International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tprs20

An efficient local search heuristic for the double row layout problem with asymmetric material flow

Chase C. Murray [a], Alice E. Smith [a] & Zeqiang Zhang [b]

[a] Auburn University, Industrial & Systems Engineering, Auburn, AL, USA

[b] Southwest Jiaotong University, School of Mechanical Engineering, Chengdu, China

Published online: 10 Jul 2013.

To cite this article: International Journal of Production Research (2013): An efficient local search heuristic for the double row layout problem with asymmetric material flow, International Journal of Production Research, DOI: 10.1080/00207543.2013.803168

To link to this article: http://dx.doi.org/10.1080/00207543.2013.803168

International Journal of Production Research, 2013
http://dx.doi.org/10.1080/00207543.2013.803168

An efficient local search heuristic for the double row layout problem with asymmetric material flow

Chase C. Murray[a*], Alice E. Smith[a] and Zeqiang Zhang[b]

[a]*Industrial & Systems Engineering, Auburn University, Auburn, AL, USA;* [b]*School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China*

(*Received 3 November 2011; final version received 30 April 2013*)

The double row layout problem (DRLP) consists of arranging a number of rectangular machines of varying widths on either side of a corridor to minimize the total cost of material handling for products that move between these machines. This problem arises in the context of many production environments, most notably semiconductor manufacturing. Because the DRLP contains both combinatorial and continuous aspects, traditional solution approaches are not well suited to obtain solutions within a reasonable time. Moreover, previous approaches to this problem did not consider asymmetric flows. In this paper, an effective local search procedure featuring linear programming is proposed for solving the DRLP with asymmetric flows (symmetric flows being a special case). This approach is compared against several constructive heuristics and solutions obtained by a commercial mixed integer linear programming solver to evaluate its performance. Computational results show that the proposed heuristic is an effective approach, both in terms of solution quality and computational effort.

Keywords: facility layout; double row layout problem; local search; linear programming; mixed integer programming

1. Introduction

The double row layout problem (DRLP) consists of arranging a number of rectangular machines of varying widths on either side of a corridor to minimise the total cost of material handling. An automated guided vehicle (AGV) system operates along the aisle to move material from one machine to another. Feasible layouts must observe a predefined minimum clearance between adjacent machines, such that each pair of machines may require a unique amount of separation. The clearance takes into account machine access, heat dissipation and work in progress (WIP) storage. This problem, as illustrated in Figure 1, was first discussed in the literature by Chung and Tanchoco (2010).

The DRLP is related to two other facility layout problems – the single-row layout problem (SRLP) and the multiple-row layout problem (MRLP). In the SRLP, machines are placed on only one side of the aisle and are separated by their minimum clearances, with the objective of minimising the total cost of material flow (c.f. Heragu and Kusiak 1988; Solimanpur, Vrat, and Shankar 2005). This makes the SRLP a purely combinatorial problem, as only the sequence of machines in the single row must be determined. Although the MRLP involves the determination of machine layouts in numerous rows, it is similarly combinatorial in nature due to the assumption that all machines are always separated by their minimum clearances. Unlike the SRLP and MRLP, the DRLP requires the determination of not only the sequence of machines in each row (relative placement), but also the actual location of each machine (absolute placement). Thus, the DRLP is a more complex problem, as it incorporates both combinatorial and continuous aspects.

Because the SRLP and MRLP do not include continuous aspects, solution approaches for these problems are not directly applicable to the DRLP. For problems of practical size, the use of commercial mixed integer programming (MIP) solvers alone is insufficient to obtain solutions to the DRLP in a reasonable time. In light of this computational burden, Chung and Tanchoco (2010) proposed five heuristic approaches to solve the DRLP. However, the empirical evidence supporting the effectiveness of these heuristics was called into question due to errors in their proposed model. Specifically, these errors admit layouts that do not observe the minimum-clearance restrictions. Although a corrected MIP formulation for the DRLP was provided by Zhang and Murray (2011), no corresponding solution approaches were proposed. Furthermore, both the original formulation of Chung and Tanchoco (2010) and the corrected formulation of Zhang and Murray (2011) assume symmetric material flow between machines. As a result, there are currently no valid solution approaches for the DRLP with asymmetric material flow.

*Corresponding author. Email: cmurray@auburn.edu

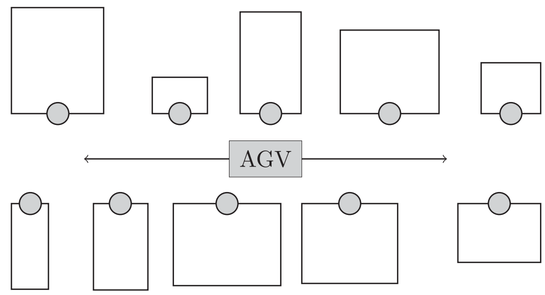

Figure 1. The DRLP (adapted from Chung and Tanchoco (2010)).

This paper makes the following contributions. First, it extends the corrected formulation of Zhang and Murray by considering asymmetric flow of material between machines, thus making the model more realistic in practice. Second, the five constructive heuristics of Chung and Tanchoco, as well as three modified constructive heuristics, are evaluated for asymmetric problems. Finally, a tractable methodology for solving the DRLP is devised and tested. The approach, which combines combinatorial optimisation using local search with continuous optimisation using LP, provides surprisingly good performance from a straightforward heuristic.

The remainder of this paper is organised as follows. Section 2 provides a review of the literature related to the DRLP. A mathematical formulation for the DRLP with asymmetric material flow is presented in Section 3. Multiple solution approaches are described in Section 4, including three variants of Chung and Tanchoco's constructive heuristics and the proposed local search heuristic. Numerical results from 200 problem instances are evaluated in Section 5, while a summary is provided in Section 6.

