow 0; hlin_end(i) \leftarrow vlin_x(J1);$
- (17) $hlin_y(i) \leftarrow sum(LH(F(index)));$
- (18) end if
- (19) elseif $J1 = 0 \ \& \ J2 \sim= 0$
- (20) if $SP(J2) = 0$
- (21) $hlin_start(i) \leftarrow hlin_start(J2); hlin_end(i) \leftarrow hlin_end(J2); hlin_y(i) \leftarrow hlin_y(J2) + sum(LH(F(index)));$
- (22) elseif $SP(J2) = 1$
- (23) $hlin_start(i) \leftarrow vlin_x(J2); hlin_end(i) \leftarrow sum(LW); hlin_y(i) \leftarrow vlin_start(J2) + sum(LH(F(index)));$
- (24) end if
- (25) elseif $J1 \sim= 0 \ \& \ J2 \sim= 0$
- (26) if $SP(J1) = 0 \ \& \ SP(J2) = 0$
- (27) $hlin_start(i) \leftarrow hlin_start(J2); hlin_end(i) \leftarrow hlin_end(J2); hlin_y(i) \leftarrow hlin_y(J2) + sum(LH(F(index)));$
- (28) elseif $SP(J1) = 1 \ \& \ SP(J2) = 0$
- (29) $hlin_start(i) \leftarrow hlin_start(J2); hlin_end(i) \leftarrow vlin_end(J1); hlin_y(i) \leftarrow hlin_y(J2) + sum(LH(F(index)));$
- (30) elseif $SP(J1) = 0 \ \& \ SP(J2) = 1$
- (31) $hlin_start(i) \leftarrow vlin_x(J2); hlin_end(i) \leftarrow hlin_end(J1); hlin_y(i) \leftarrow vlin_start(J2) + sum(LH(F(index)));$
- (32) elseif $SP(J1) = 1 \ \& \ SP(J2) = 1$
- (33) $hlin_start(i) \leftarrow vlin_x(J2); hlin_end(i) \leftarrow vlin_x(J1); hlin_y(i) \leftarrow vlin_start(J2) + sum(LH(F(index)));$
- (34) end if
- (35) end if
- (36) Connect point ($hlin_start(i), hlin_y(i)$) and point ($hlin_end(i), hlin_y(i)$)
- (37) end for

ALGORITHM 3: Process of line coordinates determination.

horizontal. The current line $L_{SO(i)1}L_{SO(i)2}$ was drawn above the line $L_{SP(J2)1}L_{SP(J2)2}$ and below the line $L_{SO(J1)1}L_{SO(J1)2}$, where $L_{SO(i)1}(x_{L_{SO(J2)1}}, y_{L_{SO(J2)1}} + \sum_{j \in setL(i).leftset} v_j)$, $L_{SO(i)2}(x_{L_{SO(J2)2}}, y_{L_{SO(i)1}})$; in Figure 6(b), when $SP(J1) = 1$ and $SP(J2) = 0$, $L_{SO(i)1}L_{SO(i)2}$ was drawn above the existing horizontal line $L_{SO(J1)1}L_{SO(J1)2}$ and on the left side of existing vertical line $L_{SO(J2)1}L_{SO(J2)2}$, where $L_{SO(i)1}(x_{L_{SO(J2)1}}, y_{L_{SO(J2)1}} + \sum_{j \in setL(i).leftset} v_j)$, $L_{SO(i)2}(x_{L_{SO(J2)2}}, y_{L_{SO(J1)1}})$; in Figure 6(c), when $SP(J1) = 0$ and $SP(J2) = 1$, $L_{SO(i)1}L_{SO(i)2}$ was drawn below the existing horizontal line $L_{SO(J1)1}L_{SO(J1)2}$ and on the right side of existing vertical line $L_{SO(J2)1}L_{SO(J2)2}$, where $L_{SO(i)1}(x_{L_{SO(J2)1}}, y_{L_{SO(J2)1}} + \sum_{j \in setL(i).leftset} v_j)$, $L_{SO(i)2}(x_{L_{SO(J1)2}}, y_{L_{SO(i)1}})$; in Figure 6(d), when $SP(J1) = 1$ and $SP(J2) = 1$, $L_{SO(i)1}L_{SO(i)2}$ was drawn between the left side of the existing vertical line $L_{SO(J1)1}L_{SO(J1)2}$ and

