



Sharing clearances to improve machine layout

X.Q. Zuo, C.C. Murray & A.E. Smith

To cite this article: X.Q. Zuo, C.C. Murray & A.E. Smith (2016) Sharing clearances to improve machine layout, International Journal of Production Research, 54:14, 4272-4285, DOI: [10.1080/00207543.2016.1142134](https://doi.org/10.1080/00207543.2016.1142134)

To link to this article: <http://dx.doi.org/10.1080/00207543.2016.1142134>



Published online: 09 Feb 2016.



Submit your article to this journal [↗](#)



Article views: 118



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 1 View citing articles [↗](#)

Sharing clearances to improve machine layout

X.Q. Zuo^a, C.C. Murray^b and A.E. Smith^{c*}

^aComputer School, Beijing University of Posts and Telecommunications, Beijing, China; ^bDepartment of Industrial & Systems Engineering, University at Buffalo, Buffalo, NY, USA; ^cDepartment of Industrial & Systems Engineering, Auburn University, Auburn, AL, USA

(Received 8 May 2015; accepted 30 December 2015)

This paper considers a double-row layout problem with shared clearances in the context of semiconductor manufacturing. By sharing some clearances, reductions in both layout area and material handling cost of approximately 7–10% are achieved. Along with minimal clearances for separating adjacent machines, clearances that can be shared by adjacent machines are considered. The shared clearances may be located on either or both sides of machines. A mixed integer linear programming formulation of this problem is established, with the objective to minimise both material flow cost and layout area. A hybrid approach combining multi-objective tabu search and heuristic rules is proposed to solve it. Computational results show that the hybrid approach is very effective for this problem and finds machine layouts with reduced areas and handling costs by exploiting shared clearances.

Keywords: facility layout; double-row layout problem; multi-objective optimisation; tabu search; heuristic rules

1. Introduction

For some industries, such as semiconductor manufacturing, machines may require two types of clearance. The first type, *minimum clearance*, describes the spacing necessary for ventilation or to minimise vibration effects among adjacent machines. The second type, *additional clearance*, is necessary to allow technician access to one or both sides of the machine or to store work in process (WIP). These additional clearances are required in addition to the minimum clearances. Considering these two types of clearances as a single clearance may result in excessive spacing between machines, as there is often the possibility of sharing additional clearances among adjacent machines. For example, technicians rarely require access to maintenance panels of two side-by-side machines simultaneously. Thus, layout area can be saved by considering the minimum clearance and additional clearance independently. Since the material flow cost is related to the distance between machines, reducing the footprint of the manufacturing area is likely to reduce the overall material flow distance (and thus, cost) as well.

Among the numerous variants of facility layout problems, the *single-row layout problem* (SRLP) and the *multiple-row layout problem* (MRLP) are the most widely implemented layout patterns in the literature and in practice (Solimanpur, Vrat, and Shankar 2005). Both the SRLP and MRLP share a common objective of arranging machines to minimise the total cost of material flow (Castillo and Peters 2004; Chiang, Kouvelis, and Urban 2006; Jaramillo and McKendall 2010). These are combinatorial problems due to the assumption that all machines are separated by minimum clearances. The SRLP arranges machines in one row, which may be a straight line, semi-circular or U-shape, with each pair of adjacent machines being separated by a minimum allowable clearance (Djellab and Gourgand 2001; Hungerländer and Rendl 2013). Several exact approaches have been proposed for this NP-hard problem, including mixed-integer linear programming using a commercial solver (Amaral 2006), mixed binary linear programming (Amaral 2008), and a semi-definite programming relaxation with cutting planes (Anjos and Vannelli 2008). An improved lower bound was provided by Amaral (2009). Meta-heuristics, which search for optimal or near-optimal solutions using a large number of iterations, have also been proposed for the SRLP, including genetic algorithms (Datta, Amaral, and Figueira 2011; Ozcelik 2012), tabu search (Samarghandi and Eshghi 2010; Kothari and Ghosh 2013), particle swarm optimisation (Samarghandi, Taabayan, and Jahantigh 2010) and ant colony optimisation (Solimanpur, Vrat, and Shankar 2005).

The MRLP is usually formulated as a quadratic assignment problem, where machines are separated by their minimum clearances (Singh and Sharma 2008). A hybrid simulated annealing algorithm for the MRLP with facilities of equal area was proposed in Heragu and Alfa (1992). Genetic algorithms have been applied to solve the MRLP with machines of different sizes (Gen, Ida, and Cheng 1995) and to solve a multi-line layout problem (Sadrzadeh 2012).

*Corresponding author. Email: smithae@auburn.edu

Ficko, Brezocnik, and Balic (2004) proposed a genetic algorithm for single or multiple-row layouts in flexible manufacturing systems, with a solution expressed as permutations of machines.

Another interesting problem is the *double-row layout problem* (DRLP), first discussed by Heragu and Kusiak (1988) and formulated by Chung and Tanchoco (2010). Unlike the SRLP and MRLP, the DRLP requires the determination of not only the sequence of machines in each row (combinatorial aspect), but also the exact location of each machine (continuous aspect) in the layout space. Determining the exact location for each machine may result in a layout with less material handling cost than only choosing the sequence separated by minimum clearances. In addition to providing a mixed integer linear programming (MILP) formulation of the DRLP, Chung and Tanchoco (2010) proposed five heuristics for its solution. Zhang and Murray (2012) proposed a corrected formulation, noting that the formulation may ignore the minimum clearance between some adjacent facilities. Murray, Smith, and Zhang (2013) proposed a methodology combining a constructive heuristic and a local search for the DRLP with asymmetric flows. In our previous work (Murray, Zuo, and Smith 2012), an extended double-row layout problem (EDRLP) was studied in which non-zero aisle width is allowed and the optimisation objectives of both cost and layout area are considered by linearly combining them to form a single objective. A commercial IP solver was used to solve this EDRLP, allowing solutions to only small-scale problems. A multi-objective heuristic, suitable for larger problems, was developed in Zuo, Murray, and Smith (2014). Wang et al. (2015) proposed a simulated annealing algorithm combined with a mathematical programming approach to solve a dynamic version of the DRLP.

It is worth noting that other works have been published under the umbrella of a 'DRLP,' but with definitions that differ from Chung and Tanchoco (2010). For example, Amaral (2013a) describes a new formulation for a DRLP as an alternative to the formulation originally proposed by Chung and Tanchoco (2010). However, the new formulation prohibits space between adjacent machines, and may be more applicable to the *corridor allocation problem* (CAP), as proposed in Amaral (2012). Meta-heuristic approaches for the CAP have been recently proposed by Ahonen, de Alvarenga, and Amaral (2014). Because no space is allowed between two adjacent facilities in the CAP, the resulting problem is combinatorial. Amaral (2013b) defines the *parallel ordering problem*, where a subset of machines are pre-selected to be placed in one row. The remaining machines must be arranged in the opposite row. As with the CAP, this is a combinatorial problem, as machines are placed without space between them.

This paper seeks to make the following contributions. First, this paper defines a more complex and realistic version of the EDRLP that exploits shared clearances between machines, a problem we term the EDRLP-SC. Second, a MILP formulation of the problem is established. The formal definition of the EDRLP-SC and the corresponding MILP are provided in Section 2. Finally, because the solution approach to the EDRLP in Zuo, Murray, and Smith (2014) cannot be applied directly, this paper proposes a new approach to address the added complexity of shared clearances. This approach, described in Section 3, combines multi-objective tabu search (MTS) and heuristic rules (HR). The computational results of Section 4 indicate the effectiveness of this hybrid approach by comparing it against a popular general-purpose multi-objective optimisation algorithm.

2. EDRLP with shared clearances

In the EDRLP-SC, there are two parallel rows separated by an aisle of width c . Each machine is assumed to be rectangular with fixed width and depth dimensions. The load/unload port of each tool is located at the midpoint of the machine's width. Two types of clearance between machines are considered: minimum and additional (shared). A binary parameter, b_i , indicates whether additional clearance is required on *both* sides of machine i . If $b_i = 1$, then an additional clearance of a_i^l must be to the left of machine i , and an additional clearance of a_i^r must be to the right of i . If $b_i = 0$, then the additional clearance of either a_i^l or a_i^r will be on the appropriate side of machine i . A summary of the problem parameters is provided in Table 1.