2. Literature review

Although research on general facility layout problems is abundant, to the best of our knowledge, only two published works are devoted to the DRLP. This includes the research of Chung and Tanchoco (2010), who proposed the first mathematical formulation of the DRLP with symmetric material flow, and the technical note of Zhang and Murray (2011), who provide a corrected MIP formulation of the DRLP.

Chung and Tanchoco (2010) first proposed the DRLP and formulated it as a MIP problem for which only small instances may be solved optimally by integer programming software. For example, the authors noted that a 10-machine problem required approximately two hours to solve optimally via CPLEX. Thus, five constructive heuristics employing different priority rules were proposed: minimum location cost first (MinLCF) minimum flow first (MinFF), maximum flow first (MaxFF), minimum width first (MinWF), and maximum width first (MaxWF). These heuristics were reported to produce solutions with large optimality gaps. For example, the best performing heuristic approach for 6-machine test problems found solutions with an optimality gap of 26.3%. Although for larger problems the heuristics demonstrated performance on par with the best (albeit not provably optimal) solution obtained by CPLEX within a 10-min time limit, the authors acknowledge that more effective solution methods are required.

Zhang and Murray (2011) discovered that the MIP model by Chung and Tanchoco (2010) allows layouts that ignore the minimum clearance requirements between some adjacent machines. Because the five heuristic approaches proposed by Chung and Tanchoco rely upon solutions from the MIP, the validity of their numerical analysis is in question. A corrected formulation for the DRLP was provided by Zhang and Murray (2011), although no solution approaches were proposed. Therefore, no proven solution approaches for the DRLP appear in the literature.

In the absence of proven solution approaches designed specifically for the DRLP, we briefly review solution approaches for related facility layout problems – the MRLP, the SRLP, and, to a lesser extent, general facility layout problems for which machines are not typically required to be separated by straight aisles. While solution approaches to these problems may not be applied directly to the DRLP, modified versions may be useful, as noted by Chung and Tanchoco (2010).

The MRLP, in which tools in multiple rows of machines are separated by the machines' minimum allowable clearances, is commonly formulated as a quadratic assignment problem (QAP). QAP formulations, such as those provided by Singh and Sharma (2008), assume predefined possible machine locations and therefore do not address the continuous nature of the problem at hand. Additionally, as the MRLP does not provide for aisles between rows of machines, some formulations do not enforce straight rows. For example, Hassan (1994) and Tate and Smith (1995) proposed QAP approaches for general facility layout problems in which all facilities are of equal area and candidate locations are provided in advance. Heragu and Kusiak (1988) formulated single- and multiple-row layout problems as QAPs and presented two construction algorithms. Heragu and Alfa (1992) further proposed a hybrid simulated annealing algorithm for the MRLP with facilities of equal area. Later, Gen, Ida, and Cheng (1995) presented genetic algorithms to solve the MRLP with unequal-sized machines in a fuzzy environment. Heragu (1992) presented formulations for single- and multiple-row layout problems which employ absolute value expressions in the objective function and constraints, as well as an additional nonlinear constraint. Layouts resulting from these models do not necessarily contain straight aisles separating rows of machines. More recently, Ficko, Brezocnik, and Balic (2004) proposed a genetic algorithm to create permutations of machines, separated by their minimum clearance requirements, in a multiple-row layout. An aisle of predetermined width separates each row of machines. The genetic algorithm is evaluated on a single test problem, making the effectiveness of this solution approach unclear.

Numerous heuristic and meta-heuristic approaches have been proposed for the SRLP, a good summary of which is presented in Chung and Tanchoco (2010). Here, we focus on the latest solution approaches that were not described therein. Amaral (2008) proposed an efficient exact approach to the SRLP, based on an equivalent mixed 0–1 linear program. Anjos and Vannelli (2008) presented the combination of a semi-definite programming relaxation with cutting planes to compute globally optimal layouts for the SRLP. Meta-heuristic approaches, such as simulated annealing (Heragu and Alfa 1992; Braglia 1996), tabu search (Samarghandi and Eshghi 2010), genetic algorithms (Datta, Amaral, and Figueira 2011), particle swarm optimization (Samarghandi, Taabayan, and Jahantigh 2010), ant colony optimization (Solimanpur, Vrat, and Shankar 2005), and hybrid ant colony/particle swarm optimization (Teo and Ponnambalam 2008) provide alternatives to traditional optimization techniques which may be more practical for large-scale problems. Ant colony optimization has also been successfully applied to several facility layout problems outside the scope of the SRLP, including those formulated as QAP (Pour and Nosraty 2006; See and Wong 2008), the dynamic facility layout problem (McKendall and Shang 2006), and unequal area facility layout problems (Komarudin and Wong 2010; Wong and Komarudin 2010; Kulturel-Konak and Konak 2011).

While research on more general facility layout problems does share some commonalities with our work, these problems typically do not address all aspects of the DRLP. For example, Peters and Yang (1997) and Yang and Peters (1997) studied problems in the context of semiconductor layout and material handling configurations in which tools are assigned to equal sized rectangular blocks, thus making the precise determination of each tool's location difficult. Montreuil, Brotherton, and Marcotte (2002) address a facility layout problem containing both combinatorial and continuous aspects, where machine centres are assigned to predefined zones and load/unload points for each centre are determined around the boundary of each zone. The model is solved via a commercial integer programming solver, and computational results show the model's effectiveness for problems containing up to 12 machine centres. Subsequently, Montreuil et al. (2004) solve this problem via ant colony optimization to determine the assignment of centres to zones, followed by the solution to a linear programming problem to determine optimal locations of the load/unload points. Results are demonstrated on a 10-machine centre problem.