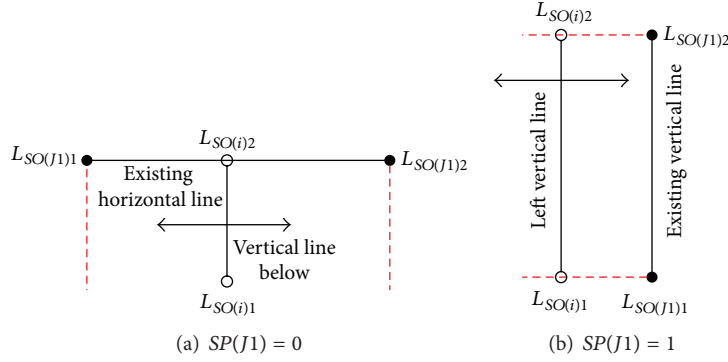
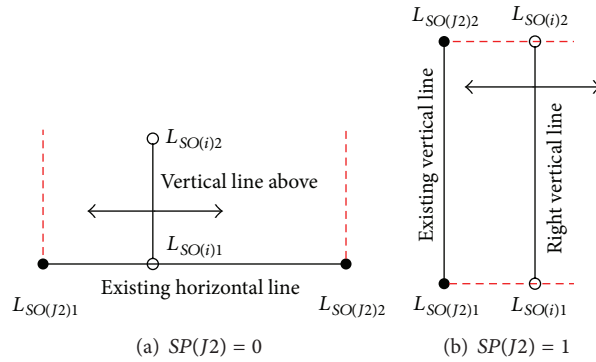
the right side of existing vertical line $L_{SO(J2)1}L_{SO(J2)2}$, where $L_{SO(i)1}(x_{L_{SO(J2)1}}, y_{L_{SO(J2)1}} + \sum_{j \in setL(i).leftset} v_j)$, $L_{SO(i)2}(x_{L_{SO(J1)1}}, y_{L_{SO(i)1}})$.

The above approach to determine the line coordinates and to draw the line was shown as the pseudocode in Algorithm 3.

(II) If the current splitting mode is horizontal, that is, $SP(i) = 1$, $i \geq 2$, the splitting line of current operation has the following cases according to the value of $J1$ and $J2$.

(i) Consider $J1 \neq 0, J2 = 0$. In this case, the left set obtained in current operation belonged to only one existing left set. The determination of splitting line according to the value of $SP(J1)$ was as shown in Figure 7.

As shown in Figure 7(a), when $SP(J1) = 0$, the corresponding existing splitting line was a horizontal line $L_{SO(J1)1}L_{SO(J1)2}$, below which the current line $L_{SO(i)1}L_{SO(i)2}$ was drawn vertically, where $L_{SO(i)1}(x_{L_{SO(J1)1}} +$


 FIGURE 7: Case of current left set only belonging to some existing left set when $SP(i) = 1$, $i \geq 2$.

 FIGURE 8: Case of current left set only belonging to some existing right set when $SP(i) = 1$, $i \geq 2$.

$\sum_{j \in \text{set}L(i).\text{leftset}} v_j, 0)$, $L_{SO(i)2}(x_{L_{SO(i)1}}, y_{L_{SO(J1)1}})$; Figure 7(b) showed the other case of $SP(J1) = 1$. The corresponding existing splitting line $L_{SO(J1)1}L_{SO(J1)2}$ was vertical, on the left side of which the current line $L_{SO(i)1}L_{SO(i)2}$ was drawn, where $L_{SO(i)1}(\sum_{j \in \text{set}L(i).\text{leftset}} v_j, y_{L_{SO(J1)1}})$, $L_{SO(i)2}(x_{L_{SO(i)1}}, y_{L_{SO(J1)2}})$.

(ii) Consider $J1 = 0$, $J2 \neq 0$. In this case, the left set obtained in current operation belonged to only one existing right set. The determination of splitting line according to the value of $SP(J2)$ was as shown in Figure 8.

