

AN EXTENDED DOUBLE ROW LAYOUT PROBLEM

Chase C. Murray
Auburn University

Xingquan Zuo
Beijing University of Posts and Telecommunications

Alice E. Smith
Auburn University

Abstract

The double row layout problem (DRLP) seeks to determine optimal machine locations on either side of an aisle, where the objective has been defined as the minimization of material flow cost among machines while meeting machine clearance constraints. In this paper, we extend existing DRLP formulations in two respects. First, we consider the minimization of layout area besides the usual material flow cost objective. Second, we present a mixed integer linear programming formulation that permits non-zero aisle widths. This new formulation also includes new constraints that eliminate layout “mirroring,” thus reducing the solution space significantly and thus solution times. Although small-scale problems may be solved optimally by commercial integer programming solvers, solution times are highly sensitive to the number of machines in a layout. A tabu search heuristic is shown to work well for moderately-sized problems. Numerical examples demonstrating the impact of both flow and area objectives, as well as aisle widths, are included.

1 Introduction

In this paper we consider a facility layout problem where rectangular machines of unequal size must be placed in two rows separated by a straight aisle of predetermined width. Each pair of adjacent machines must be separated by *at least* a minimum pre-specified clearance. It is assumed that the load/unload port of each tool is located at the midpoint of the tool’s width. In this problem, machines must be placed such that both the total cost of material flow and the area consumed by the layout are minimized. While the material handling facility layout literature has focused on cost

of material flow, generally area consumed has been ignored, or at most, treated as a fixed constraint. However, for some industries and situations, floor space is at a premium and the area objective can be an important one. Such a problem has widespread application particularly in the semiconductor industry, where material flow can be complex and clean room construction costs are approximately \$3,500 per square foot [9].

While the particular problem of interest here shares similarities with other existing facility layout problems, there are some key characteristics that prohibit the direct application of these mathematical models. The first similar problem is the double row layout problem (DRLP), for which the first mathematical model was proposed by [1] and was later corrected by [12]. In the DRLP, the problem is to minimize the total weighted cost of material flow by placing machines in one of two rows. Adjacent machines must be separated by at least a minimum pre-determined distance. Thus, this problem is unique in that it contains both combinatorial and continuous aspects. However, previous DRLP formulations assume aisle widths of zero and the area of the layout is not considered. We demonstrate that ignoring these aspects can lead to inferior solutions.

The incorporation of a center aisle of fixed width was considered by [6] and [11] in the context of semiconductor layout and material handling configurations. In these models, tools are assigned to equally-sized rectangular blocks, thus making the precise determination of each tool's location difficult.

Another related problem is the multi-row layout problem (MRLP). While the MRLP would appear to be a generalization of our problem, it has not allowed machines to be separated by a distance greater than the minimum required clearance (c.f., [2]). Furthermore, the MRLP has not sought to minimize the area of the layout. Yet another related study is provided by [4]. However, his model relies upon an absolute value expression in the objective function and a nonlinear constraint, thus making optimal solution approaches difficult. Because that model does not require straight aisles between rows of machines, resulting layouts may not be applicable to semiconductor fabs or other production environments. Facility layout problems that are formulated as quadratic assignment problems (QAP), such as [8], assume predefined possible machine locations and therefore do not address the continuous nature of the problem at hand.

A block layout problem that mixes combinatorial and continuous aspects was considered by [5]. Machine centers are assigned to pre-specified zones within a layout of fixed dimensions. Once machine centers are assigned to a zone, the input/output points of the center are determined in continuous space. To solve this problem, ant colony optimization is utilized to determine the assignment of centers to zones, followed by the solution to a linear programming problem to determine optimal locations of the input/output points. A similar problem, the continuous facility layout problem, was addressed by [10]. This problem did consider minimum clearance re-

quirements between machines, however neither aisle widths nor area minimization are included. A branch-and-bound solution approach was tested on problems with up to 12 machines.

And finally, unlike [1], who encountered runtimes of two hours when solving problems involving only 10 machines, we demonstrate that our formulation can solve 10-machine problems optimally using CPLEX in less than 30 seconds. This runtime reduction is driven by improvements in the formulation of the model, including two simple, yet highly effective, constraints that significantly decrease solution times by reducing symmetry in the problem.

To summarize, our work makes the following contributions. First, unlike other related studies on the DRLP, our model allows non-zero aisle widths and considers layout area minimization. Numerical examples demonstrate that the consideration of aisle width and area minimization can significantly alter the best layout design. Second, the proposed mathematical formulation allows the minimization of layout area via a linear objective function and linear constraints. Finally, a tabu search heuristic that utilizes the mathematical formulation is shown to be effective and tractable for problems of reasonable size.