A similar problem, the continuous facility layout problem, was addressed by Xie and Sahinidis (2008). A branch-and-bound solution approach was tested on problems with up to 12 machines. Kouvelis, Chiang, and Yu (1995) consider the row layout problem (RLP), which addresses both single and double row layouts with the objective of minimizing material handling backtracking. This problem does not consider clearance restrictions explicitly, and involves the assignment of machines to discrete candidate locations in the layout. A dynamic programming approach is applied to determine optimal machine sequences.

3. Mathematical model

In this section, we provide the MIP formulation for the DRLP with asymmetric material flow. Consistent with Chung and Tanchoco (2010), and the subsequent corrected model of Zhang and Murray (2011), we make the following two assumptions. First, the width of the aisle separating the two rows is considered to be zero. Second, the material is assumed to be loaded and unloaded at a point in the middle of the width of each machine. However, unlike the aforementioned models, we do not require material flow to be symmetric. This provides for a more realistic model, as material flows within a production facility are typically asymmetric, and this relaxation requires only minor modifications to the model of Zhang and Murray.

3.1 *Notation*

For ease of explanation, we adopt the notation of Zhang and Murray (2011) wherever practical, which was itself derived from the original formulation of Chung and Tanchoco (2010). The model parameters are defined as follows:

m Number of machines.
I Set of machines, $I = \{1, 2, \dots, m\}$.
i, j Machine indices, $i, j \in I$.
r Row index, $r \in R$, where set $R = \{U, L\}$ specifies the upper and lower rows.
w_i Width of machine $i \in I$.
f_{ij} Flow frequency times unit cost from machine $i \in I$ to machine $j \in \{I \setminus i\}$.
a_{ij} Minimum clearance required between machines i and j for all $i \in I^1$ and $j \in I^2$, where $I^1 = \{1, 2, \dots, m-1\}$ and $I^2 = \{i+1, i+2, \dots, m\}$.

An important, albeit subtle, modification from the notation of the previously published DRLP models lies in the definition of f_{ij}, which is no longer assumed to be upper triangular. This change allows the incorporation of asymmetric material flow.

The required decision variable definitions are given by:

x_{ir} Continuous variable representing the location of machine $i \in I$ in row $r \in R$, such that $x_{ir} = 0$ if i is not assigned to row r.
y_{ir} Binary variable, defined for all $i \in I$ and $r \in R$, where $y_{ir} = 1$, if machine i is assigned to row r ($y_{ir} = 0$ otherwise).
z_{rij} Binary variable, defined for all $r \in R$, $i \in I$, and $j \in \{I \setminus i\}$, such that $z_{rij} = 1$ if machines i and j are placed in row r and j is located to the right of i.

3.2 *Mathematical formulation of the DRLP*

The DRLP with asymmetric material flow is presented as the following mixed integer linear programming problem, Formulation T (for total).

$$\text{(T)} \quad \text{Min} \quad Z = \sum_{i \in I^1} \sum_{j \in I^2} \left(f_{ij} + f_{ji} \right) \left(v_{ij}^+ + v_{ij}^- \right) \tag{1}$$

$$\text{s.t.} \qquad x_{ir} \le M y_{ir} \qquad \forall\, i \in I, r \in R, \tag{2}$$

$$\sum_{r \in R} y_{ir} = 1 \qquad \forall\, i \in I, \tag{3}$$

$$\frac{w_i y_{ir} + w_j y_{jr}}{2} + a_{ij} z_{rji} \le x_{ir} - x_{jr} + M(1 - z_{rji}) \qquad \forall\, i \in I^1, j \in I^2, r \in R, \tag{4}$$

$$\frac{w_i y_{ir} + w_j y_{jr}}{2} + a_{ij} z_{rij} \le x_{jr} - x_{ir} + M(1 - z_{rij}) \qquad \forall\, i \in I^1, j \in I^2, r \in R, \tag{5}$$

$$\sum_{r \in R} x_{ir} - \sum_{r \in R} x_{jr} = v_{ij}^- - v_{ij}^+ \qquad \forall\, i \in I^1, j \in I^2, \tag{6}$$

$$z_{rij} + z_{rji} \le \frac{y_{ir} + y_{jr}}{2} \qquad \forall\, i \in I^1, j \in I^2, r \in R, \tag{7}$$

$$z_{rij} + z_{rji} + 1 \ge y_{ir} + y_{jr} \qquad \forall\, i \in I^1, j \in I^2, r \in R, \tag{8}$$

$$x_{ir} \ge 0 \qquad \forall\, i \in I, r \in R, \tag{9}$$

$$v_{ij}^+, v_{ij}^- \ge 0 \qquad \forall\, i \in I^1, j \in I^2, \tag{10}$$

$$y_{ir} \in \{0, 1\} \qquad \forall\, i \in I, r \in R, \tag{11}$$

$$z_{rij} \in \{0, 1\} \qquad \forall\, i \in I, j \in \{I \setminus i\}, r \in R. \tag{12}$$

In this model, M represents a 'large' number, and is calculated as

$$M = \sum_{i \in I} \left\{ w_i + \max_{j \in \{I \setminus i\}} (a_{ij}) \right\}.$$