As shown in Figure 8(a), when $SP(J2) = 0$, the corresponding existing splitting line was a horizontal line $L_{SO(J2)1}L_{SO(J2)2}$, above which the current line $L_{SO(i)1}L_{SO(i)2}$ was drawn vertically, where $L_{SO(i)1}(x_{L_{SO(J2)1}} + \sum_{j \in \text{set}L(i).\text{leftset}} v_j, 0)$, $L_{SO(i)2}(x_{L_{SO(i)1}}, \sum v_i)$; Figure 8(b) showed the other case of $SP(J2) = 1$. The corresponding existing splitting line $L_{SO(J2)1}L_{SO(J2)2}$ was vertical, on the left side of which the current line $L_{SO(i)1}L_{SO(i)2}$ was drawn, where $L_{SO(i)1}(x_{L_{SO(J2)1}} + \sum_{j \in \text{set}L(i).\text{leftset}} v_j, y_{L_{SO(J2)1}})$, $L_{SO(i)2}(x_{L_{SO(i)1}}, y_{L_{SO(J1)2}})$.

(iii) Consider $J1 \neq 0$, $J2 \neq 0$. In this case, the left set obtained in current operation belonged to both one existing right set and one existing left set. The determination of splitting line according to the value of $SP(J1)$ and $SP(J2)$ was as shown in Figure 9.

As shown in Figure 9(a), when $SP(J1) = 0$ and $SP(J2) = 0$, the corresponding two existing splitting lines were $L_{SO(J1)1}L_{SO(J1)2}$ and $L_{SO(J2)1}L_{SO(J2)2}$, and they

were horizontal. The current line $L_{SO(i)1}L_{SO(i)2}$ was drawn between line $L_{SP(J2)1}L_{SP(J2)2}$ and line $L_{SO(J1)1}L_{SO(J1)2}$, where $L_{SO(i)1}(x_{L_{SO(J2)1}} + \sum_{j \in \text{set}L(i).\text{leftset}} h_j, y_{L_{SO(J2)1}})$, $L_{SO(i)2}(x_{L_{SO(i)1}}, y_{L_{SO(J1)2}})$; in Figure 9(b), when $SP(J1) = 1$ and $SP(J2) = 0$, $L_{SO(i)1}L_{SO(i)2}$ was drawn above the existing horizontal line $L_{SO(J1)1}L_{SO(J1)2}$ and on the left side of existing vertical line $L_{SO(J2)1}L_{SO(J2)2}$, where $L_{SO(i)1}(x_{L_{SO(J2)1}} + \sum_{j \in \text{set}L(i).\text{leftset}} h_j, y_{L_{SO(J2)1}})$, $L_{SO(i)2}(x_{L_{SO(i)1}}, y_{L_{SO(J1)2}})$; in Figure 9(c), when $SP(J1) = 0$ and $SP(J2) = 1$, $L_{SO(i)1}L_{SO(i)2}$ was drawn below the existing horizontal line $L_{SO(J1)1}L_{SO(J1)2}$ and on the right side of existing vertical line $L_{SO(J2)1}L_{SO(J2)2}$, where $L_{SO(i)1}(x_{L_{SO(J2)1}} + \sum_{j \in \text{set}L(i).\text{leftset}} h_j, y_{L_{SO(J2)1}})$, $L_{SO(i)2}(x_{L_{SO(i)1}}, y_{L_{SO(J1)2}})$; in Figure 9(d), when $SP(J1) = 1$ and $SP(J2) = 1$, $L_{SO(i)1}L_{SO(i)2}$ was drawn between the left side of the existing vertical line $L_{SO(J1)1}L_{SO(J1)2}$ and the right side of existing vertical line $L_{SO(J2)1}L_{SO(J2)2}$, where $L_{SO(i)1}(x_{L_{SO(J2)1}} + \sum_{j \in \text{set}L(i).\text{leftset}} h_j, y_{L_{SO(J2)1}})$, $L_{SO(i)2}(x_{L_{SO(i)1}}, y_{L_{SO(J1)2}})$.

For the case of $SP(i) = 1$, $i \geq 2$, the pseudocode to realize the above approach can be achieved by replacing $hlin_start(i)$, $hlin_end(i)$, and $hlin_y(i)$ in Figure 5 with $vlin_start(i)$, $vlin_end(i)$, and $vlin_x(i)$, respectively. The coordinate values of each splitting line can be determined according to Figures 7~9.