The remainder of this paper is organized as follows. The proposed mixed integer linear program (MILP) is formulated and explained in Section 2. Section 3 provides numerical examples to highlight the impact of ignoring aisle widths and area. A tabu search heuristic, integrated with the MILP model, is described in Section 4. Numerical analysis of the proposed tabu search heuristic is provided in Section 5. Finally, Section 6 concludes with a summary and suggestions for future research.

2 Mathematical Formulation

The notation and mathematical model are rooted in previous DRLP research by [1] and [12]. Two important features that distinguish the proposed model from previous DRLP models are the non-zero aisle width and the minimization of total layout area. We assume rectilinear travel for material flow across and along the aisle.

Notation for parameters and decision variables are defined in Tables 1 and 2, respectively.

The following MILP model serves to minimize material handling costs and total layout area. These two (possibly) competing objectives are incorporated in a linearly additive fashion by virtue of a scaling parameter $0 \leq \alpha \leq 1$, such that if $\alpha = 1$ the objective is to minimize material handling cost only and if $\alpha = 0$ the objective is to minimize the total area only.

Table 1: Parameters

m	Number of machines
I	Set of machines, where $I = \{1, \dots, m\}$
R	Set of rows, where $R = \{1, 2\}$ specifies the upper (1) and lower (2) rows
w_i	Width of machine $i \in I$
d_i	Depth of machine $i \in I$
a_{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 $i \in I^1$.
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 the upper and lower rows.

Table 2: Decision Variables

x_{ir}	Continuous decision variable representing the location of machine $i \in I$ in row $r \in R$, such that $x_{ir} = 0$ if i is not placed in row r .
y_{ir}	Binary decision variable, such that $y_{ir} = 1$ if machine $i \in I$ is placed in row $r \in R$.
z_{rij}	Binary decision variable, such that $z_{rij} = 1$ if machine $i \in I$ is placed to the left of machine $j \in \{I \setminus i\}$ in row $r \in R$.
W	Width of the resulting layout. This is the maximum distance between the left side of the first machine in either row and the right side of the last machine in either row.
s_r	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 (and the aisle between the rows). Clearance space for end machines of rows is not considered.
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.

$$\text{Minimize} \quad \alpha \sum_{i \in I^1} \sum_{j \in I^2} (f_{ij} + f_{ji}) (v_{ij}^+ + v_{ij}^- + c(1 - q_{ij})) + (1 - \alpha)A \quad (1)$$

$$\text{subject to:} \quad x_{ir} \leq M y_{ir}, \quad \forall i \in I, r \in R, \quad (2)$$

$$\sum_{r \in R} y_{ir} = 1 \quad \forall i \in I, \quad (3)$$

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

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

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

$$z_{rij} + z_{rji} \leq y_{ir} \quad \forall i \in I^1, j \in I^2, r \in R, \quad (7)$$

$$z_{rij} + z_{rji} \leq y_{jr} \quad \forall i \in I^1, j \in I^2, r \in R, \quad (8)$$

$$z_{rij} + z_{rji} + 1 \geq y_{ir} + y_{jr} \quad \forall i \in I^1, j \in I^2, r \in R, \quad (9)$$

$$W \geq x_{ir} + \frac{1}{2} w_i y_{ir} \quad \forall i \in I, r \in R, \quad (10)$$

$$x_{ir} - \frac{1}{2} w_i y_{ir} \geq 0 \quad \forall i \in I, r \in R, \quad (11)$$

$$s_r \geq d_i W - d_i M(1 - y_{ir}) \quad \forall r \in R, i \in I, \quad (12)$$

$$A = s_1 + s_2 + cW, \quad (13)$$

$$q_{ij} = \sum_{r \in R} (z_{rij} + z_{rji}) \quad \forall i \in I^1, j \in I^2, \quad (14)$$

$$x_{ir} \geq 0 \quad \forall i \in I, r \in R, \quad (15)$$

$$v_{ij}^+, v_{ij}^- \geq 0 \quad \forall i \in I^1, j \in I^2, \quad (16)$$

$$y_{ir} \in \{0, 1\} \quad \forall i \in I, r \in R, \quad (17)$$

$$z_{rij} \in \{0, 1\} \quad \forall i \in I, j \in \{I \setminus i\}, r \in R, \quad (18)$$