The objective function (1) seeks to minimise the total cost of material handling, where $(v_{ij}^+ + v_{ij}^-)$ represents the absolute value of the distance between machines $i \in I^1$ and $j \in I^2$. Such representation of the distance between machines was proposed by Chung and Tanchoco (2010). The costs associated with asymmetric material flow are incorporated by the inclusion of the $(f_{ij} + f_{ji})$ term, such that $f_{ij}(v_{ij}^+ + v_{ij}^-)$ and $f_{ji}(v_{ij}^+ + v_{ij}^-)$ represent the flow cost from machine i to machine j, and from machine j to machine i, respectively. For symmetric flow problems, these two terms would be equal. Together, Constraints (2) and (3) guarantee that each machine will be placed in exactly one location in exactly one row. Constraints (4) and (5) ensure that the minimum clearance requirements among machines in the same row are satisfied, while the absolute distance between machines is governed by Constraint (6). Constraints (7) and (8) establish the relationship between binary decision variables z_{rij} and y_{ir}, such that when $y_{ir} = y_{jr} = 1$ (i.e. machines i and j are both assigned to row r), either z_{rij} or z_{rji} should be equal to 1; otherwise, $z_{rij} = z_{rji} = 0$.

4. Solution approaches

The complex combinatorial and continuous optimisation problem represented in Formulation T is very difficult to solve, as the DRLP is an NP-hard problem (Chung and Tanchoco 2010). Therefore, Chung and Tanchoco proposed five constructive heuristics for the DRLP with symmetric material flow. These heuristics begin with an empty layout and select two machines to be placed on opposite rows. An iterative procedure, guided by one of five rules – MinLCF (minimum location cost first), MinFF (minimum flow first), MaxFF, MinWF (minimum width first) and MaxWF – chooses one machine at a time to be placed in the layout. This machine may be inserted in any position of the partially constructed layout, where the choice of a position determines a sequence of machines in each row. Given these two sequences, it is possible to directly ascertain the required values of the binary y_{ir} and z_{rij} decision variables. Therefore, for a particular ordering of machines, Formulation T may be solved as a linear program (LP). The solution to the resulting LP provides the absolute location of each machine (in continuous space). Each heuristic chooses the relative location for the inserted machine that results in the minimum objective function value. The procedure ends when all machines have been added to the layout.

The results from Chung and Tanchoco's numerical analysis did not indicate that any of the five constructive heuristics were particularly effective, although this may have been due to the aforementioned issues with their model. Furthermore, these constructive heuristics have not been applied to a DRLP with asymmetric material flow. Therefore, we propose three alternative rules for selecting machines within the framework provided by Chung and Tanchoco to accommodate asymmetric material flow. In the remainder of this section, we first provide a detailed description of the resulting constructive heuristics. Afterwards, we propose an effective local search procedure that seeks to improve upon initial solutions obtained via one of these constructive heuristics.

4.1 *Modified constructive heuristic rules*

We begin by defining the notation required by the constructive heuristic. Let I^a represents the unordered set of assigned machines (initially an empty set). Similarly, I^u represents the unordered set of unassigned machines, such that $I = I^a \cup I^u$. We define $\boldsymbol{\pi}_U \subseteq I^a$ ($\boldsymbol{\pi}_L \subseteq I^a$) to represent the ordered sequence of machines placed in the upper (lower) row, where $\pi_{U,p}$ is the pth machine in the upper row ($p = 1, \ldots, |\boldsymbol{\pi}_U|$) and $\pi_{L,q}$ is the qth machine in the lower row ($q = 1, \ldots, |\boldsymbol{\pi}_L|$). These sets describe the relative locations of machines in each row, but do not reflect the absolute position of any machine. Given these sequences, it is straightforward to determine the necessary values of the y_{ir} and z_{rij} decision variables. As Chung and Tanchoco (2010) observed, when these decision variable values are set, the determination of the corresponding optimal absolute location of each machine requires solving only a linear (rather than integer linear) program. This LP, given by Formulation C (for continuous), is as follows:

$$\text{(C)} \quad \text{Min} \quad Z^{LP}(\boldsymbol{\pi}_U, \boldsymbol{\pi}_L) = \sum_{i \in I^{1'}} \sum_{j \in I^{2'}} \left(f_{ij} + f_{ji}\right) \left(v_{ij}^+ + v_{ij}^-\right) \tag{13}$$

$$\text{s.t.} \quad x_{ir} = 0 \quad \forall \{i \in I', r \in R : y_{ir} = 0\}, \tag{14}$$

$$\frac{w_i + w_j}{2} + a_{ij} \le x_{ir} - x_{jr} \quad \forall \{i \in I^{1'}, j \in I^{2'}, r \in R : z_{rji} = 1\}, \tag{15}$$

$$\frac{w_i + w_j}{2} + a_{ij} \le x_{jr} - x_{ir} \quad \forall \{i \in I^{1'}, j \in I^{2'}, r \in R : z_{rij} = 1\}, \tag{16}$$

$$\textstyle\sum_{r \in R} x_{ir} - \sum_{r \in R} x_{jr} = v_{ij}^- - v_{ij}^+ \quad \forall\, i \in I^{1'}, j \in I^{2'}, \tag{17}$$

$$x_{ir} \ge 0 \quad \forall\, i \in I', r \in R, \tag{18}$$

$$v_{ij}^+, v_{ij}^- \ge 0 \quad \forall\, i \in I^{1'}, j \in I^{2'}. \tag{19}$$

Because this is a constructive heuristic, some machines will be unassigned until the last iteration. For the sake of completeness, we let $I' = \{\boldsymbol{\pi}_U \cup \boldsymbol{\pi}_L\}$ represent the set of machines that are currently assigned. Similarly, let $I^{1'} = \{I' \setminus \max(I')\}$ and $I^{2'} = \{j \in I' : j > i\}$ for all $i \in I^{1'}$. Note that $z_{rij} = z_{rji} = 0$ unless both i and j are assigned to row r. Also, if both i and j are assigned to row r, then $z_{rij} + z_{rji} = 1$ (i.e. i is either to the left or right of j).