Plane segmentation process is to determine facilities relative position in the layout area. For the aforementioned FLP with 6 facilities (as shown in Figure 3), the process of

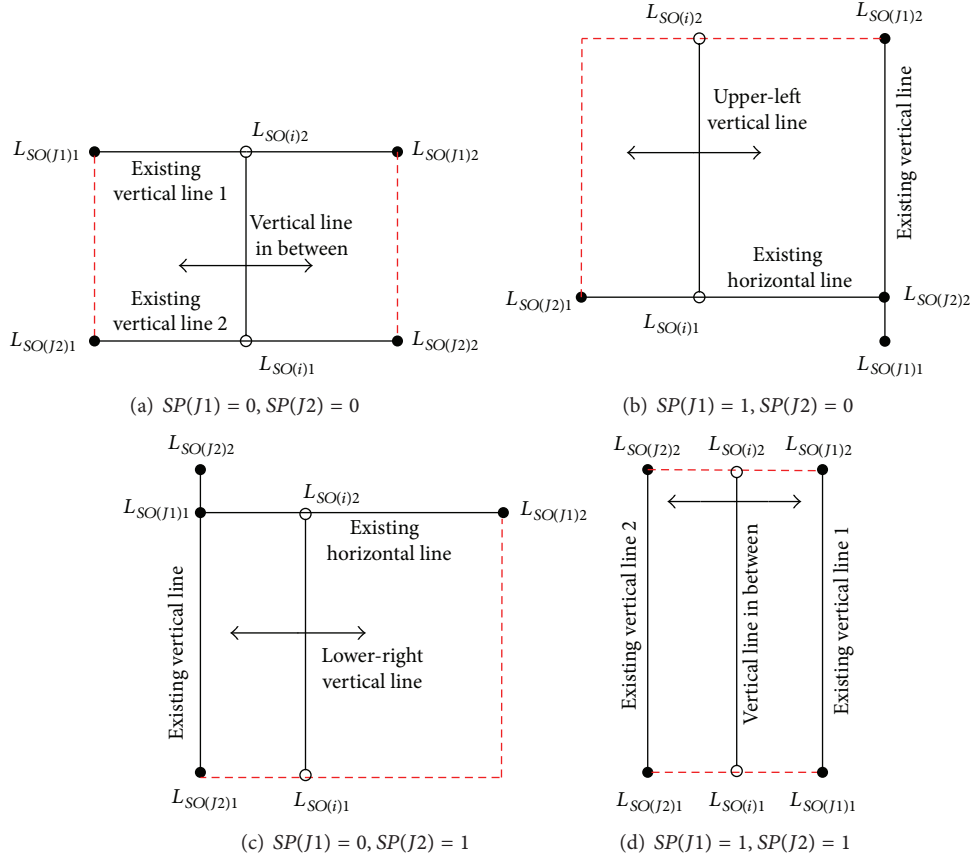


FIGURE 9: Case of current left set belonging to both some existing right set and some existing left set, $SP(i) = 1, i \geq 2$.

plane splitting and splitting line coordinate determining can be described as follows.

The initial layout area was a plane determined by the lower-left corner $O(0, 0)$ and top-right corner $O'(\sum h_i, \sum v_i)$. For the first splitting operation, the vertical splitting line $L_{31}L_{32}, L_{31}(\sum_{i=1,2,3} h_i, 0), L_{32}(\sum_{i=1,2,3} h_i, \sum v_i)$, divided the initial layout area into two parts, left and right side. The left was for facilities 1, 2, and 3, and the right was for facilities 4, 5, and 6. The second splitting line $L_{21}L_{22}, L_{21}(0, \sum_{i=1,2} v_i), L_{22}(\sum_{i=1,2,3} h_i, \sum_{i=1,2} v_i)$, was horizontal and partitioned the left side of the first time into two parts, lower and upper part. The lower area was for facilities 1 and 2, and the upper part was for facility 3. The third splitting line $L_{41}L_{42}, L_{41}(\sum_{i=1,2,3} h_i, v_4), L_{42}(\sum h_i, v_4)$, was horizontal and partitioned the right side of the first time into two parts, lower and upper. The upper area was for facilities 5 and 6, and the lower part was for facility 4. The fourth vertical splitting line $L_{11}L_{12}, L_{11}(v_1, 0), L_{12}(h_1, \sum_{i=1,2} v_i)$, divided the lower part of the second time into two sides, left and right. The left was for facility 1 and the right was for facility 2. The fifth horizontal splitting line $L_{51}L_{52}, L_{51}(\sum_{i=1,2,3} h_i, v_4 + v_5), L_{52}(\sum h_i, v_4 + v_5)$, divided the upper part of the third time into two sides, upper and lower. The upper one was for facility 6 and the lower one was for facility 5. So far, the plane was divided into blocks whose amount was equal to the facility number. Each block was for a special facility. The position