$$q_{ij} \in \{0, 1\} \quad \forall i \in I^1, j \in I^2, \quad (19)$$

$$s_r \geq 0 \quad \forall r \in R, \quad (20)$$

$$A \geq 0, \quad (21)$$

$$W \geq 0. \quad (22)$$

The objective function (1) seeks to minimize the weighted total cost of material handling plus the weighted area of the resulting layout. Constraints (2) and (3) ensure that each machine is placed in exactly one row. Constraints (4) and (5) guarantee that the minimum clearance between adjacent machines is satisfied. The absolute value of the horizontal distance between machines is determined by Constraint (6). Constraints (7) – (9) relate binary 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, $z_{rij} = z_{rji} = 0$. Due to the difficulties associated with calculating the area of a layout via linear relationships, Constraints (10) – (12) are employed to determine lower bounds on the width (horizontal dimension) and area of a layout. The objective function term that seeks to minimize the total layout area, A , serves to make constraints (10) and (12) binding. Constraint (13) determines the total area of the resulting layout as the sum of the areas of the upper row, the

lower row, and the corridor (aisle) separating the rows. Again, the objective function seeks to minimize the total area of the layout. Constraint (14) determines whether machines i and j are in the same row. Finally, (15) – (22) describe the decision variable definitions. The constant M is a sufficiently large number, and may be given by

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

It should be noted that decision variable q_{ij} may be defined as continuous, but bounded between 0 and 1. One of the nice properties of this formulation is that these decision variables will necessarily be binary. Thus, the only decision variables that must be explicitly solved as integer are y_{ir} and z_{rij} . Intuitively, this reduction in the number of integer decision variables should reduce solution times. Preliminary testing indicates that modest performance improvements are observed by relaxing the integer requirement on q_{ij} . However, results from these tests are not included here due to numerical issues within CPLEX, which produced some solutions containing fractional values of q_{ij} . The exact cause of these issues is unknown. Therefore, in all problem instances we explicitly define q_{ij} to be binary.

A feature of layout problems is that they are often plagued by “mirroring” issues. For example, suppose an optimal solution is found where machines in the upper row are arranged in the following sequence: 1, 5, 2, 7. There is also an optimal solution whereby these machines may be arranged in reverse (i.e., 7, 2, 5, 1). Furthermore, there are optimal solutions containing these sequences of machines in the lower row. This behavior significantly increases the computational complexity of the problem. This behavior was observed by [7], who proposed a symmetry-breaking constraint for layout problems of known widths. Fortunately, the effects of symmetry may also be reduced in our problem, where the width is unknown, using the following constraints:

$$y_{i^*,1} = 1 \tag{23}$$

$$x_{i^*,1} \leq \frac{W}{2} \tag{24}$$

In these constraints, machine i^* may be chosen randomly. However, to improve the bounds on the area of resulting layouts, we have chosen to define i^* to be the machine with the maximum depth (i.e., $d_{i^*} = \max_{i \in I} \{d_i\}$). Constraint (23) forces machine i^* to be placed in the upper row (row 1). Constraint (24) locates machine i^* in the left half of the upper row. It is important to note that these are valid constraints, in that they do not prohibit the determination of a globally optimal solution. They do, however, prevent mirror layouts from being considered.

For the particular case of $\alpha = 1$ (i.e., the objective is solely to minimize the flow cost), the symmetry-elimination constraints may not work well. This is due to the fact that there is no active objective function term that seeks to minimize the value of A . Therefore, Constraint (10) is not guaranteed to be tight, and W (the width of the layout) can become an arbitrarily large number. Furthermore, the value of W can become decoupled from the location of the right-most machine in the layout. One option to address this issue would be to limit α to be strictly less than 1. We believe a better option is to incorporate the following constraints when $\alpha = 1$:

$$W \leq x_{i1} + \frac{w_i}{2} + M \left((1 - y_{i1}) + \sum_{\substack{j \in I \\ j \neq i}} z_{1ij} + (1 - l) \right) \quad \forall i \in I, \quad (25)$$

$$W \leq x_{i2} + \frac{w_i}{2} + M \left((1 - y_{i2}) + \sum_{\substack{j \in I \\ j \neq i}} z_{2ij} + l \right) \quad \forall i \in I, \quad (26)$$

$$l \in \{0, 1\}. \quad (27)$$

Constraints (25) and (26) establish an upper bound on the width of the layout. Note that $\sum_{\substack{j \in I \\ j \neq i}} z_{rij}$ equals zero if machine i is the right-most machine in row r . The binary

decision variable l , which assumes a value of one (zero) if the upper (lower) row is the longest, ensures that either constraint (25) or (26) will be active.