The following constructive heuristic contains two slight differences from Chung and Tanchoco's framework. First, the selection of the first pair of machines to place in the layout (Step 2, below) incorporates the total flow between the pair to accomodate asymmetric material flow. Next, three modified rules for determining the subsequent machine to insert into the layout (Step 5) are proposed. These rules – named minFFasym, maxFFasym, and maxFFmod – are based on Chung and Tanchoco's MinFF and MaxFF heuristics. The complete minFFasym heuristic is described below, followed by a description of the maxFFasym and maxFFmod variants.

Step 1 – Initialization:

$I^a = \varnothing$;
$I^u = I$;

Step 2 – Select an initial pair of machines, i' and j', such that:

$\left(f_{i'j'} + f_{j'i'}\right)\left(\frac{w_{i'}+w_{j'}}{2}\right) = \max_{i\in I, j\in I}\left(f_{ij} + f_{ji}\right)\left(\frac{w_i+w_j}{2}\right)$;

Step 3 – Place i' and j' in opposite rows:

$\boldsymbol{\pi}_U = \{i'\}$;
$\boldsymbol{\pi}_L = \{j'\}$;

Step 4 – Update the lists of assigned and unassigned machines:

$I^a \leftarrow I^a \cup \{i', j'\}$;
$I^u \leftarrow I^u \setminus \{i', j'\}$;

Step 5 – Iteratively add one machine to the layout until all machines are assigned:

while $\left(|I^U| > 0\right)$ **do**

Select machine i' according to the following rule:
$i' = \underset{i\in I^u}{\text{argmin}} \sum_{j\in I^a}\left(f_{ij} + f_{ji}\right)\left(\frac{w_i+w_j}{2}\right)$;
$Z^* = \infty$;
Try inserting i' into each position of the partially constructed layout:
for all $(r \in \{U, L\})$ **do**

for all $(p \in \{1, 2, \ldots, |\boldsymbol{\pi}_r|\})$ **do**

$\boldsymbol{\pi}'_r = \{\pi_{r,1}, \ldots, \pi_{r,p-1}, i', \pi_{r,p}, \ldots, \pi_{r,|\boldsymbol{\pi}_r|}\}$;
Set the corresponding values of y_{ir} and z_{rij};
Use Formulation C to determine $Z^{LP}\left(\boldsymbol{\pi}'_U, \boldsymbol{\pi}'_L\right)$;
if $\left(Z^{LP}\left(\boldsymbol{\pi}'_U, \boldsymbol{\pi}'_L\right) < Z^*\right)$ **then**

$Z^* \leftarrow Z^{LP}\left(\boldsymbol{\pi}'_U, \boldsymbol{\pi}'_L\right)$;
$\boldsymbol{\pi}^*_U \leftarrow \boldsymbol{\pi}'_U$;
$\boldsymbol{\pi}^*_L \leftarrow \boldsymbol{\pi}'_L$;

end if

end for

end for
Update the lists of assigned and unassigned machines:
$I^a \leftarrow I^a \cup i'$;
$I^u \leftarrow I^u \setminus i'$;
Update the sequences of assigned machines:
$\boldsymbol{\pi}_U \leftarrow \boldsymbol{\pi}^*_U$;
$\boldsymbol{\pi}_L \leftarrow \boldsymbol{\pi}^*_L$;

end while

Step 6 – When all machines have been assigned, return the objective function value and row sequences, as provided by Z^*, $\boldsymbol{\pi}_U^*$ and $\boldsymbol{\pi}_L^*$, respectively.

The key difference between the MinFF and minFFasym heuristics is found in Step 5, where Chung and Tanchoco's MinFF heuristic selects machine i' according to

$$i' = \underset{i \in I^u}{\text{argmin}} \sum_{j \in I^a} f_{ij} \left(\frac{w_i + w_j}{2} \right)$$

for the case of symmetric material flow. Thus, minFFasym captures the total asymmetric flow between machines i and j. In a similar fashion, the proposed maxFFasym heuristic selects machine i' in Step 5 according to

$$i' = \underset{i \in I^u}{\text{argmax}} \sum_{j \in I^a} \left(f_{ij} + f_{ji} \right) \left(\frac{w_i + w_j}{2} \right),$$

and the proposed modMaxFF heuristic selects i' according to

$$i' = \underset{i \in I^u}{\text{argmax}} \sum_{j \in I^a} \frac{f_{ij} + f_{ji}}{\left(\frac{w_i + w_j}{2} \right)}.$$

The maxFFasym heuristic is modified from Chung and Tanchoco's symmetric MaxFF heuristic, while the new modMaxFF heuristic selects machines based on their ratio of material flow to the distance between load/unload points for a given pair of machines.

4.2 *Solution improvement via local search*

The aforementioned constructive heuristics do not include a mechanism for finding better neighbouring solutions. We propose a local search procedure that seeks to improve upon a solution obtained from any of the constructive heuristics described above. While it is possible to start the local search procedure with a random solution, we demonstrate in the next section that this is typically not an effective approach.