of the above 6 facilities can be located by a point pair of the lower-left corner and the upper-right corner of the 6 blocks as follows: $f_1\{O, L_{12}\}, f_2\{L_{11}, L_{22}\}, f_3\{L_{21}, L_{32}\}, f_4\{L_{31}, L_{42}\}, f_5\{L_{41}, L_{52}\}$, and $f_6\{L_{51}, O'\}$. So a feasible facility layout solution was obtained.

4. Case Study

4.1. Basic Information of Facilities. This paper took the FLP of a toolkit manufacturer as a case to validate the approach mentioned above. The area and aspect ratio of every facility were shown in Table 2.

4.2. Run Settings of Plane Segmentation Algorithm and the Results. According to the layout approach of plane splitting proposed in Section 3, the input information for this case is as follows: $F = \{f_1, f_2, \dots, f_{14}\}$, $SO = \{6, 3, 11, 7, 8, 5, 1, 2, 4, 13, 9, 10, 12\}$, and $SP = \{0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0\}$, and size matrix of facility *Aera* is as follows:

Aera

$$= \begin{bmatrix} 36 & 24 & 12 & 24 & 6 & 5 & 48 & 8 & 6 & 6 & 12 & 5 & 72 & 36 \\ 24 & 24 & 6 & 12 & 6 & 6 & 36 & 8 & 6 & 6 & 6 & 4 & 18 & 8 \end{bmatrix}. \quad (1)$$

TABLE 2: Facility information of a toolkit manufacturer.

Symbol	Name of facility	Facility function	Length (m)	Width (m)
f_1	Machining workshop	Blanking and punching	36	24
f_2	Welding workshop	Welding body parts	24	24
f_3	Painting workshop	Painting the body	12	6
f_4	Assembly workshop	Riveting the handle, lock, and hinge	24	12
f_5	Cargo-receiving areas	Receiving the supplied materials	6	6
f_6	Storage areas	Storing the rough parts	5	6
f_7	Warehouse	Storing the finished products	48	36
f_8	Shipping areas	Shipping the finished products	8	8
f_9	Lounge	For break	6	6
f_{10}	Maintenance area	Maintaining the equipment	6	6
f_{11}	Tool room	Storing the production tools	12	6
f_{12}	Changing room	Changing work clothes	5	4
f_{13}	Restaurant	Life services	72	18
f_{14}	Office	Office space	36	8

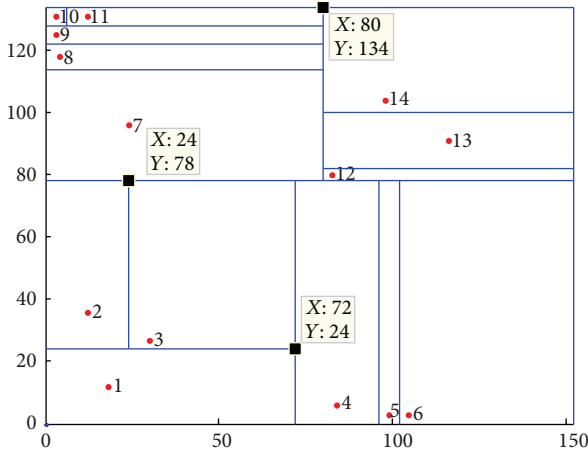


FIGURE 10: A feasible layout approach of FLP with 14 facilities using plane splitting method.

In *Aera*, the data in the first and second row denoted the length and width of each facility, respectively. By running the algorithm in Matlab, a feasible layout approach was obtained as in Figure 10. The Arabic numerals labeled in Figure 10 denoted the facility number, and the block that was labeled with a numeral was the layout area of the corresponding facility. Each block was located by the coordinates of lower-left corner and upper-right corner. The red point in each block denoted the centroid position of each facility.