An example EDRLP-SC instance is provided to more clearly introduce the problem. Consider a facility with $m = 6$ machines, each of width $w_i = 2$ and depth $d_i = 2$ for all $i = 1, 2, \dots, 6$. The minimum required clearance between any pair of machines is $c_{ij} = 1$, while the flow frequency times unit cost between each pair of machines is $f_{ij} = 1$. The aisle width separating the two rows is $c = 1$. Machines 2 and 3 require additional clearance on both sides (i.e. $b_2 = b_3 = 1$), while the remaining machines require additional clearance on only one side (i.e. $b_1 = b_4 = b_5 = b_6 = 0$). When required, machines 1, 4 and 5 have a left-side additional clearance of $a_1^l = a_4^l = a_5^l = 0.5$, while machines 2, 3 and 6 have additional clearances of $a_2^l = a_2^r = a_3^l = a_3^r = a_6^r = 1$.

The decision variables employed by the MILP formulation of the EDRLP-SC are defined in Table 2. Figure 1 depicts a solution (machine layout) to the example problem instance. Machines 1, 2 and 4 are placed in row 1, while machines 3, 5 and 6 are in row 2. This machine allocation and sequence establishes decision variable values for y_{ir} , z_{rij} , q_{ij} , e_{rij} and l_{ir} . In this particular solution, additional clearances are allocated on both sides of machines 2 and 3 because $b_2 = b_3 = 1$, on the left side of machines 1, 4 and 5, and on the right side of machine 6. Thus, decision variables p_i^l (p_i^r) and s_{rij} are determined. The total clearance between any two machines in the same row should be no smaller than the minimum required clearance

Table 1. Problem parameters.

m	Number of machines
I	Set of machines, where $I = \{1, \dots, m\}$
R	Set of rows, where $R = \{1, 2\}$ specifies row 1 and row 2
w_i	Width of machine $i \in I$
d_i	Depth of machine $i \in I$
c_{ij}	Minimum clearance required between machines $i \in I^1$ and $j \in I^2$, where $I^1 = \{1, \dots, m-1\}$ and $I^2 = \{i+1, \dots, m\}$ for all $i \in I^1$
$a_i^l (a_i^r)$	Additional clearance, not including c_{ij} , between the left (right) side of machine $i \in I$ and the right (left) side of the machine immediately to left (right) of machine i
b_i	A binary parameter indicating whether additional clearance must be provided on both sides of machine i
f_{ij}	Flow frequency times unit cost between machines $i \in I$ and $j \in I \setminus i$
c	Width of the aisle (corridor) separating row 1 from row 2

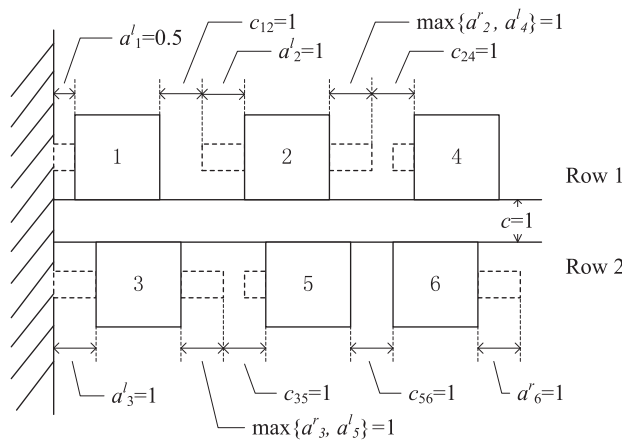


Figure 1. A solution to a sample instance of the EDRLP-SC. Parameter notation is defined in Table 1.

between them plus their additional clearances. For example, the minimum clearance between machines 1 and 2 is c_{12} and the additional clearance between them is $a_{12} = a_2^l$. Thus, the total clearance between them is $c_{12} + a_{12} = c_{12} + a_2^l = 2$. Note that additional clearances can be shared with each other. For example, additional clearance exists to the right (left) side of machine 2 (4), and machine 2 is immediately to the left of machine 4. In this case, the additional clearance between machines 2 and 4 can be shared (that is, the additional clearance is $a_{24} = \max\{a_2^r, a_4^l\}$), such that the minimum total clearance between machines 2 and 4 is $c_{24} + a_{24} = c_{24} + \max\{a_2^r, a_4^l\} = 2$. If no additional clearance exists between two machines, such as between machines 5 and 6, then the additional clearance between them is $a_{56} = 0$. In this case, their minimum total clearance equals the minimum required clearance between them, namely c_{56} . By this means, all decision variables a_{ij} may be determined.

In general, optimal solutions to an EDRLP-SC may correspond to adjacent machines being separated by a distance greater than the minimum required total clearance. However, for the specific solution shown in Figure 1, each pair of adjacent machines is separated by the minimum total clearance between them. Thus, the exact location of each machine, the layout width and the layout area can be identified as follows. The exact locations of machines are given by $x_1 = 1.5$, $x_2 = 5.5$, and $x_4 = 9.5$ in row 1, and by $x_3 = 2.0$, $x_5 = 6.0$, and $x_6 = 9.0$ in row 2. The width of resulting layout is $W = \max\{x_4 + w_4/2, x_6 + w_6/2 + a_6^r\} = 11$. This leads to the determination of area consumed in row 1 ($A_1^{\text{row}} = \max\{d_1, d_2, d_4\}W = (2)(11) = 22$) and in row 2 ($A_2^{\text{row}} = \max\{d_3, d_5, d_6\}W = (2)(11) = 22$). The total area consumed by this layout is given by $A = A_1^{\text{row}} + A_2^{\text{row}} = 22 + 22 = 44$.

The problem involves two objectives of material flow cost and layout area. Using problem parameter f_{ij} and decision variables x_i and q_{ij} , the objective function value of cost can be calculated by Equation (2) of the MILP formulation introduced below. The cost objective value is 141 for the layout (solution) in Figure 1. As shown in Equation (3), the objective function value of area equals the value of $A = 44$.

Table 2. Decision variables.

x_i	Continuous decision variable representing the absolute location of machine $i \in I$
y_{ir}	Binary decision variable, such that $y_{ir} = 1$ if machine $i \in I$ is placed in row $r \in R$; otherwise, $y_{ir} = 0$
z_{rij}	Binary decision variable, such that $z_{rij} = 1$ if machine $i \in I$ is placed <i>anywhere</i> to the left of machine $j \in I \setminus i$ in row $r \in R$; otherwise, $z_{rij} = 0$
e_{rij}	Binary decision variable, such that $e_{rij} = 1$ if machine $i \in I$ is placed <i>immediately</i> to the left of machine $j \in I \setminus i$ in row $r \in R$; otherwise, $e_{rij} = 0$
q_{ij}	Binary decision variable, such that $q_{ij} = 1$ if machines $i \in I^1$ and $j \in I^2$ are placed in the same row; otherwise, $q_{ij} = 0$
a_{ij}	Additional distance between machines $i \in I^1$ and $j \in I^2$ due to their additional clearance requirements
s_{rij}	Binary decision variable, such that $s_{rij} = 1$ if machine $i \in I$ is placed immediately to the left of machine $j \in I \setminus i$ in row $r \in R$ and machines i and j can share their additional clearances; otherwise, $s_{rij} = 0$
$p_i^l (p_i^r)$	Binary decision variable, such that $p_i^l = 1$ ($p_i^r = 1$) if the additional clearance of a_i^l (a_i^r) is applied to the left (right) of machine $i \in I$
l_{ir}	Integer decision variable, such that l_{ir} equals to the number of machines placed to the left of machine $i \in I$ in row $r \in R$
W	Width of the resulting layout
A_r^{row}	Area consumed by the machines in row $r \in R$
A	Total area consumed by the resulting layout, as determined by the area of the smallest rectangle enclosing all machines with their additional clearances
v_{ij}^+, v_{ij}^-	Auxiliary decision variables employed to determine the horizontal distance between machines i and j

We establish a MILP formulation below to formally describe the EDRLP-SC.