The local search procedure utilizes two operators to find candidate neighbouring sequences of machines: a pairwise swap of locations among all machines, and a move of each machine from its current location to each position in the opposite row. For each candidate sequence, it is possible to use the LP of Formulation C to find the optimal absolute location of each machine for that sequence. However, our initial testing indicated that this approach is time consuming and may lead the procedure to local minima. Instead, we evaluate each candidate sequence by determining the total cost of the layout with each pair of adjacent machines separated by exactly their minimum required clearances. This is shown in Figure 2, where the left most machines are placed along a virtual wall. The advantage of this approach is that finding the absolute location of each machine requires simply solving the following system of equations:

$$x_{\pi_{U,1}} = \frac{w_{\pi_{U,1}}}{2}, \tag{20}$$

$$x_{\pi_{L,1}} = \frac{w_{\pi_{L,1}}}{2}, \tag{21}$$

$$x_{\pi_{U,p}} = x_{\pi_{U,p-1}} + \frac{w_{\pi_{U,p-1}} + w_{\pi_{U,p}}}{2} + a_{\pi_{U,p-1}\pi_{U,p}} \qquad \forall\, p \in \{2, 3, \ldots, |\boldsymbol{\pi}_U|\}, \tag{22}$$

$$x_{\pi_{L,q}} = x_{\pi_{L,q-1}} + \frac{w_{\pi_{L,q-1}} + w_{\pi_{L,q}}}{2} + a_{\pi_{L,q-1}\pi_{L,q}} \qquad \forall\, q \in \{2, 3, \ldots, |\boldsymbol{\pi}_L|\}, \tag{23}$$

$$Z^{\text{minClear}}(\boldsymbol{\pi}_U, \boldsymbol{\pi}_L) = \sum_{i \in I^1} \sum_{j \in I^2} \left(f_{ij} + f_{ji} \right) \left| x_i - x_j \right|. \tag{24}$$

When the best neighbouring solution is found, this sequence is evaluated by Formulation C, whereby an exact optimization of the clearances between machines is obtained to determine if the resulting layout is better than the incumbent layout. This procedure is repeated until either a predetermined time limit is reached or no neighbouring solutions offer an improvement, based on the minimum-clearance objective function value given by Equations (20)–(24). Pseudocode of the proposed local search procedure is as follows:

Step 1 – Use a constructive heuristic (e.g. minFFasym) to find an initial solution, thus establishing an incumbent objective function value (Z^*) and layout sequences for each row ($\boldsymbol{\pi}_U^*$ and $\boldsymbol{\pi}_L^*$).

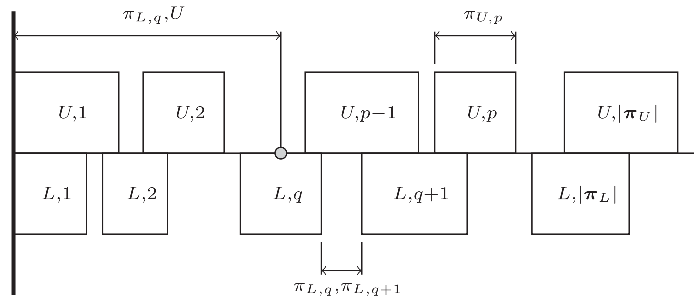

Figure 2. Illustration of parameters and decision variables for the DRLP in the minimum clearance situation.

Step 2 – Find the minimum-clearance objective function value using (20) – (24):

$Z' = Z^{\text{minClear}}(\boldsymbol{\pi}^*_U, \boldsymbol{\pi}^*_L)$;

Step 3 – Perform local search until termination criterion is reached:

$t = 0$;
`isImpr` = 1;
while $(t < t^{\max})$ and (`isImpr` == 1) **do**

  `isImpr` ← 0;
  Apply local search operators (swap and move), calculating $Z^{\text{minClear}}\left(\boldsymbol{\pi}'_U, \boldsymbol{\pi}'_L\right)$ for each candidate sequence of machines;
  if $Z^{\text{minClear}}\left(\boldsymbol{\pi}'_U, \boldsymbol{\pi}'_L\right) < Z'$ **then**

    $Z' \leftarrow Z^{\text{minClear}}\left(\boldsymbol{\pi}'_U, \boldsymbol{\pi}'_L\right)$;
    `isImpr` ← 1;
    $\boldsymbol{\pi}_U \leftarrow \boldsymbol{\pi}'_U$;
    $\boldsymbol{\pi}_L \leftarrow \boldsymbol{\pi}'_L$;

  end if
  if (`isImpr` == 1) **then**

    Calculate the LP objective function value using Formulation C;
    if $Z^{LP}(\boldsymbol{\pi}_U, \boldsymbol{\pi}_L) < Z^*$ **then**

      $Z^* \leftarrow Z^{LP}(\boldsymbol{\pi}_U, \boldsymbol{\pi}_L)$;
      $\boldsymbol{\pi}^*_U \leftarrow \boldsymbol{\pi}_U$;
      $\boldsymbol{\pi}^*_L \leftarrow \boldsymbol{\pi}_L$;

    end if

  end if
  Update the elapsed time, t;

end while

5. Computational results

A numerical analysis was conducted to evaluate the performance of the eight constructive heuristics and the local search procedure initialized with a solution from the minFFasym heuristic. Additionally, test problems were solved using CPLEX on Formulation T and the local search procedure was evaluated by initializing it with a random starting solution. All test problems were conducted using GNU Octave (an open-source MATLAB equivalent) version 3.2.4 on an HP 8100 Elite desktop PC with a quad-core Intel i7-860 processor running Ubuntu Linux 10.10 in 64-bit mode. When required, solutions to linear and integer programs were obtained by CPLEX version 12.2.0.