From the layout scheme of Figure 10, it can be seen that facility 1 was in the lower-left corner of the plane, and, for facilities adjacent to facility 1, the facility with smaller number was located in the left and lower side and the facility with larger number was located in the right and upper side. This result was related to the order given by the input set. By changing the element order of F , SO , and SP , different feasible layout approaches can be obtained and then be further combined with heuristic intelligent algorithms to search the optimum. Figure 11 showed the different schemes of different F , SO , and SP .

4.3. *Analysis on the Influence of Facilities Quantity on the Algorithm.* In order to analyze the sensitivity of the proposed solutions to the quantity of the facilities, this paper selected 40 kinds of data with every 5th number from 5 to 200 to test. The machine parameters to run the algorithm were as follows: Intel Celeron CPU E3300, dual core, 2.5 GHz, 2 GB memory. The test results were displayed in Figure 12.

Figure 12 showed the running time and memory usage of algorithm to generate a feasible layout scheme under different facility quantity. It can be seen that facility quantity has a great influence on the algorithm performance, and the facility quantity was nearly linear with the running time and memory usage. When facility quantity was 20 and 100, the running time was 0.0506 s and 0.7871 s and the memory usage was 74.419 Kb and 118.035 Kb, respectively. Supposing that this method was combined with genetic algorithm (GA) for iterative optimization, the population size of genetic algorithm was 50 and the iteration number was 100, the running time of GA was $0.0506 * 50 * 100 = 253$ s and $0.7871 * 50 * 100 = 3935.5$ s, and the memory usage was $74.419 \text{ Kb} * 50 * 100 = 372.095 \text{ Mb}$ and $118.035 \text{ Kb} * 50 * 100 = 590.175 \text{ Mb}$ for the above two cases. So the proposed approach is suitable for the FLP of limited facility quantity.

5. Conclusions and Prospects

According to the characteristics of FLP with continuous block and unequal area facilities, this paper proposed plane splitting method to divide the layout area into the same number of blocks with the facility quantity and also developed the algorithm pseudocode to realize the method. The splitting process and the corresponding results were recorded in a data structure. Based on this and the splitting mode of each operation, the splitting line coordinates were determined. The results of case study show the effectiveness and the reliability of the proposed method.

The following conclusions are drawn from the above study:

- (i) It is reasonable and effective to solve FLP with continuous block and unequal area facilities by use of plane