$$\text{Min } Obj = \{Obj_1, Obj_2\} \tag{1}$$

$$Obj_1 = \sum_{i \in I^1} \sum_{j \in I^2} (f_{ij} + f_{ji}) (v_{ij}^+ + v_{ij}^- + c(1 - q_{ij})) \tag{2}$$

$$Obj_2 = A \tag{3}$$

$$\text{s.t. } \sum_{r \in R} y_{ir} = 1 \quad \forall i \in I, \tag{4}$$

$$\frac{w_i y_{ir} + w_j y_{jr}}{2} + c_{ij} z_{rji} + a_{ij} \leq x_i - x_j + M(1 - z_{rji}) \quad \forall i \in I^1, j \in I^2, r \in R, \tag{5}$$

$$\frac{w_i y_{ir} + w_j y_{jr}}{2} + c_{ij} z_{rij} + a_{ij} \leq x_j - x_i + M(1 - z_{rij}) \quad \forall i \in I^1, j \in I^2, r \in R, \tag{6}$$

$$M(1 - z_{rij}) + a_{ij} \geq a_i^r p_i^r + a_j^l p_j^l - \min\{a_i^r, a_j^l\} s_{rij} \quad \forall i \in I^1, j \in I^2, r \in R, \tag{7}$$

$$M(1 - z_{rji}) + a_{ij} \geq a_j^r p_j^r + a_i^l p_i^l - \min\{a_j^r, a_i^l\} s_{rji} \quad \forall i \in I^1, j \in I^2, r \in R, \tag{8}$$

$$a_{ij} \leq a_i^r p_i^r + a_j^l p_j^l - \min\{a_i^r, a_j^l\} s_{rij} + M(1 - z_{rij}) \quad \forall i \in I^1, j \in I^2, r \in R, \tag{9}$$

$$a_{ij} \leq a_j^r p_j^r + a_i^l p_i^l - \min\{a_j^r, a_i^l\} s_{rji} + M(1 - z_{rji}) \quad \forall i \in I^1, j \in I^2, r \in R, \tag{10}$$

$$Mq_{ij} \geq a_{ij} \quad \forall i \in I^1, j \in I^2, \tag{11}$$

$$\sum_{j \in I \setminus i} z_{rji} = l_{ir} \quad \forall i \in I, r \in R, \tag{12}$$

$$N(1 - z_{rij}) + l_{jr} - l_{ir} + e_{rij} \geq 2 \quad \forall i \in I, j \in I \setminus i, r \in R, \tag{13}$$

$$e_{rij} \leq z_{rij} \quad \forall i \in I, j \in I \setminus i, r \in R, \tag{14}$$

$$\sum_{\substack{i \in I \\ j \in I \setminus i}} e_{rij} \leq \sum_{i \in I} y_{ir} - 1 \quad \forall r \in R, \tag{15}$$

$$s_{rij} \leq e_{rij} \quad \forall i \in I, j \in I \setminus i, r \in R, \tag{16}$$

$$s_{rij} \geq p_i^r + p_j^l + e_{rij} - 2 \quad \forall i \in I, j \in I \setminus i, r \in R, \tag{17}$$

$$\sum_{\substack{r \in R \\ j \in I \setminus i}} s_{rij} \leq p_i^r \quad \forall i \in I, \tag{18}$$

$$\sum_{\substack{r \in R \\ i \in I \setminus j}} s_{rij} \leq p_j^l \quad \forall j \in I, \quad (19)$$

$$p_i^r + p_i^l \geq b_i + 1 \quad \forall i \in I, \quad (20)$$

$$z_{rij} + z_{rji} \leq y_{ir} \quad \forall i \in I^1, j \in I^2, r \in R, \quad (21)$$

$$z_{rij} + z_{rji} \leq y_{jr} \quad \forall i \in I^1, j \in I^2, r \in R, \quad (22)$$

$$z_{rij} + z_{rji} + 1 \geq y_{ir} + y_{jr} \quad \forall i \in I^1, j \in I^2, r \in R, \quad (23)$$

$$q_{ij} = \sum_{r \in R} (z_{rij} + z_{rji}) \quad \forall i \in I^1, j \in I^2, \quad (24)$$

$$x_i - \frac{1}{2}w_i - a_i^l p_i^l \geq 0 \quad \forall i \in I, \quad (25)$$

$$W \geq x_i + \frac{1}{2}w_i + a_i^r p_i^r \quad \forall i \in I, \quad (26)$$

$$A_r^{\text{row}} \geq d_i W - d_i M(1 - y_{ir}) \quad \forall i \in I, r \in R, \quad (27)$$

$$A = A_1^{\text{row}} + A_2^{\text{row}} + cW, \quad (28)$$

$$x_i - x_j = v_{ij}^- - v_{ij}^+ \quad \forall i \in I^1, j \in I^2, \quad (29)$$

$$y_{ir} \in \{0, 1\}; l_{ir} \in \mathbb{Z}^+ \quad \forall i \in I, r \in R, \quad (30)$$

$$e_{rij}, s_{rij}, z_{rij} \in \{0, 1\} \quad \forall i \in I, j \in I \setminus i, r \in R, \quad (31)$$

$$a_{ij}, v_{ij}^+, v_{ij}^- \geq 0; q_{ij} \in \{0, 1\} \quad \forall i \in I^1, j \in I^2, \quad (32)$$

$$x_i \geq 0; p_i^l, p_i^r \in \{0, 1\} \quad \forall i \in I, \quad (33)$$

$$A_r^{\text{row}} \geq 0 \quad \forall r \in R, \quad (34)$$

$$A, W \geq 0. \quad (35)$$

The objective function (1) seeks to minimise material flow cost (2) and layout area (3) simultaneously. Constraint (4) ensures that each machine is placed in exactly one row. Constraints (5) and (6) guarantee that the minimum and additional clearance requirements between any two machines are satisfied. The constant M is a sufficiently large number. Constraints (7)–(10) determine the additional distance between any two machines. Constraint (11) forces decision variable a_{ij} equal to zero if machines i and j are located in different rows.

Constraints (12)–(15) serve to establish proper values for e_{rij} . The constant N may be given by $N = 2m$. Constraints (16)–(20) determine appropriate values for s_{rij} . Constraints (21)–(23) relate decision variables z_{rij} and y_{ir} , such that when machines i and j are both assigned to row r (i.e. $y_{ir} = y_{jr} = 1$), either z_{rij} or z_{rji} should be equal to 1; otherwise, $z_{rij} = z_{rji} = 0$. Constraint (24) determines whether machines i and j are in the same row. Constraint (25) guarantees the additional clearance between the left wall and machine i . Constraints (26)–(27) are employed to determine lower bounds on the width (horizontal dimension) and area of a layout. Constraint (28) determines the total area of the resulting layout. Constraint (29) determines the horizontal distance between machines i and j . Finally, (30)–(35) describe the decision variable definitions.

3. Proposed solution approach

The EDRLP-SC requires specification of the sequences of machines in both rows, additional clearances and exact machine locations. Due to the NP-hard property of the machine sorting problem, a multiobjective tabu search is chosen as it performs effectively for combinatorial optimisation problems. HR are devised to determine additional clearances during the search of MTS. Mathematical programming (MP) is used to find the optimal exact machine locations and additional clearances for each machine sequence found by MTS. The proposed MTS-HR consists of two parts: (1) find the set of non-dominated machine sequences by combining MTS and HR; (2) use MP to obtain the set of non-dominated solutions for all non-dominated machine sequences found, where the final Pareto solutions of EDRLP-SC are generated from the set of non-dominated solutions.

3.1 Finding the set of non-dominated machine sequences

Tabu search has proven superior in many combinatorial applications (Kulturel-Konak, Smith, and Norman 2006) including facility layout problems (Samarghandi and Eshghi 2010). Therefore, a MTS is adopted to find the set of non-dominated machine sequences.

In the MTS, each solution S represents a sequence of machines in both rows. A *move* is the swap of locations of any two machines. All moves of the current solution, S_c , comprise a move set $V(S_c)$. All solutions that can be reached by one move from S_c comprise the neighbourhood of S_c , denoted by $N(S_c)$. The tabu list, T , contains the most-recently swapped machine pairs. If T is full, the earliest move in the list is deleted and the current move is added. A dynamic length tabu list, where every 20 iterations the length of the tabu list changes between $minTabu$ and $maxTabu$, is used. The steps of the proposed MTS procedure are described in Algorithm 1.