To assess the effectiveness of the mirror-prevention constraints, 80 10-machine problem instances were solved optimally both with and without these additional constraints. Details about the manner in which these problems were generated are contained in Section 5. When only constraints (2) – (22) were employed, the average solution time was 46.33 seconds. However, when constraints (23) – (27) were added, the average solution time decreased to 20.51 seconds. The results of this analysis indicate that the mirror-prevention constraints offer dramatic decreases in average runtimes which means that problems of larger size may be solved optimally.

3 Numerical Example and Observations

This section presents an analysis of a small-scale numerical example to demonstrate the impacts of aisle widths and area on optimal layouts. Information related to material flow costs, minimum required clearances, and machine sizes are described in Tables 3, 4, and 5, respectively. All of the representative layouts were solved optimally via CPLEX 12.0 in under 30 seconds.

Table 3: Cost of material flow between machines, f_{ij} .

from\to	1	2	3	4	5	6	7	8	9	10
1	-	-	32	-	-	33	-	40	-	33
2	-	-	56	-	-	-	20	-	-	-
3	-	-	-	-	33	-	-	58	55	-
4	-	-	23	-	-	-	-	-	-	23
5	73	35	33	-	-	30	-	23	-	-
6	-	20	-	23	-	-	-	33	-	30
7	33	-	-	-	-	-	-	-	-	-
8	32	-	-	23	35	23	-	-	-	40
9	23	32	-	-	-	-	-	-	-	-
10	-	33	-	-	33	-	-	30	-	-

Table 4: Minimum required clearance between machines, a_{ij} .

from\to	1	2	3	4	5	6	7	8	9	10
1	-	1.13	1.54	1.79	1.17	2.13	0.59	1.94	0.59	0.83
2	1.13	-	0.62	0.95	2.05	1.75	1.98	0.43	1.7	1.25
3	1.54	0.62	-	0.36	2.15	1.42	1.82	0.35	1.76	0.94
4	1.79	0.95	0.36	-	0.58	0.84	1.88	0.34	1.49	1.7
5	1.17	2.05	2.15	0.58	-	1.76	0.43	1.44	1.61	0.96
6	2.13	1.75	1.42	0.84	1.76	-	0.45	1.44	0.67	1.57
7	0.59	1.98	1.82	1.88	0.43	0.45	-	1.81	1.41	1.72
8	1.94	0.43	0.35	0.34	1.44	1.44	1.81	-	0.57	0.29
9	0.59	1.7	1.76	1.49	1.61	0.67	1.41	0.57	-	0.71
10	0.83	1.25	0.94	1.7	0.96	1.57	1.72	0.29	0.71	-

Table 5: Machine sizes.

Machine, i	1	2	3	4	5	6	7	8	9	10
w_i	1.14	0.51	0.75	0.82	1.36	0.76	0.84	1.48	1.27	1.32
d_i	0.81	1.18	1.19	0.70	1.15	0.50	1.40	1.13	1.46	1.10

Figure 1 contains layouts resulting from changes to the width of the aisle separating the two rows of machines for the two extreme values of α . This example demonstrates that the width of the aisle can dramatically affect the optimal layout. Such a consideration was overlooked in all previous research on the DRLP, where aisle widths were assumed to be zero.