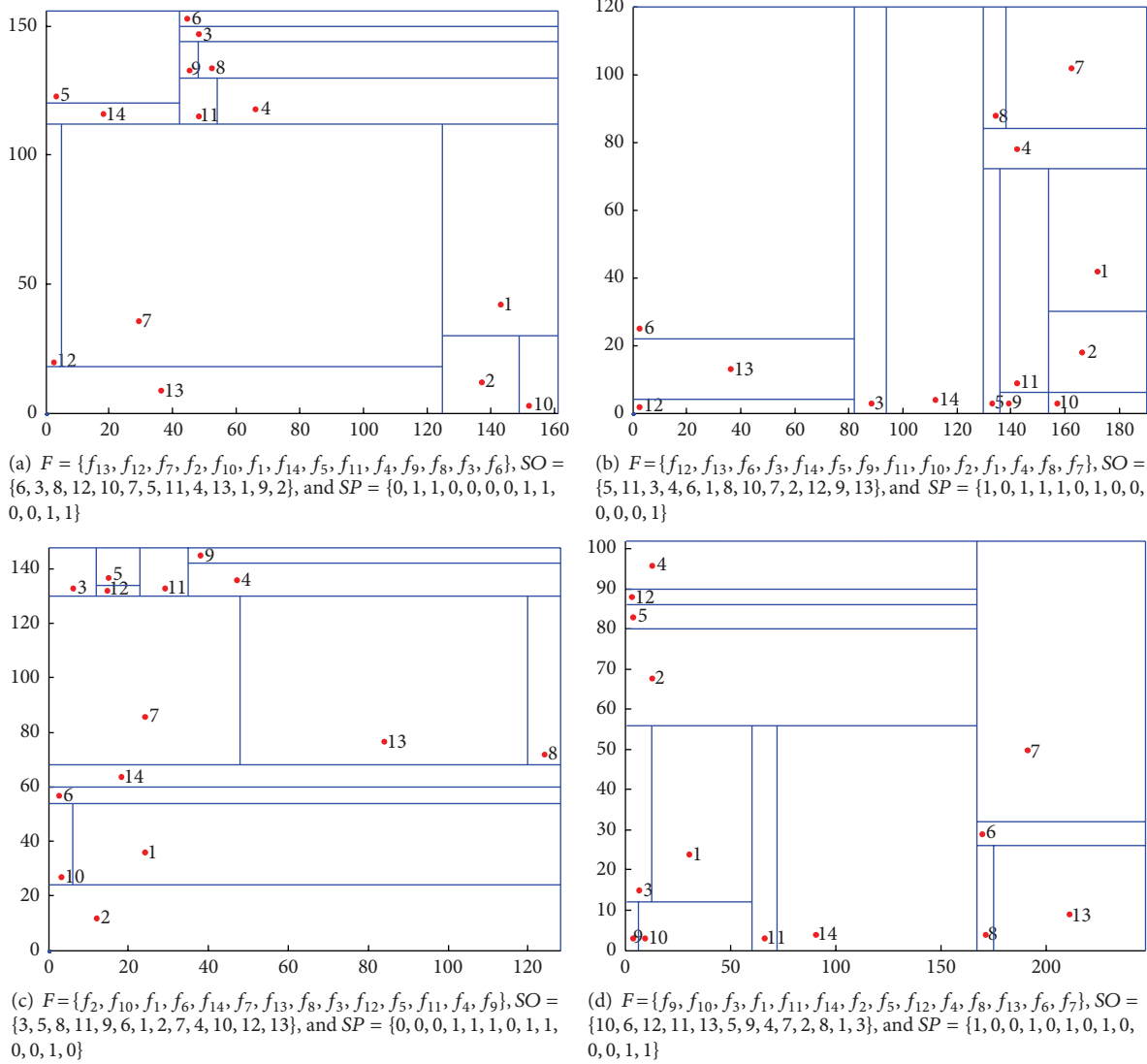


FIGURE 11: Feasible layout scheme corresponding to different element order of input set.

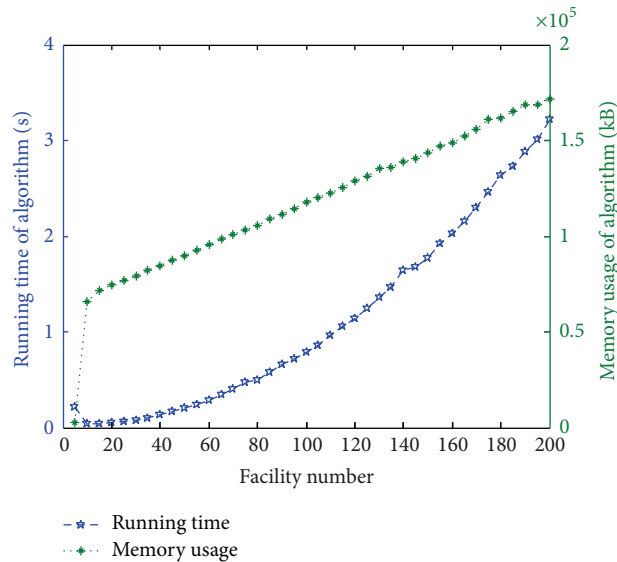


FIGURE 12: The relation curve between the facilities quantity and running time and memory usage.

segmentation method, which can divide the layout area into the same number of blocks with the facility quantity and so generate feasible layout scheme.

- (ii) The proposed plane segmentation algorithm can provide support to CAFP (computer aided facilities planning), generate feasible solution fast, and search the optimal layout approach combined with intelligent optimization algorithm.
- (iii) The encapsulated parameters of the proposed algorithm make it easy to solve different FLPs. When the FLP changed, the values of F , SO , and SP can be passed to the algorithm as parameters without changing the algorithm itself.