Algorithm 1. Multi-objective tabu search

- Step 1: Initialisation. A feasible initial solution is randomly produced as the current solution S_c . Let $T = \emptyset$, $Archive = \emptyset$, and $iter = 0$.
- Step 2: A single objective is randomly selected as the active one from the two objectives in Equations (2) and (3).
- Step 3: Construct neighbourhood set $N(S_c)$ of S_c . Calculate the objective function value of each neighbour in $N(S_c)$.
- Step 4: Construct candidate solution set of S_c , denoted by $D(S_c)$. Set $D(S_c)$ contains all non-tabu solutions in $N(S_c)$ and those that are in $N(S_c)$ and T but dominate any individual solution in $Archive$ (an aspiration criterion).
- Step 5: Select the best solution S' from $D(S_c)$ to replace the current solution, i.e. $S_c = S'$. Update T and $Archive$ using S' .
- Step 6: An insertion local search is used to change the location of a machine from one row to the other. Construct the insertion set $E(S_c)$ of S_c , and select the best solution S'' from $E(S_c)$. If the objective function value of S'' is better than that of S_c , then let $S_c = S''$; otherwise, S'' is discarded. Update $Archive$ using S'' .
- Step 7: If $Archive$ is not updated over $reintMax$ iterations, then one of solutions in $Archive$ is randomly selected to replace S_c , and let $T = \emptyset$.
- Step 8: Let $iter = iter + 1$. If the number of iterations over which $Archive$ set is not updated is less than $maxIter$, return to Step 2; otherwise, the algorithm stops.
-

In the MTS, each solution (machine sequence), S , may correspond to an infinite number of solutions to EDRLP-SC because the machines can be placed at different exact locations for the given machine sequence. Assume that the set of all solutions corresponding to S is $AD(S)$. Thus, the most appropriate solution from $AD(S)$ to evaluate S must be determined. According to Zuo, Murray, and Smith (2014), the best option is to evaluate S by MP. However, this requires running MP for each evaluation, which is computationally intensive, and as such, this method is only used to evaluate the best machine sequences in $N(S)$ and $E(S)$, namely S' and S'' , before updating $Archive$. The detailed evaluation procedure is introduced in Section 3.1.1.

To reduce computational time, the candidate solutions (machine sequences) in $N(S)$ and $E(S)$ are evaluated by an approximate method. For a candidate machine sequence, S , if the optimal decision variables p_i^l and p_i^r ($i \in I$) for the area objective can be found, then a layout (denoted by X_S) can be readily constructed by arranging machines one by one from left to right in each row according to the sequence S , separating machines by their minimum and additional clearances. The X_S must be the solution with the minimum area among $AD(S)$ and is used to evaluate S . In this paper, we design simple rules to determine p_i^l and p_i^r quickly. The pseudo-code of HR is introduced in Section 3.1.2.

The solution representation, move (insertion) operator and the procedure of updating the archival set are the same as in Zuo, Murray, and Smith (2014).

3.1.1 Mathematical programming

For a machine sequence $S \in \{S', S''\}$, there exist two special solutions in $AD(S)$. One is the solution with minimal cost, X_c , and the other is the solution with minimal area, X_a . The cost value of X_c and the area value of X_a are regarded as the cost and area objective values of S , respectively. Algorithm 2 describes the method used to find the two solutions quickly.

Algorithm 2. Determining two boundary solutions

- Step 1: The S is used to fix the machine-sequence-related binary and integer decision variables z_{rij} , q_{ij} , y_{ir} , e_{rij} and l_{ir} in the formulation from Section 2, to obtain a simple mixed binary programming formulation P .
- Step 2: Considering Obj_2 (area) as the only objective function, CPLEX is used to solve P , obtaining values of all decision variables p_i^l , p_i^r , x_i , which indicate a solution with minimal area, X_a .
- Step 3: Considering Obj_1 (cost) as the only objective function, CPLEX is used to solve P , obtaining a solution with minimal cost, X_c .
-

3.1.2 Heuristic rules

Finding the optimal p_i^l and p_i^r ($i \in I$) is a combinatorial problem that is time-consuming. In this paper, we design the following HR to quickly determine p_i^l and p_i^r , encouraging adjacent machines to share their additional clearances to minimise area and handling cost. Let m_r represent the number of machines in the current row; $mach[i]$ indicates the machine in the i th position of the current row; $mk[i]$ indicates whether p_i^l and p_i^r have been processed by the rules. The HR are described in Algorithm 3.

Algorithm 3. Heuristic rules

```

Let  $p_i^l = 1, p_i^r = 1, mk[i] = 1$  for all  $i \in \{i | b_i = 1, i \in I\}$ ;
Let  $mk[i] = 0$  for all  $i \in \{i | b_i = 0, i \in I\}$ ;
for (each row) do
   $i = 1$ ;
  while ( $i \leq m_r$ ) do
    if ( $b_{mach[i]} = 0$ ) then
      if ( $i = 1$ ) then
         $p_{mach[i]}^l = 0; p_{mach[i]}^r = 1$ ;
      else if ( $i = m_r$ ) then
         $p_{mach[i]}^l = 1; p_{mach[i]}^r = 0$ ;
      else
        if ( $mk[mach[i-1]] = mk[mach[i+1]] = 1$ ) then
          if ( $p_{mach[i-1]}^r = 1$  and  $p_{mach[i+1]}^l = 1$ ) then
             $t1 = \max \{a_{mach[i-1]}^r, a_{mach[i]}^l\} + a_{mach[i+1]}^l$ 
             $t2 = \max \{a_{mach[i]}^r, a_{mach[i+1]}^l\} + a_{mach[i-1]}^r$ 
            if ( $t1 \geq t2$ ) then
               $p_{mach[i]}^l = 0, p_{mach[i]}^r = 1$ ;
            else
               $p_{mach[i]}^l = 1, p_{mach[i]}^r = 0$ ;
            end if
          else
             $t1 = a_{mach[i]}^l + a_{mach[i+1]}^l$ 
             $t2 = \max \{a_{mach[i]}^r, a_{mach[i+1]}^l\}$ 
            if ( $t1 \geq t2$ ) then
               $p_{mach[i]}^l = 0, p_{mach[i]}^r = 1$ ;
            else
               $p_{mach[i]}^l = 1, p_{mach[i]}^r = 0$ ;
            end if
          end if
        else
          if ( $p_{mach[i-1]}^r = 1$ ) then
             $p_{mach[i]}^l = 1$  and  $p_{mach[i]}^r = 0$ ;
          else
             $p_{mach[i]}^l = 0$  and  $p_{mach[i]}^r = 1$ ;
          end if
        end if
      end if
       $mk[mach[i]] = 1$ ;
    end if
     $i = i + 1$ ;
  end while
end for

```

First, p_i^l and p_i^r are set to 1 for each machine i with $b_i = 1$, and these machines are marked. For each machine i whose $b_i = 0$, its p_i^l and p_i^r need to be determined. If machine i is located at the leftmost (rightmost) position in a row, then let $p_i^r = 1$ ($p_i^l = 1$) to make its additional clearance have the opportunity to be shared with others. If the two machines immediately adjacent to both sides of machine i are marked, then let either $p_i^r = 1$ or $p_i^l = 1$ to make the additional distance

Table 3. Algorithm parameters.

Problem Instances	<i>maxIter</i>	<i>reintMax</i>	<i>minTabu</i>	<i>maxTabu</i>
$P_{10}^{0.1} - P_{10}^{0.9}$	400	50	2	4
$P_{20}^{0.1} - P_{20}^{0.9}$	1000	500	5	8
$P_{30}^{0.1} - P_{30}^{0.9}$	1000	500	7	11
$P_{50}^{0.1} - P_{50}^{0.9}$	1500	750	9	14

between the two machines be minimum. If only the machine immediately to the left of machine i is marked (only marking the machine immediately to the right will never happen because machines are marked from left to right), then let $p_i^l = 1$ ($p_i^r = 1$) if the additional clearance of the marked machine is placed on the right (left) side, allowing the additional clearance of machine i to be either shared with that of the marked machine or a future placed machine.

3.2 Finding the set of final non-dominated solutions

After the MTS stops, for each non-dominated machine sequence, S , stored in the set *Archive*, we can generate a set of non-dominated solutions, denoted by $C(S) \subseteq AD(S)$. Then, the set of final non-dominated solutions is generated by

$$FS = \{s | s \in NS \text{ and } \nexists s' \in NS : s' \text{ dominates } s\}, \quad (36)$$

where

$$NS = \bigcup_{S \in \text{Archive}} C(S). \quad (37)$$

The procedure of generating $C(S)$ is as follows. Suppose that W_c and W_a are the width of the two boundary solutions X_c and X_a , respectively. Since the depth of a solution is determined by its deepest machine in each row, each solution in $AD(S)$ has the same depth, D . Thus, the area values of X_c and X_a are $W_c \times D$ and $W_a \times D$, respectively. The width of a non-dominated solution in $AD(S)$ must be in the range $[W_c, W_a]$. We let W increase from W_c to W_a by a small step size; for each W , the sequence S is used to fix machine-sequence-related decision variables from the formulation in Section 2 and the constraint $A \leq D \times W$ (where A is the area decision variable) is added into the formulation; then, the MP can find a solution with the minimal cost for each width W by solving the formulation. By this means, a set of solutions, $C(S)$, can be obtained.