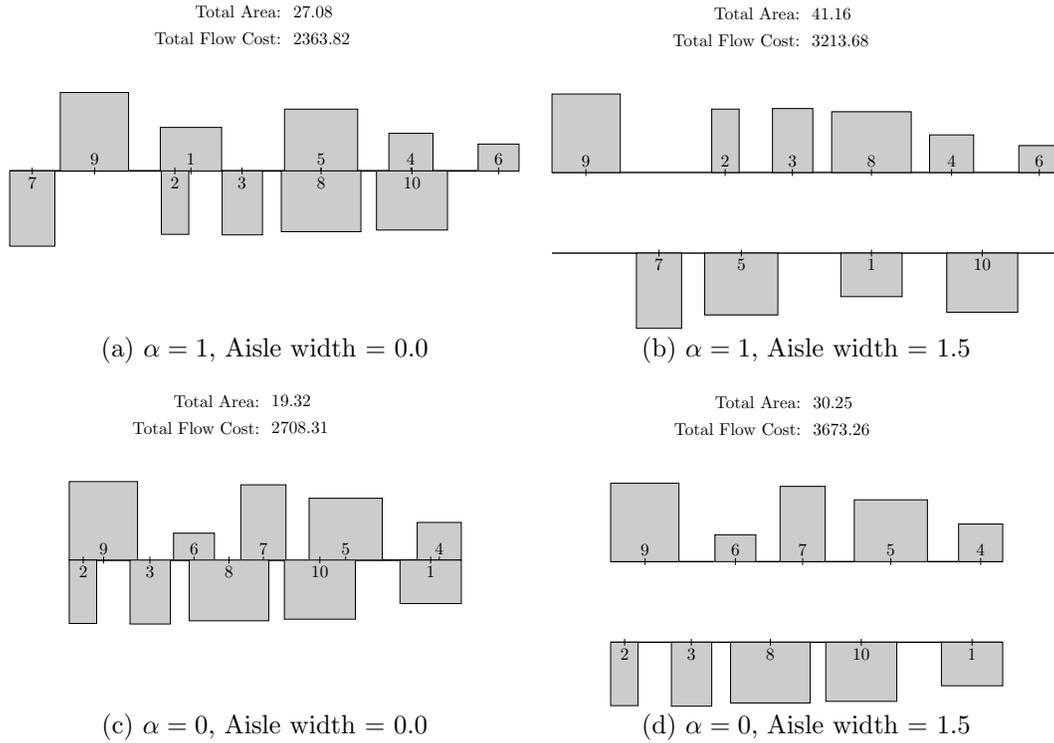


Figure 1: Effects of changing α and aisle width.

Figure 2 demonstrates layout changes as focus is shifted from minimizing total cost of material flow ($\alpha = 1.0$) to minimizing total area of layout ($\alpha = 0.0$). It should be noted that $\alpha = 0.5$ does not imply that the minimization of total flow cost and area are equally weighted due to the significant differences in the scale of these two metrics. For example, the optimal layout with an aisle width of 1.5 and $\alpha = 0.5$ has a total flow cost of 2675.82 and consumes an area of 31.82, which is the same as the case of $\alpha = 1.0$ (Figure 2a). Similarly scaled individual objective function contributions are seen in Figure 2c, with $\alpha = 0.01$, where the weighted area is $(24.16)(0.99) = 23.92$ and the weighted flow cost is $(2824.29)(0.01) = 28.24$. Future research opportunities exist for transforming this problem into a bi-objective formulation which would avoid the issue of scaling the two objectives. Comparing the layouts in Figures 1 and 2 indicates that the consideration of both aisle width and total layout area are important when developing a design for two rows of machine, especially if space is expensive or constrained.

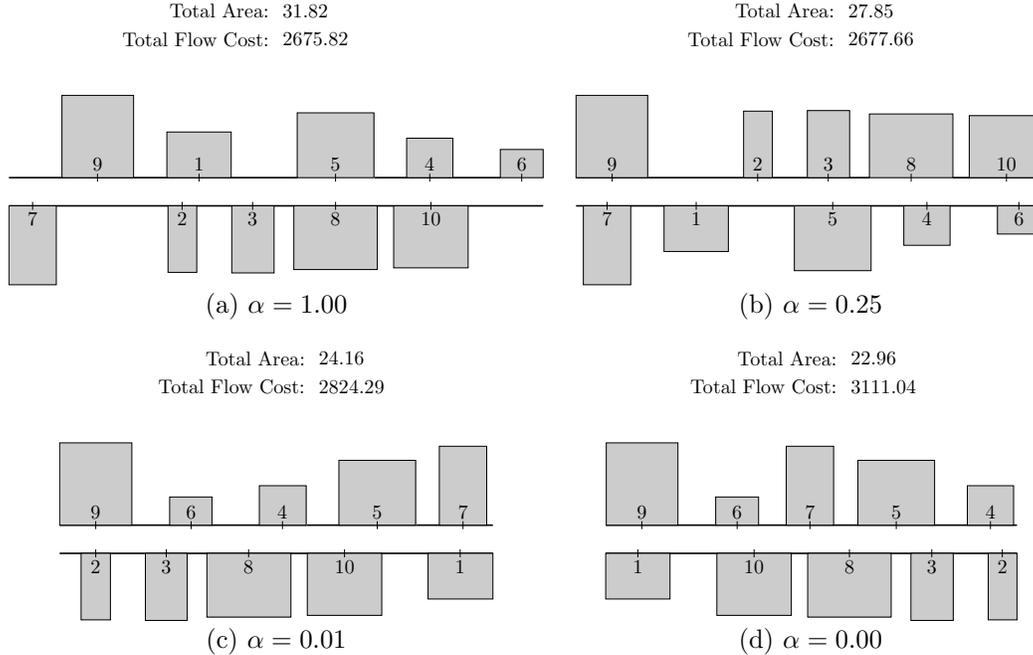


Figure 2: Effects of changing α for a fixed aisle width of 0.5.