For particular facility layout goals, the combination of the proposed method with different heuristic intelligent optimization algorithm and the best combination scheme should be studied further. The proposed approach only considered the dimensions of facilities and did not analyze the impact of existing constraints within the layout area, which is common for FLP in renovated or expanded engineering. So it is worthy of further research.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This paper is supported by MOE (Ministry of Education in China) Project of Humanities and Social Sciences (Grant no. 13YJC630049), China, and the Natural Science Foundation of Shanxi Province (Grant no. 2013021021-2), China.

References

- [1] J. A. Tompkins, J. A. White, Y. A. Bozer, and J. M. A. Tanchoco, *Facilities Planning*, John Wiley & Sons, New York, NY, USA, 4th edition, 2010.
- [2] A. Drira, H. Pierreval, and S. Hajri-Gabouj, "Facility layout problems: a survey," *Annual Reviews in Control*, vol. 31, no. 2, pp. 255–267, 2007.
- [3] A. S. Ramkumar, S. G. Ponnambalam, and N. Jawahar, "A new iterated fast local search heuristic for solving QAP formulation in facility layout design," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 3, pp. 620–629, 2009.
- [4] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido, "A survey for the quadratic assignment problem," *European Journal of Operational Research*, vol. 176, no. 2, pp. 657–690, 2007.
- [5] Z.-Q. Zhang, S.-J. Tan, Y.-Z. Huang, and W.-M. Cheng, "Variable neighborhood search for single row layout problem," *China Mechanical Engineering*, vol. 24, no. 20, pp. 2791–2796, 2013.
- [6] X.-H. Suo and Z.-Q. Liu, "Modeling and simulation of single-row layout based on logistics routing," *China Mechanical Engineering*, vol. 18, no. 21, pp. 2576–2579, 2007.
- [7] H. Djellab and M. Gourgand, "A new heuristic procedure for the single-row facility layout problem," *International Journal of Computer Integrated Manufacturing*, vol. 14, no. 3, pp. 270–280, 2001.
- [8] R. Kothari and D. Ghosh, "Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods," *European Journal of Operational Research*, vol. 224, no. 1, pp. 93–100, 2013.
- [9] D. M. Simmons, "One-dimensional space allocation: an ordering algorithm," *Operations Research*, vol. 17, no. 5, pp. 812–826, 1969.
- [10] A. R. S. Amaral, "An exact approach to the one-dimensional facility layout problem," *Operations Research*, vol. 56, no. 4, pp. 1026–1033, 2008.
- [11] D. Datta, A. R. Amaral, and J. R. Figueira, "Single row facility layout problem using a permutation-based genetic algorithm," *European Journal of Operational Research*, vol. 213, no. 2, pp. 388–394, 2011.
- [12] Z.-Q. Zhang and W.-M. Cheng, "Decomposition strategies and heuristic for double row layout problem," *Computer Integrated Manufacturing Systems*, vol. 20, no. 3, pp. 559–568, 2014.
- [13] Z. Zhang and C. C. Murray, "A corrected formulation for the double row layout problem," *International Journal of Production Research*, vol. 50, no. 15, pp. 4220–4223, 2012.
- [14] C.-B. Kim, S. S. Kim, and B. L. Foote, "Assignment problems in single-row and double-row machine layouts during slow and peak periods," *Computers & Industrial Engineering*, vol. 30, no. 3, pp. 411–422, 1996.
- [15] C.-C. Chang, T.-H. Wu, and C.-W. Wu, "An efficient approach to determine cell formation, cell layout and intracellular machine sequence in cellular manufacturing systems," *Computers and Industrial Engineering*, vol. 66, no. 2, pp. 438–450, 2013.
- [16] I. Mahdavi, B. Shirazi, and M. M. Paydar, "A flow matrix-based heuristic algorithm for cell formation and layout design in cellular manufacturing system," *The International Journal of Advanced Manufacturing Technology*, vol. 39, no. 9-10, pp. 943–953, 2008.
- [17] C. C. Murray, A. E. Smith, and Z. Zhang, "An efficient local search heuristic for the double row layout problem with asymmetric material flow," *International Journal of Production Research*, vol. 51, no. 20, pp. 6129–6139, 2013.
- [18] R.-F. Yu, Y.-X. Wang, and Y.-Q. Zhu, "Optimizing research on facility layout problem in workplace based on human factors," *System Engineering Theory & Practice*, vol. 24, no. 3, pp. 40–45, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