4. Computational experience

The EDRLP-SC formulation has more parameters than the EDRLP, including a_i^l , a_i^r , and b_i . We let a_i^l and $a_i^r \sim \text{unif}[0.2w_i, 0.25w_i]$, and $c \sim \text{unif}[0, \max_{i \in I} \{d_i\}/2]$. Parameter b_i , which indicates whether both p_i^l and p_i^r must be set to 1, is created such that $b_i = 1$ for a predetermined percentage of machines. The settings of other parameters are the same as the EDRLP in (Zuo, Murray, and Smith 2014).

Problem instances involving 10, 20, 30 and 50 machines are used to test our algorithm, as these are reasonable for most production environments. For each problem size, five representative problem instances are generated whose parameters are produced randomly. Instances $P_{10}^{0.1} (P_{20}^{0.1}, P_{30}^{0.1}, P_{50}^{0.1})$, $P_{10}^{0.3} (P_{20}^{0.3}, P_{30}^{0.3}, P_{50}^{0.3})$, $P_{10}^{0.5} (P_{20}^{0.5}, P_{30}^{0.5}, P_{50}^{0.5})$, $P_{10}^{0.7} (P_{20}^{0.7}, P_{30}^{0.7}, P_{50}^{0.7})$, and $P_{10}^{0.9} (P_{20}^{0.9}, P_{30}^{0.9}, P_{50}^{0.9})$ contain 10 (20, 30, 50) machines, where $b_i = 1$ for 10, 30, 50, 70, and 90% of machines, respectively. All test problem instances are available for download from the publisher, as an electronic companion to this paper.

4.1 Algorithm parameters

The parameters of MTS-HR include those of the MTS in addition to the step size of the area A in MP. The step size of area in MP depends on the number of solutions in FS (see Equation (36)) that should be identified. For more Pareto solutions, a smaller step size should be used. The step size for MP is fixed at 0.1, 0.5, 0.5 and 1.0 for problems with 10, 20, 30, 50 machines, respectively. The archive size is an upper bound on the number of non-dominated solutions stored in *Archive* and has no impact on MTS performance. The archive size is set to 30 for all problem instances. Other parameters are presented in Table 3. Parameter *maxIter* is set to a sufficiently large value to make MTS adequately converge (i.e. further increasing the value of *maxIter* cannot improve the quality of Pareto solutions). The parameters of *reintMax*, *minTabu* and *maxTabu* are

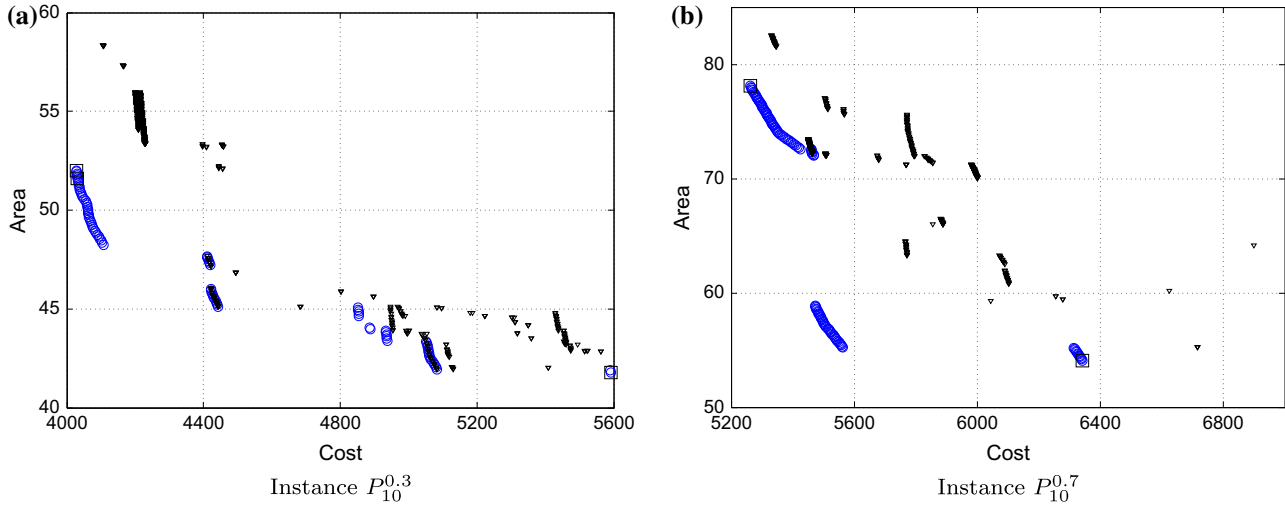


Figure 2. Pareto solutions found by MTS-HR, CPLEX, and NSGA-II for 10-machine problem instances.

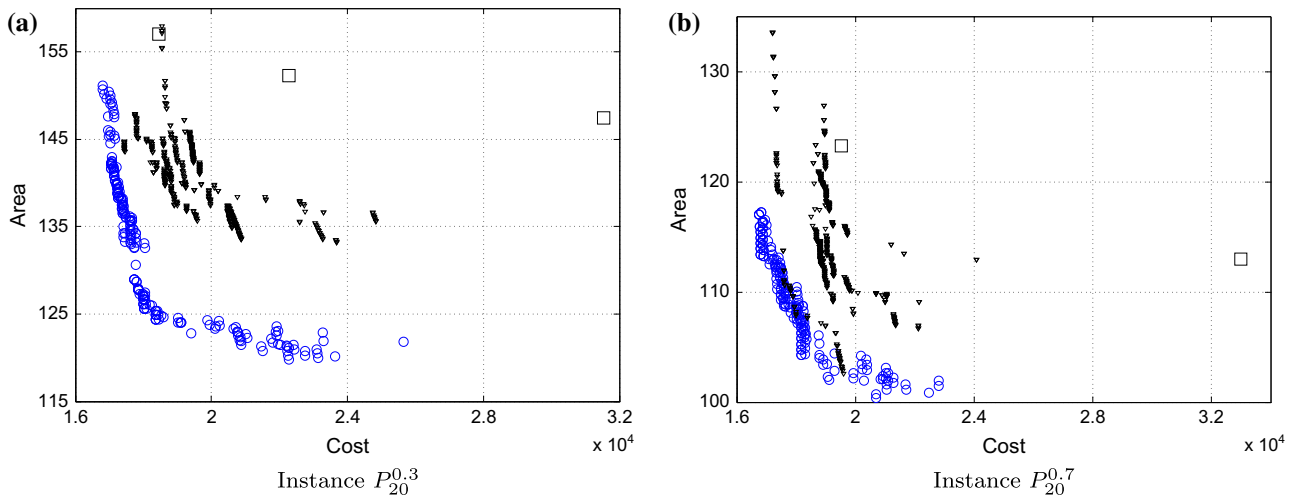


Figure 3. Pareto solutions found by MTS-HR, CPLEX, and NSGA-II for 20-machine problem instances.

set to be larger values for problem instances with more machines. In the experiments, we found that MTS-HR with different *reintMax* and ranges of tabu list length (*minTabu* and *maxTabu*) produce very similar Pareto solutions, indicating that the performance of MTS-HR is not particularly sensitive to small changes in *reintMax*, *minTabu* and *maxTabu* values. Hence, it is straightforward to choose appropriate values for *reintMax*, *minTabu* and *maxTabu* to make MTS-HR produce a set of high-quality Pareto solutions. However, we observe that the performance of MTS-HR may degrade when *reintMax* is given a very small value or *minTabu* and *maxTabu* are set to extremely large values.

4.2 Comparative algorithms

MTS-HR is compared against an exact approach (CPLEX) and non-dominated sorting genetic algorithm II (NSGA-II) (Deb et al. 2002). CPLEX is used to solve the formulation in Section 2, where the two objectives are combined linearly to form a single objective. The weight value of the cost (area) objective is set to $\alpha \in [0, 1](1 - \alpha)$. We let α increase from 0 to 1 by the step size of 0.1. For each α value, a solution can be produced by CPLEX. By this means, we can obtain a set of Pareto solutions. For problems with 20 and 30 machines, the runtime of CPLEX is restricted to 1 h for each α value because CPLEX cannot find the optimal solution within a reasonable time. CPLEX could not solve the 50-machine test problems, and is therefore not used to compare against MTS-HR.