4 Tabu Search Heuristic

A tabu search (TS) heuristic is proposed for large-scale DRLP instances that cannot be solved by CPLEX within reasonable computational time. TS is a meta-heuristic algorithm that has been successfully applied to a variety of combinatorial optimization problems [3]. It is especially effective and efficient for problems with a good neighborhood structure, which we believe exists here. The procedure starts with an initial randomly-generated solution, s , which represents the sequence of machines in each row of a layout. In each iteration of the procedure the current solution is replaced by a neighboring solution via the two operators described below. To limit search cycles, a *tabu list* containing a history of recent operators employed to obtain neighboring solutions is maintained. Operators contained in the tabu list, whose length may change over time, are forbidden when constructing candidate solutions. As new operators are added to the list, old operators are removed, thus maintaining only a recent history of forbidden moves. The procedure terminates when either a pre-specified maximum runtime, $maxTime$, is encountered, or the number of consecutive iterations for which no improving solutions are found, $maxIter$, is reached.

Candidate solutions in the neighborhood of s are first determined by a *move* operation. A move is designed to swap the locations of any two machines in a solution.

In each iteration, all possible moves of current solution s are calculated to compose the move set $V(s)$. For a problem with n machines, the number of solutions that can be reached by all moves of current solution s is $n(n-1)/2$. The solutions obtained by utilizing the moves in $V(s)$ comprise the neighborhood of s . The best non-forbidden move selected from $V(s)$ is applied to the current solution, resulting in a new solution, s' . By default, the objective function value of this new solution, represented by $f(s')$, is determined by separating each machine in the sequence s' by its minimum clearance with all adjacent machines. Later we describe a procedure in which the integer programming formulation is employed to determine the optimal location of each machine given a particular sequence.

The move used to create s' is added to the tabu list, T , which contains the most-recently swapped machine pairs. The proposed procedure utilizes a dynamic tabu list, such that every 20 iterations the length of the tabu list is determined by a random integer within pre-specified lower and upper bounds. It has been noted in the literature that a dynamic length tabu list tends to result in more robust search. If T is full, the earliest move in the list is removed and the current move is appended. At this point, new solution, s' , replaces the previous solution (i.e., $s = s'$).

Next, a *permutation* operator considers changing the assignment of each machine from one row to the other. For a problem with n machines, if the current solution has p machines in row 1 and q machines in row 2 ($p + q = n$), then the total number of solutions produced by the permutation operator is $2pq + n$. These solutions comprise the permutation set $U(s)$. The best one, s'' , is selected from $U(s)$ to compare with current solution s . Unlike the move operator, which will always update the current solution with a new solution, the permutation operator will only result in an update to the current solution if s'' represents a better solution than s . Thus, if $f(s'') < f(s)$ then s is replaced by s'' . This part of the TS essentially acts as local search after the best swap move has been determined. It serves to search areas that change the number of machines per row (which the swap operator cannot do).

The steps of the proposed TS heuristic are as follows, where s^* represents the best-known solution, *iter* is an iteration counter, and *time* reflects the elapsed time:

Step 1: Randomly initialize the current solution, s ; let $s^* = s$, $T = \emptyset$, and $iter = 0$.

Step 2: Calculate the move set $V(s)$ of current solution s . Let $iter = iter + 1$.

Step 3: Choose the best non-forbidden move $v \in V(s)$, then v is applied to current solution s to produce a new solution s' . Let $s = s'$. Update tabu list T .

Step 4: Calculate the permutation set $U(s)$ of current solution s .

Step 5: Choose the best solution $s'' \in U(s)$. If $f(s'') < f(s)$, then let $s = s''$; otherwise, s'' is discarded.

Step 6: If $f(s) < f(s^*)$, then let $s^* = s$, $iter = 0$ and return to Step 2.

Step 7: If $(iter < maxIter)$ and $(time < maxTime)$, then return to Step 2; otherwise TS stops.

As previously mentioned, the TS heuristic defaults to using minimum machine clearances when determining the objective function value. However, for a given sequence of machines in each row, optimal layouts when considering material flow may require additional clearance between some pairs of machines. Fortunately, the determination of optimal machine locations for a given sequence can be easily obtained by solving the mixed integer program defined in Section 2. Specifically, a sequence obtained by TS is used to set the values of the binary q_{ij} , y_{ir} , and z_{rij} decision variables. The only remaining unknown decision variable values are for continuous variables. Thus, the optimal machine locations (x_{ir} decision variables) may be obtained by solving a linear program, which may be solved very quickly. Montreuil et al. [5] employed a similar approach using ant colony optimization for a block layout problem.