The test problems were created by adjusting the following parameter values: $m \in \{5, 10, 15, \ldots, 50\}$, f_{ij} ∼unif[0, 50], w_i ∼unif[0, 20], and a_{ij} ∼unif[1, 2] for all $i \in I$, $j \in \{I \setminus i\}$. Twenty instances were generated for each value of m, providing

Table 1. Comparison of average gap between each solution approach and the best-found solution [%].

m	MinLCF	MinFF	MaxFF	MinWF	MaxWF	maxFFmod	minFFasym	maxFFasym	CPLEX	LS-rand	LS-minFFasym
5	3.20	3.32	2.21	4.98	4.00	3.48	2.86	3.22	**0.00**	0.54	0.53
10	4.99	5.26	4.38	6.85	3.83	6.18	7.18	3.95	**0.00**	1.29	1.03
15	4.68	4.86	4.64	5.51	3.81	5.80	4.91	3.69	1.48	1.28	**0.61**
20	3.16	3.59	3.34	4.88	2.82	4.54	4.68	3.74	3.09	**0.38**	0.55
25	2.81	4.02	3.56	4.41	3.58	4.05	3.81	3.43	4.96	0.31	**0.30**
30	2.33	3.65	3.54	3.81	3.01	3.98	3.41	4.36	7.44	0.53	**0.12**
35	2.13	2.54	3.19	3.21	3.49	3.11	3.06	3.22	15.13	2.47	**0.03**
40	1.33	2.18	3.00	2.30	3.02	2.50	2.05	2.72	21.12	11.38	**0.03**
45	0.58	1.21	1.94	1.86	2.18	1.90	1.40	2.02	22.62	22.36	**0.17**
50	**0.14**	1.05	1.96	1.35	1.69	1.34	1.16	1.62	23.24	26.95	0.36
Best	22	7	7	4	7	4	5	10	33	47	108

Table 2. Average runtime [seconds].

m	MinLCF	MinFF	MaxFF	MinWF	MaxWF	maxFFmod	minFFasym	maxFFasym	CPLEX	LS-rand	LS-minFFasym
5	0.32	0.16	0.17	0.20	0.18	0.17	0.18	0.16	0.07	0.09	0.25
10	2.61	0.77	0.75	0.78	0.75	0.75	0.76	0.75	61.04	1.24	1.35
15	10.42	3.16	2.45	2.39	2.42	2.45	2.45	2.44	600.11	7.66	7.04
20	33.27	6.54	6.56	6.49	6.52	6.56	6.55	6.57	600.18	35.10	23.13
25	90.05	15.50	15.46	15.42	15.56	15.72	15.52	15.52	600.09	110.10	66.73
30	210.43	30.60	30.66	30.54	30.40	30.70	30.66	30.59	600.10	257.00	158.09
35	429.78	54.24	54.28	54.06	53.97	54.33	54.39	54.22	600.07	310.54	280.64
40	802.79	90.05	89.98	89.76	89.78	90.30	90.28	90.34	600.08	317.67	326.32
45	1410.06	145.20	145.64	144.69	144.02	145.36	145.21	145.38	600.14	324.39	333.89
50	2372.43	226.23	225.63	225.37	225.34	225.97	226.16	226.08	600.24	344.26	368.93

a total of 200 unique problems. CPLEX was given a time limit of 10 min, while the constructive and local search heuristics were given a time limit of 5 min.

Table 1 contains the average gap obtained by each solution approach over all problem sizes, as measured by the percentage difference between the approach's solution and the best-known solution. Results from the local search procedure initialised with the solution from the minFFasym heuristic are shown in the 'LS-minFFasym' column, while the 'LS-rand' column contains results from the local search initialised with a random starting solution. The last row of Table 1 indicates the number of test problems for which each solution approach found the best-known solution.

Table 2 contains the average runtime for each solution approach. Although the constructive and local search heuristics were given 300 s time limits, some of the larger problems exceeded this restriction because the timing check is performed at the beginning of each iteration. The MinLCF violated this limit in a particularly egregious manner, as this constructive heuristic considers the addition of all unassigned machines individually in Step 5 of the heuristic. As a result, MinLCF takes significantly longer to execute.

Figure 3 provides a boxplot summarising the spread of the results. To improve the readability of the plot, the LS-rand approach was omitted due to its poor performance on the larger-sized problems. As can be seen in the figure, the LS-minFFasym approach offers significantly better performance than any of the competing methods.

As expected, CPLEX performs well for small problems, solving 5- and 10-machine instances to optimality in about a minute. Interestingly, the CPLEX runtimes reach the 10-min limit when the number of machines is increased from 10 to 15. Not surprisingly, CPLEX's performance is quite poor for the larger problem sizes. Chung and Tanchoco's heuristics perform better than previously reported, presumably because the formulation issues led to erroneous results in their original paper. Of the eight constructive heuristics evaluated, MinLCF appears to provide the best solutions, at the expense of significantly longer runtimes. For the 50-machine problems, the local search heuristics were able to perform only two additional iterations. Given longer runtimes, we would expect the LS-minFFasym approach to outperform MinLCF on problems of this size.

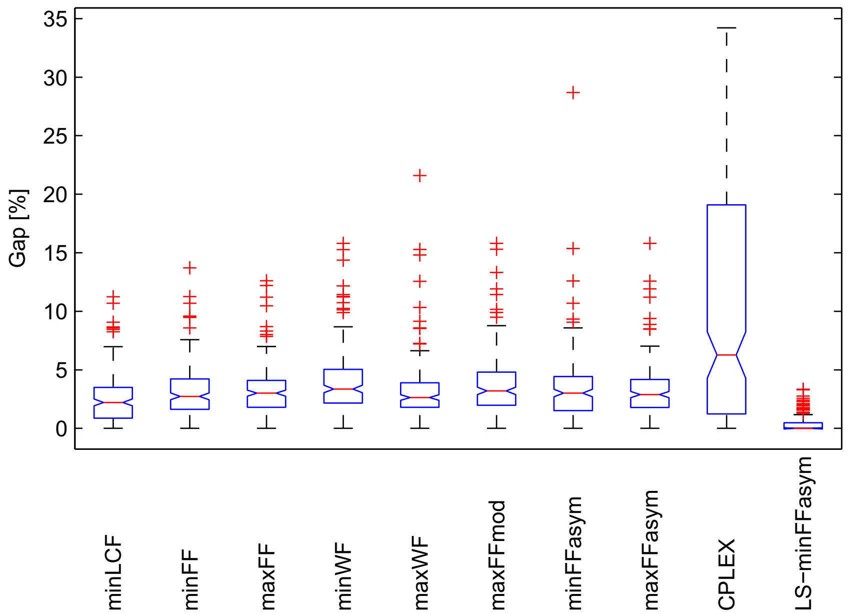

Figure 3. Performance of each solution across all problem sizes, as measured by the percentage gap from the best-found solution.