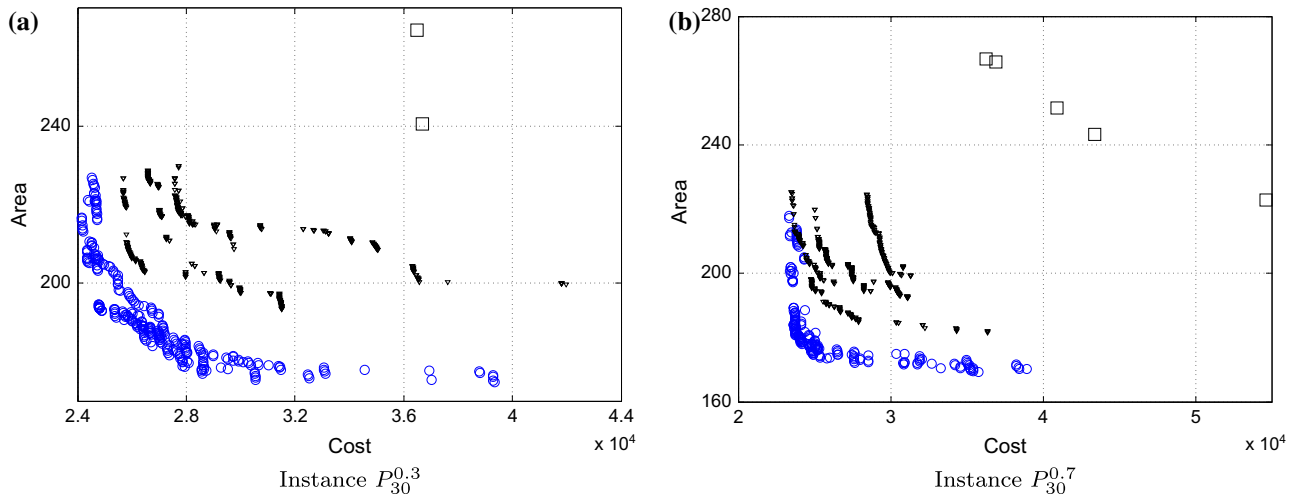


Figure 4. Pareto solutions found by MTS-HR, CPLEX, and NSGA-II for 30-machine problem instances.

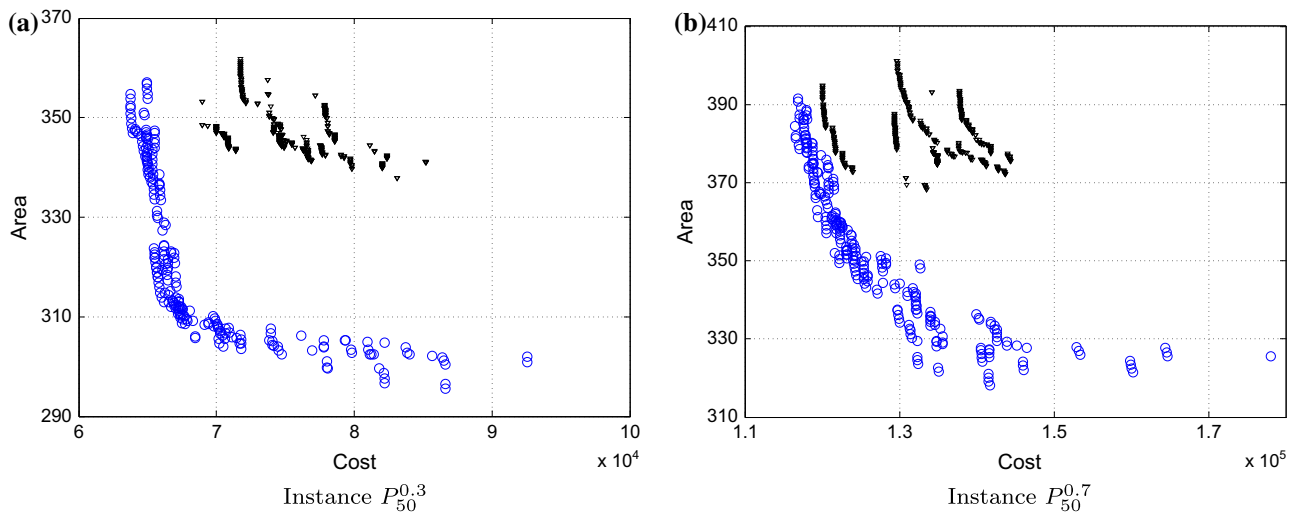


Figure 5. Pareto solutions found by MTS-HR and NSGA-II for 50-machine problem instances.

NSGA-II is a popular multi-objective meta-heuristic that has proved to be effective for both continuous and combinatorial multi-objective optimisation problems. As the EDRLP-SC has been first defined here, there exist no published works using NSGA-II to solve the EDRLP-SC. Therefore, we extend the NSGA-II for EDRLP in Zuo, Murray, and Smith (2014) to solve the EDRLP-SC. An integer vector is added into the solution coding in Zuo, Murray, and Smith (2014) to express the additional clearance. Its i th gene identifies the values of decision variables p_i^l and p_i^r . If the value of the i th gene is equal to 0, then $p_i^l = p_i^r = 1$. If the value is 1 (2), then $p_i^l = 1$ and $p_i^r = 0$ ($p_i^l = 0$ and $p_i^r = 1$). Single point crossover and random mutation are used for this integer vector. In accordance with parameter settings in Deb et al. (2002), we choose the population size as 100, crossover probability as 0.8, and mutation probability as $1/m$, where m is the number of machines. The number of generations is set to 30,000, 60,000, 90,000 and 150,000 for problem instances with 10, 20, 30 and 50 machines, respectively. We observe that a larger number of generations does not appear to improve the quality of solutions for these instances.

4.3 Experimental results

MTS-HR and NSGA-II are coded in C and are executed, along with CPLEX version 12.5.1.0, on an HP 8100 Elite desktop PC with a quad-core Intel i7-860 processor running Ubuntu Linux 10.10. Five independent runs of MTS-HR and NSGA-II are performed for each of the 20 problem instances.

Table 4. Comparison of MTS-HR, CPLEX and NSGA-II for convergence.

Instances	MTS-HR			CPLEX		NSGA-II		
	Avg iters	Avg time (s)	MID values	Time (s)	MID values	Avg iters	Avg time (s)	MID values
$P_{10}^{0.1}$	2,796	349	3,636.7	8,418	3,611.9	30,000	58	3,733.8
$P_{10}^{0.3}$	14,978	2,278	4,485.3	6,412	4,550.4	30,000	59	4,555.9
$P_{10}^{0.5}$	5,858	1,041	3,715.0	10,708	3,639.9	30,000	60	3,909.2
$P_{10}^{0.7}$	6,614	522	5,505.3	6,051	5,801.5	30,000	58	5,695.2
$P_{10}^{0.9}$	7,225	227	6,156.8	6,980	6,198.6	30,000	58	6,090.9
$P_{20}^{0.1}$	23,100	4,700	11,815.8	39,600	13,811.1	60,000	195	12,591.5
$P_{20}^{0.3}$	21,558	6,066	18,277.3	39,600	24,091.9	60,000	199	19,651.7
$P_{20}^{0.5}$	20,519	5,772	13,894.7	39,600	17,924.2	60,000	198	14,087.8
$P_{20}^{0.7}$	11,052	2,547	18,336.7	39,600	26,254.9	60,000	192	19,091.3
$P_{20}^{0.9}$	21,847	3,267	18,219.1	39,600	21,391.2	60,000	193	19,319.0
$P_{30}^{0.1}$	16,560	5,579	32,309.4	39,600	54,352.3	90,000	481	35,585.6
$P_{30}^{0.3}$	25,347	9,764	27,091.8	39,600	36,596.5	90,000	481	29,082.9
$P_{30}^{0.5}$	23,511	8,609	39,619.1	39,600	58,737.4	90,000	478	41,307.7
$P_{30}^{0.7}$	17,551	6,405	26,189.4	39,600	42,426.9	90,000	480	27,349.4
$P_{30}^{0.9}$	26,121	8,207	54,107.1	39,600	72,756.1	90,000	754	57,595.1
$P_{50}^{0.1}$	10,939	9,860	101,780.1	–	–	150,000	1,945	113,222.8
$P_{50}^{0.3}$	16,674	13,707	69,001.7	–	–	150,000	1,913	75,314.5
$P_{50}^{0.5}$	12,709	11,089	140,481.5	–	–	150,000	1,909	149,032.1
$P_{50}^{0.7}$	12,999	10,423	127,191.6	–	–	150,000	1,913	132,909.1
$P_{50}^{0.9}$	13,370	10,187	97,346.7	–	–	150,000	1,954	108,836.6

For the sake of brevity, we only show Pareto solutions of two samples among the five instances for each problem size. For $P_{10}^{0.3}$ and $P_{10}^{0.7}$, Figure 2 illustrates Pareto solutions obtained by five runs of MTS-HR, CPLEX and five runs of NSGA-II, which are represented by \circ , \square and ∇ , respectively. The MTS-HR finds the same solutions for each of those instances in its five runs. Pareto solutions found by CPLEX are identical to those of MTS-HR, indicating that MTS-HR can find the true Pareto front for small-size problems. Importantly, MTS-HR finds many Pareto solutions distributed along the Pareto front, while CPLEX can only find a few Pareto solutions on the front. MTS-HR is able to find much better Pareto solutions than NSGA-II, both in terms of solution quality and distribution. NSGA-II can only find a part of the Pareto front obtained by MTS-HR or cannot find the front at all. Each run of NSGA-II produces much different solutions, indicating that the performance of NSGA-II is not stable for this problem.