In the next section we investigate the four following methods in which the linear program may be employed within the TS heuristic:

TS-0: Tabu search using only minimum clearance separation between tools. The linear program is not employed.

TS-1: Tabu search with only the final solution sent to CPLEX (i.e., the linear program is solved once and only after the completion of Step 7).

TS-2: Tabu search with only the best candidate neighbor solution sent to CPLEX (i.e., the linear program is solved once per neighborhood and only for the best solution obtained in Step 5).

TS-3: Tabu search with all neighbors (candidate moves) sent to CPLEX (i.e., the linear program is solved for each solution investigated in Step 5).

5 Numerical Analysis

In this section we examine the effectiveness of the tabu search heuristic and its variants. Table 6 contains the parameter settings used to generate the test problems, where 80 test problems were created for each value of m (20 distinct settings of machine sizes, clearances, and flow costs (that is, 20 problem instances); two values of α ; and two aisle widths).

To generate realistic product flow frequencies, we assumed that p distinct product types exist, where $p \sim \text{unif}[8, 10]$, such that each product within a particular type visits the same sequence of machines. Let r equal the percentage of machines visited

by each product type’s route, where $r \sim \text{unif}[0.25, 0.75]$. The number of products of each type, n , is such that $n \sim \text{unif}[20, 50]$. Assuming unit costs, the f_{ij} values were calculated as the sum of products whose routes included machine i immediately preceding machine j . All problem data is available upon request from the authors.

Table 6: Parameter values for problem instance creation.

Parameter	Values
m	10, 15, 20, 25
α	0, 1
w_i	$\sim \text{unif}[0.5, 2.5]$
d_i	$\sim \text{unif}[0.5, 2.5]$
c	$0, (\max_{i \in I} \{d_i\} + \min_{i \in I} \{d_i\})/2$
a_{ij}	$\sim \text{unif}[0.25, 1.5]$

Tabu search parameter settings for each problem size are described in Table 7. These parameter values were chosen after preliminary testing on other randomly-generated test problems. The TS is not particularly sensitive to these values – the main idea is to scale up the parameters as problem size grows.

Table 7: Tabu search parameter configurations.

m	TS List Length		Termination Criteria	
	min	max	$maxIter$	$maxTime$ [sec]
10	2	4	800	300
15	3	5	1200	300
20	5	8	1500	300
25	6	9	1700	300

A comparison of the four TS heuristics is shown in Table 8. In this analysis, 80 problem instances (20 settings of tool sizes \times 2 settings of α \times 2 settings of c) were solved for 10-machine layouts, where corridor clearance values were chosen to be either zero ($c = 0$) or equal to the average of the minimum and the maximum values of the tool depths ($c = \text{avg}$). Each TS heuristic was executed for one iteration, with $maxTime = 300$ seconds. The integer programming formulation of Section 2 was solved optimally via CPLEX. These optimal solutions were used in the determination of the average gap for each TS variant. Each TS heuristic ran for the full allotment of 300-seconds, while the average runtime for CPLEX was 20.51 seconds. Although these results do not suggest that any of the TS variants are recommended over CPLEX for small problems, they do indicate that TS-2 appears to be the most effective of

the variants. Furthermore, as we will see shortly, TS-2 becomes a more attractive approach relative to CPLEX for larger problem sizes. These results also demonstrate the value of not only considering the minimum clearances. Note that TS-0 which does not use CPLEX to adjust the machine locations in continuous space does very poorly.

Table 8: Gap comparison of TS heuristics on 80 10-machine problem instances.

	$\alpha = 0$		$\alpha = 1$		Avg. Gap	# Optimal Solns
	$c = 0$	$c = \text{avg}$	$c = 0$	$c = \text{avg}$		
TS-0	1.30%	0.80%	1.17%	1.12%	1.10%	26 of 80
TS-1	1.30%	0.80%	0.39%	0.43%	0.73%	46 of 80
TS-2	1.22%	0.80%	0.00%	0.05%	0.52%	63 of 80
TS-3	1.39%	0.82%	0.21%	0.30%	0.68%	59 of 80

A separate analysis involving 20 10-machine problems, with $\alpha = 0.75$ and an aisle width equal to the average of the minimum and maximum machine depths, was conducted to determine solution variability due to the random number seed used in TS-2. Six replications were run for each problem instance. In 16 of 20 problems all six seeds produced identical objective function values. For the remaining four problems, the average standard deviation corresponded to 0.6% of the lowest objective function value. As a result, it is apparent that the TS heuristic is not very dependent upon the random number generator.