6. Conclusions

The DRLP is widely used in real-world production environments. This paper addresses the DRLP with asymmetric material flows, which had been neither modelled nor solved up to now. The problem is first formulated as a mixed integer linear programming problem, and is then solved by a combination of a constructive heuristic and a local search, both of which utilize an LP to obtain optimal machine locations for given sequences of machines in each row. The modelling thus allows the problem to be tractable and also reflective of the production goal: minimize flow times distance over all machine-to-machine flows, without restriction on the format of the flows, while maintaining required machine clearances.

A computational study revealed that the eight constructive heuristics (five from Chung and Tanchoco and three proposed variants) offer solutions of similar quality. However, by coupling a constructive heuristic with a simple local search heuristic, performance improved dramatically. These results show that the proposed algorithm is an effective approach for the DRLP, both in terms of solution quality and computational effort.

References

Amaral, A. R. S. 2008. “An Exact Approach to the One-Dimensional Facility Layout Problem.” *Operations Research* 56 (4): 1026–1033.

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.

Braglia, M. 1996. “Optimisation of a Simulated-Annealing-Based Heuristic for Single Row Machine Layout Problem by Genetic Algorithm.” *International Transactions in Operational Research* 3 (1): 37–49.

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.

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.

Hassan, M. M. D. 1994. “Machine Layout Problem in Modern Manufacturing Facilities.” *International Journal of Production Research* 32 (11): 25–59.

Heragu, S. 1992. “Recent Models and Techniques for Solving the Layout Problem.” *European Journal of Operational Research* 57 : 136–144.

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.

Komarudin, and K. Y. Wong. 2010. “Applying Ant System for Solving Unequal Area Facility Layout Problems.” *European Journal of Operational Research* 202 (3): 730–746.

Kouvelis, P., W.-C. Chiang, and G. Yu. 1995. “Optimal Algorithms for Row Layout Problems in Automated Manufacturing Systems.” *IIE Transactions* 27 : 99–104.

Kulturel-Konak, S., and A. Konak. 2011. “Unequal Area Flexible Bay Facility Layout Using Ant Colony Optimization.” *International Journal of Production Research* 49 (7): 1877–1902.

McKendall, A. R., Jr., and J. Shang. 2006. “Hybrid Ant Systems for the Dynamic Facility Layout Problem.” *Computers & Operations Research* 33 (3): 790–803.

Montreuil, B., E. Brotherton, and S. Marcotte. 2002. “Zone-Based Facilities Layout Optimization.” In *Proceedings of the Industrial Engineering Research Conference, IIE Annual Conference*, 1–6, Orlando, FL.

Montreuil, B., N. Ouazzani, E. Brotherton, and M. Nourelfath. 2004. “Coupling Zone-Based Layout Optimization, Ant Colony System and Domain Knowledge.” In *Proceedings of the 8th International Material Handling Research Colloquium*, 301–331, Graz: Material Handling Institute of America.

Peters, B. A., and T. Yang. 1997. “Integrated Facility Layout and Material Handling System Design in Semiconductor Fabrication Facilities.” *IEEE Transactions on Semiconductor Manufacturing* 10 (3): 360–369.

Pour, H. D., and M. Nosraty. 2006. “Solving the Facility Layout and Location Problem by Ant-Colony Optimization-Meta Heuristic.” *International Journal of Production Research* 44 (23): 5187–5196.

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.

See, P. C., and K. Y. Wong. 2008. “Application of Ant Colony Optimisation Algorithms in Solving Facility Layout Problems Formulated as Quadratic Assignment Problems: A Review.” *Journal of Industrial and Systems Engineering* 3 (6): 644–672.

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.

Tate, D. M., and A. E. Smith. 1995. “A Genetic Approach to the Quadratic Assignment Problem.” *Computers & Operations Research* 22 (1): 73–83.

Teo, Y. T., and S. G. Ponnambalam. 2008. “A Hybrid ACO/PSO Heuristic to Solve Single Row Layout Problem.” In *4th IEEE Conference on Automation Science and Engineering*, 597–602, Washington, DC.

Wong, K. Y., and Komarudin. 2010. “Solving Facility Layout Problems Using Flexible Bay Structure Representation and Ant System Algorithm.” *Expert Systems with Applications* 37 (7): 5523–5527.

Xie, W., and N. V. Sahinidis. 2008. “A Branch-and-Bound Algorithm for the Continuous Facility Layout Problem.” *Computers & Chemical Engineering* 32 (4): 1016–1028.

Yang, T., and B. A. Peters. 1997. “A Spine Layout Design Method for Semiconductor Fabrication Facilities Containing Automated Material-Handling Systems.” *International Journal of Operations & Production Management* 17 (5): 490–501.

Zhang, Z., and C. C. Murray. 2011. “A Corrected Formulation for the Double Row Layout Problem.” *International Journal of Production Research* 50 (15): 4220–4223.