For $P_{20}^{0.3}$ and $P_{20}^{0.7}$ ($P_{30}^{0.3}$ and $P_{30}^{0.7}$), Pareto solutions obtained by five runs of MTS-HR, CPLEX and five runs of NSGA-II, denoted by \circ , \square and ∇ , respectively, are shown in Figures 3 and 4. Each run of MTS-HR is able to find a set of Pareto solutions which are much better than those found by NSGA-II and CPLEX. NSGA-II is able to find more and better Pareto solutions than CPLEX. The performance of NSGA-II is still not stable for these instances, as each of its runs produces different Pareto solutions. Pareto solutions produced by MTS-HR and NSGA-II for $P_{50}^{0.3}$ and $P_{50}^{0.7}$ are illustrated in Figure 5. Again, MTS-HR is able to find better Pareto solutions than NSGA-II for instances with 50 machines. In particular, NSGA-II can only find Pareto solutions in localised regions.

Table 4 provides mean ideal distances (MID) of Pareto solutions produced by MTS-HR, NSGA-II and CPLEX. The MID is a performance metric to measure the convergence rate of Pareto fronts to the ideal point, namely the mean distance between Pareto solutions and the point (0,0) (Asefi et al. 2014). Smaller MID values are indicative of better algorithm convergence. MTS-HR can obtain the smallest MID values for all problem instances except $P_{10}^{0.1}$, $P_{10}^{0.5}$ and $P_{10}^{0.9}$. We observe that CPLEX produces the smallest MID values for $P_{10}^{0.1}$ and $P_{10}^{0.5}$, while NSGA-II yields the smallest value for $P_{10}^{0.9}$. For $P_{10}^{0.1}$ and $P_{10}^{0.5}$, CPLEX can only produce a subset of Pareto solutions located on the Pareto front found by MTS-HR. Those solutions are not well distributed over the entirety of the Pareto front, but happen to be close to (0,0). As such, CPLEX earns smaller MID values than MTS-HR. Therefore, for $P_{10}^{0.1}$ and $P_{10}^{0.5}$, MTS-HR and CPLEX have the same performance in terms of

Table 5. Reduction of average cost and area values via sharing clearances.

Instances	No sharing clearances		Sharing clearances		Savings (%)	
	Avg cost	Avg area	Avg cost	Avg area	Avg cost	Avg area
$P_{10}^{0.1}$	4010.19	66.63	3647.49	62.62	9.04%	6.02%
$P_{10}^{0.3}$	5182.45	48.12	4796.48	44.68	7.45%	7.15%
$P_{10}^{0.5}$	4194.19	67.58	3798.74	63.39	9.43%	6.20%
$P_{10}^{0.7}$	6142.73	71.39	5579.57	64.93	9.17%	9.06%
$P_{10}^{0.9}$	7039.47	61.22	6277.31	56.59	10.83%	7.56%
$P_{20}^{0.1}$	13044.20	140.58	12037.09	131.23	7.72%	6.65%
$P_{20}^{0.3}$	20265.19	140.91	18727.14	130.94	7.59%	7.08%
$P_{20}^{0.5}$	15210.16	111.31	13953.77	103.07	8.26%	7.40%
$P_{20}^{0.7}$	19926.61	116.61	18292.00	108.02	8.20%	7.36%
$P_{20}^{0.9}$	20328.93	155.44	18453.31	141.22	9.23%	9.14%
$P_{30}^{0.1}$	35651.21	181.51	33119.92	169.38	7.10%	6.69%
$P_{30}^{0.3}$	29729.75	206.34	27473.17	190.71	7.59%	7.57%
$P_{30}^{0.5}$	43787.07	201.99	40577.33	187.43	7.33%	7.21%
$P_{30}^{0.7}$	28941.68	197.93	26621.85	181.83	8.02%	8.13%
$P_{30}^{0.9}$	60822.54	215.81	54946.96	195.03	9.66%	9.63%
$P_{50}^{0.1}$	111370.56	308.02	104046.85	288.55	6.58%	6.32%
$P_{50}^{0.3}$	77203.33	339.96	71527.97	314.65	7.35%	7.45%
$P_{50}^{0.5}$	153400.02	363.49	141751.90	334.65	7.59%	7.93%
$P_{50}^{0.7}$	139806.54	389.18	127107.16	354.00	9.08%	9.04%
$P_{50}^{0.9}$	107887.49	349.32	97816.44	316.96	9.33%	9.26%

solution convergence while MTS-HR has better performance than CPLEX in terms of solution distribution. Similar logic can explain why NSGA-II nets smaller MID values than MTS-HR in $P_{10}^{0.9}$. For problem instances with 10 machines, NSGA-II and CPLEX obtain similar MID values. For instances with 30 machines, NSGA-II is able to obtain much smaller MID values than CPLEX, indicating that NSGA-II outperforms CPLEX in terms of solution convergence for large-size instances.

Table 4 also presents the mean number of iterations and runtime of the three approaches. MTS-HR takes about 3–20 min to find the Pareto fronts for $P_{10}^{0.1}$, $P_{10}^{0.5}$, $P_{10}^{0.7}$ and $P_{10}^{0.9}$ and about 40 min for $P_{10}^{0.3}$. CPLEX takes 1.5–3 h to obtain several Pareto solutions for $P_{10}^{0.1}$ – $P_{10}^{0.9}$. MTS-HR takes about 40–100 min for each instance with 20 machines and an average of 1.5 to 3 h for each instance with 30 machines, while the runtime of CPLEX is 11 h for each. While the runtime of NSGA-II is shorter than MTS-HR, NSGA-II cannot escape from the local optima, even if allowed longer runtimes.

Section 2 shows that separately considering minimum clearance and additional clearance can result in a layout with a smaller cost and area because any two adjacent machines have the possibility to share their additional clearances. To quantitatively investigate the cost and area reduction due to clearance sharing, we construct a layout without sharing clearance for each machine sequence in the set *Archive*. To do so, decision variables p_i^l (p_i^r) must be determined for each machine i to decide to which side of machine i additional clearances a_i^l (a_i^r) should be applied. Subsequently, the two types of clearances are considered as a single clearance. If $b_i = 1$, we let both p_i^l and p_i^r be 1; otherwise, let p_i^l (p_i^r) be 1 according to a 50% probability (in practice the WIP can be stored on either side of a machine). The layout without sharing clearances is constructed by separating each pair of adjacent machines i and j with a single clearance ($p_i^l a_i^l + c_{ij} + p_j^r a_j^r$).

For each problem instance, Table 5 presents the average cost and area values of the layouts without sharing clearances for all machine sequences in the *Archive* set. As stated in Section 3.1.1, for each machine sequence in *Archive*, MP is used to determine p_i^l and p_i^r to make adjacent machines share their additional clearance to achieve the minimum cost and area values. For each problem instance, Table 5 gives the average cost and area values produced by MP for all machine sequences in *Archive* sets. From Table 5 we see that allowing shared clearances can reduce the average material handling cost by 6.58%–10.83%, and can reduce the area consumed by 6.02%–9.63%. As the number of machines requiring additional

clearance on both sides (i.e. where $b_i = 1$) increases, the cost and area consumption tend to decrease, as more additional clearances may be shared.

5. Conclusions

In this paper, the EDRLP with shared clearances is proposed and formulated as a mixed integer linear program, in which the objective is to determine the exact location and additional clearance of each machine while simultaneously minimising both total material flow cost and layout area. This problem arises in production environments such as microelectronics where the floor space is highly expensive. For this type of problem, considering shared clearances between machines is very effective in reducing the layout area and decreasing material flow cost.