Results of the numerical experiments are summarized in Table 9, where 80 problem instances were investigated for each value of m . All problems were restricted to a 5-minute (300-second) maximum runtime. With the exception of CPLEX on the 10-machine problems, both solution approaches were terminated due to the runtime limit. The average gap is reported with respect to the best solution obtained by either method. The last three columns of the table contain the number of problem instances in which each method determined the best known solution.

Table 9: TS-2 heuristic performance over varying problem sizes.

m	Avg. Gap		# Best Solutions		
	TS-2	CPLEX	TS-2	CPLEX	Tie
10	0.52%	0.00%	0	17	63
15	0.79%	4.10%	53	25	2
20	0.44%	6.55%	71	9	0
25	0.08%	10.16%	78	2	0

6 Summary

This paper presented an extended formulation of the DRLP, where machines are placed on either side of an aisle, subject to minimum clearance requirements between machines. While previous DRLP formulations assumed aisle widths of zero, the proposed formulation considers arbitrary aisle widths. Furthermore, unlike most existing layout formulations that seek to minimize material flow costs only, this formulation also considers the minimization of the layout area. Numerical results demonstrate the effects of non-zero aisle widths and area minimization on optimal layouts and show that ignoring area and assuming zero width aisles would identify layouts that would be far from optimal. Furthermore, considering machine placement at more than the minimum required clearance is important. For most of our test problems, machines were not placed at their minimum clearance only, but rather extra space was used to reduce flow costs.

For small-scale problems, the proposed MILP formulation may be solved optimally via commercial integer programming solvers quite quickly, in part due to the inclusion of our symmetry-elimination constraints. However, even for moderately-sized problems, exact approaches become time-prohibitive. Therefore, a tabu search heuristic was developed. This heuristic determines sequences of machines in each row of the layout, which is fed to CPLEX for exact solving of the best machine spacing for a given sequence. Numerical analysis shows that the tabu search heuristic coupled with CPLEX significantly outperforms CPLEX alone for problems of non-trivial size.

Future research opportunities on this topic include effective methods for determining Pareto optimality in a true bi-objective context (as opposed to the linearly-additive manner in which the objectives of minimizing flow cost and layout area were treated in this paper). This problem could also be extended to consider multiple rows. Finally, more effective exact solution methods that exploit the structure of the problem could be developed to solve larger-scale problems.

References

- [1] J. Chung and J.M.A. Tanchoco. The double row layout problem. *International Journal of Production Research*, 48(3):709–727, 2010.
- [2] M. Ficko, M. Brezocnik, and J. Balic. Designing the layout of single-and multiple-rows flexible manufacturing system by genetic algorithms. *Journal of Materials Processing Technology*, 157:150–158, 2004.
- [3] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1998.
- [4] S. Heragu. Recent models and techniques for solving the layout problem. *European Journal of Operational Research*, 57:136 – 144, 1992.
- [5] B. Montreuil, N. Ouazzani, E. Brotherton, and M. Nourelfath. Coupling zone-based layout optimization, ant colony system and domain knowledge. In *Proceedings of the 8th*

- International Material Handling Research Colloquium*, pages 301–331, Graz, Austria, 2004. Material Handling Institute of America.
- [6] B.A. Peters and T. Yang. Integrated facility layout and material handling system design in semiconductor fabrication facilities. *IEEE Transactions on Semiconductor Manufacturing*, 10(3):360–369, 1997.
 - [7] H.D. Sherali, B.M.P. Fraticelli, and R.D. Meller. Enhanced model formulations for optimal facility layout. *Operations Research*, 51(4):629–644, 2003.
 - [8] S.P. Singh and R.R.K. Sharma. Two-level modified simulated annealing based approach for solving facility layout problem. *International Journal of Production Research*, 46(13):3563–3582, 2008.
 - [9] J. Turley. *The Essential Guide to Semiconductors*. Prentice Hall, 2002.
 - [10] W. Xie and N.V. Sahinidis. A branch-and-bound algorithm for the continuous facility layout problem. *Computers & Chemical Engineering*, 32(4):1016–1028, 2008.
 - [11] T. Yang and B.A. Peters. A spine layout design method for semiconductor fabrication facilities containing automated material-handling systems. *International Journal of Operations & Production Management*, 17(5):490–501, 1997.
 - [12] Z. Zhang and C.C. Murray. A corrected formulation for the double row layout problem. *International Journal of Production Research*, 0(0):1–4, 2011.