A MTS combined with HR is proposed to effectively solve this problem. The MTS is used to find the non-dominated sequences of machines and the HR are used to determine the additional clearances of machines. Computational experience shows that MTS-HR consistently and significantly improves upon the comparative approaches and can find high-quality non-dominated solutions that cover the breadth of the Pareto front.

There are several promising future research directions related to this work. One such avenue is to extend the problem setting to larger production environments, perhaps involving more than two rows of machines. The continuous location formulation of Montreuil (1991) may be beneficial to such an effort. Additionally, the establishment of tight lower bounds for the EDRLP-SC would be useful for better quantifying the effectiveness of the MTS-HR heuristic, or for other future solution approaches. The work of Meller, Narayanan, and Vance (1998) and Sherali, Fraticelli, and Meller (2003), in the context of general facility layout problems, may serve as a foundation for such study. Finally, more constrained methods of transport, such as one-way material flow, might be incorporated into the EDRLP-SC.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This paper was partly supported by the National Natural Science Foundation of China [grant number 61374204].

References

- Ahonen, H., A. G. de Alvarenga, and A. R. S. Amaral. 2014. "Simulated Annealing and Tabu Search Approaches for the Corridor Allocation Problem." *European Journal of Operational Research* 232 (1): 221–233.
- Amaral, A. R. S. 2006. "On the Exact Solution of a Facility Layout Problem." *European Journal of Operational Research* 173 (2): 508–518.
- Amaral, A. R. S. 2008. "An Exact Approach to the One-dimensional Facility Layout Problem." *Operations Research* 56 (4): 1026–1033.
- Amaral, A. R. S. 2009. "A New Lower Bound for the Single Row Facility Layout Problem." *Discrete Applied Mathematics* 157 (1): 183–190.
- Amaral, A. R. S. 2012. "The Corridor Location Problem." *Computers & Operations Research* 39: 3325–3330.
- Amaral, A. R. S. 2013a. "Optimal Solutions for the Double Row Layout Problem." *Optimization Letters* 7 (2): 407–413.
- Amaral, A. R. S. 2013b. "A Parallel Ordering Problem in Facilities Layout." *Computers & Operations Research* 40 (12): 2930–2939.
- Anjos, M. F., and A. Vannelli. 2008. "Computing Globally Optimal Solutions for Single-row Layout Problems Using Semidefinite Programming and Cutting Planes." *INFORMS Journal on Computing* 20 (4): 611–617.
- Asefi, H., F. Jolai, M. Rabiee, and M. E. T. Araghi. 2014. "A Hybrid NSGA-II and VNS for Solving a bi-objective No-wait Flexible Flowshop Scheduling Problem." *International Journal of Advanced Manufacturing Technology* 75: 1017–1033.
- Castillo, I., and B. A. Peters. 2004. "Integrating Design and Production Planning Considerations in Multi-bay Manufacturing Facility Layout." *European Journal of Operational Research* 157: 671–687.
- Chiang, W. C., P. Kouvelis, and T. L. Urban. 2006. "Single- and Multi-objective Facility Layout with Workflow Interference Considerations." *European Journal of Operational Research* 174: 1414–1426.
- Chung, J., and J. M. A. Tanchoco. 2010. "The Double Row Layout Problem." *International Journal of Production Research* 48 (3): 709–727.
- Datta, D., A. R. S. Amaral, and J. R. Figueira. 2011. "Single Row Facility Layout Problem Using a Permutation-based Genetic Algorithm." *European Journal of Operational Research* 213 (2): 388–394.
- Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197.
- Djellab, H., and A. Gourgand. 2001. "A New Heuristic Procedure for the Single Row Facility Layout Problem." *International Journal of Computer Integrated Manufacturing* 14 (3): 270–280.

- Ficko, M., M. Brezocnik, and J. Balic. 2004. "Designing the Layout of Single-and Multiple-rows Flexible Manufacturing System by Genetic Algorithms." *Journal of Materials Processing Technology* 157: 150–158.
- Gen, M., K. Ida, and C. Cheng. 1995. "Multirow Machine Layout Problem in Fuzzy Environment Using Genetic Algorithms." *Computers & Industrial Engineering* 29 (1–4): 519–523.
- Heragu, S., and A. Alfa. 1992. "Experimental Analysis of Simulated Annealing Based Algorithms for the Layout Problem." *European Journal of Operational Research* 57 (2): 190–202.
- Heragu, S., and A. Kusiak. 1988. "Machine Layout Problem in Flexible Manufacturing Systems." *Operations Research* 36 (2): 258–268.
- Hungerländer, P., and F. Rendl. 2013. "A Computational Study and Survey of Methods for the Single-row Facility Layout Problem." *Computational Optimization and Applications* 55 (1): 1–20.
- Jaramillo, J. R., and A. R. McKendall. 2010. "The Generalized Machine Layout Problem." *International Journal of Production Research* 48: 4848–4859.
- Kothari, R., and D. Ghosh. 2013. "Tabu Search for the Single Row Facility Layout Problem Using Exhaustive 2-opt and Insertion Neighborhoods." *European Journal of Operational Research* 224: 93–100.
- Kulturel-Konak, S., A. E. Smith, and B. A. Norman. 2006. "Multi-objective Tabu Search Using a Multinomial Probability Mass Function." *European Journal of Operational Research* 169 (3): 918–931.
- Meller, R. D., V. Narayanan, and P. H. Vance. 1998. "Optimal Facility Layout Design." *Operations Research Letters* 23 (3): 117–127.
- Montreuil, B. 1991. "A Modelling Framework for Integrating Layout Design and Flow Network Design." *Material Handling '90*, Progress in Material Handling and Logistics, edited by R. J. Graves, M. R. Wilhelm, L. F. McGinnis and R. E. Ward. Vol. 2, 95–115. Hebron, KY: Springer.
- Murray, C. C., A. E. Smith, and Z. Q. Zhang. 2013. "An efficient local search heuristic for the double row layout problem with asymmetric material flow." *International Journal of Production Research* 51 (20): 6129–6139.
- Murray, C. C., X. Q. Zuo, and A. E. Smith. 2012. "An Extended Double Row Layout Problem." In *12th International Material Handling Research Colloquium, Gardanne, France*, edited by B. Montreuil, A. Carrano, M.M.R. de Kostner, K.R. Gue, M. Ogle, and J. Smith, Material Handling Institute, 554–569.
- Ozcelik, F. 2012. "A Hybrid Genetic Algorithm for the Single Row Layout Problem." *International Journal of Production Research* 50 (20): 5872–5886.
- Sadrzadeh, A. 2012. "A Genetic Algorithm with the Heuristic Procedure to Solve the Multi-line Layout Problem." *Computers & Industrial Engineering* 62 (4): 1055–1064.
- Samarghandi, H., and K. Eshghi. 2010. "An Efficient Tabu Algorithm for the Single Row Facility Layout Problem." *European Journal of Operational Research* 205 (1): 98–105.
- Samarghandi, H., P. Taabayan, and F. F. Jahantigh. 2010. "A Particle Swarm Optimization for the Single Row Facility Layout Problem." *Computers & Industrial Engineering* 58 (4): 529–534.
- Sherali, H. D., B. M. P. Fraticelli, and R. D. Meller. 2003. "Enhanced Model Formulations for Optimal Facility Layout." *Operations Research* 51 (4): 629–644.
- Singh, S. P., and R. R. K. Sharma. 2008. "Two-level Modified Simulated Annealing Based Approach for Solving Facility Layout Problem." *International Journal of Production Research* 46 (13): 3563–3582.
- Solimanpur, M., P. Vrat, and R. Shankar. 2005. "An Ant Alogorithm for the Single Row Layout Problem in Flexible Manufacturing Systems." *Computers & Operations Research* 32 (3): 583–598.
- Wang, S. L., X. Q. Zuo, X. Q. Liu, X. C. Zhao, and J. Q. Li. 2015. "Solving Dynamic Double Row Layout Problem via Combining Simulated Annealing and Mathematical Programming." *Applied Soft Computing* 37: 303–310.
- Zhang, Z., and C. C. Murray. 2012. "A Corrected Formulation for the Double Row Layout Problem." *International Journal of Production Research* 15 (50): 4220–4223.
- Zuo, X. Q., C. C. Murray, and A. E. Smith. 2014. "Solving an Extended Double Row Layout Problem Using Multi-objective Tabu Search and Linear Programming." *IEEE Transactions on Automation Science and Engineering* 11 (4): 1122–1132.