

An improved mixed-integer programming model for the double row layout of facilities

L. D. Secchin¹ · A. R. S. Amaral² 

Received: 20 May 2017 / Accepted: 17 April 2018 / Published online: 21 April 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract We consider the double row layout problem, which is how to allocate a given number of machines at locations on either side of a corridor so that the total cost to transport materials among these machines is minimized. We propose modifications to a mixed-integer programming model in the literature, obtaining a tighter model. Further, we describe variants of the new model that are even tighter. Computational results show that the new model and its variants perform considerably better than the one in the literature, leading to both fewer enumeration tree nodes and smaller solution times.

Keywords Facility layout · Integer programming · Combinatorial optimization

1 Introduction

The facility layout problem involves positioning facilities in a certain area in order to optimize an objective function. Usually the objective is to minimize the total cost of transporting materials among *facilities* (departments, machines, etc). Facility layout

An earlier version of this paper has appeared in the Proceedings of the Brazilian Symposium on Operations Research (SBPO), 2014 (in Portuguese).

✉ A. R. S. Amaral
amaral@inf.ufes.br

L. D. Secchin
leonardo.secchin@ufes.br

¹ Department of Applied Mathematics (DMA), Universidade Federal do Espírito Santo (UFES), Vitória 29932-540, Brazil

² Graduate School of Computer Science (PPGI), Universidade Federal do Espírito Santo (UFES), Vitória 29075-910, Brazil

problems often occur in practice, for example, in the design of automated manufacturing systems. An automated manufacturing system consists of a set of machines and an automated material handling device that transports materials among machines. A designer should plan a physical configuration of the machines that minimizes the material handling cost. An automated guided vehicle (AGV) is commonly utilized as the material handling device. For greater efficiency, the AGV is set to move along a straight line. In that context, the designer can implement a single row layout, i.e. a layout in which all machines are arranged on the same side of a straight line corridor; Alternatively, a double row layout may be implemented, i.e. a layout in which machines may lie on both sides of a corridor (see [13]). These types of layouts give rise, respectively, to the single row facility layout problem (SRFLP) (e.g. [1, 2, 8, 17]); and the double-row layout problem (DRLP) (e.g. [6, 11]).

The DRLP is how to arrange a given number of machines on either side of the corridor so that the total cost to transport materials among these machines is minimized. Chung and Tanchoco [11] described a practical application of the DRLP in an LCD manufacturing line.

The DRLP is an NP-Hard combinatorial optimization problem (e.g. [6]), and therefore, as the number of machines to be arranged increases, the computational cost to solve the problem increases considerably. In recent years, several authors in the literature have applied integer programming to different facility layout problems to exactly solve medium sized problems (e.g. [3, 7, 9, 10, 12, 14–16]). Specifically for the DRLP, a mixed-integer programming (MIP) model was presented by Chung and Tanchoco [11]. Amaral [6] proposed a different MIP model for the DRLP, which proved to be more efficient than the model of Chung and Tanchoco [11]. In this paper, we propose modifications to the MIP model of Amaral [6], obtaining a tighter model. The new model leads to significant reduction in computer time and in the number of branch-and-bound nodes needed to solve problems.

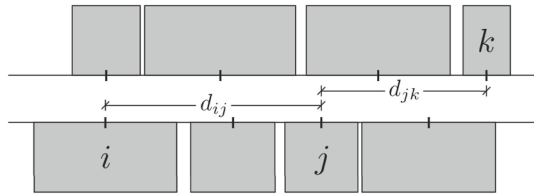
2 The model of Amaral [6]

The mixed-integer programming (MIP) model of Amaral [6] considers the following definitions.

2.1 Parameters

n	Number of machines
$N = \{1, \dots, n\}$	Set of machines
$R = \{\text{upper row, lower row}\}$	Set of rows that define a corridor
c_{ij}	Amount of flow between machines i e j ($1 \leq i < j \leq n$)
ℓ_i	Length of machine i
$L = \sum_{i=1}^n \ell_i$	

Fig. 1 A double row layout. The distance d_{ij} between two machines i and j is the horizontal distance between their centers



It is assumed that the corridor lies along the x axis in the range $[0, L]$, the width of the corridor is neglectable and the distance between two machines is taken between their centers (see Fig. 1).

2.2 Variables

x_i	Abscissa of the center of machine i ($1 \leq i \leq n$)
d_{ij}	Distance between machines i e j ($1 \leq i < j \leq n$)
α_{ij}	Binary variable that equals 1 if machine i is on the left of machine j , both on the same row; and 0 otherwise ($1 \leq i, j \leq n, i \neq j$)

Amaral [6] proposes the following MIP formulation for the double row layout problem:

$$M0: \min \sum_{(i,j):i<j} c_{ij}d_{ij} \tag{1}$$

$$d_{ij} \geq x_i - x_j, \quad i < j \tag{2}$$

$$d_{ij} \geq x_j - x_i, \quad i < j \tag{3}$$

$$d_{ij} - \alpha_{ij}(\ell_i + \ell_j)/2 - \alpha_{ji}(\ell_i + \ell_j)/2 \geq 0, \quad i < j \tag{4}$$

$$x_i + (\ell_i + \ell_j)/2 \leq x_j + L(1 - \alpha_{ij}), \quad i \neq j \tag{5}$$

$$x_p \leq x_q, \quad (p, q) = \arg \min_{i<j} c_{ij} \tag{6}$$

$$- \alpha_{ij} - \alpha_{ji} + \alpha_{ik} + \alpha_{ki} + \alpha_{jk} + \alpha_{kj} \leq 1, \quad i < j, k \neq i, k \neq j \tag{7}$$

$$- \alpha_{ij} + \alpha_{ji} + \alpha_{ik} - \alpha_{ki} - \alpha_{jk} + \alpha_{kj} \leq 1, \quad i < j, k < j, k \neq i \tag{8}$$

$$\alpha_{ij} + \alpha_{ji} + \alpha_{ik} + \alpha_{ki} + \alpha_{jk} + \alpha_{kj} \geq 1, \quad i < j < k \tag{9}$$

$$\alpha_{ij} \in \{0, 1\}, \quad i \neq j \tag{10}$$

$$x_i \in [\ell_i/2, L - \ell_i/2], \quad 1 \leq i \leq n \tag{11}$$

The objective function (1) minimizes the total flow among machines; Constraints (2) and (3) provide the distance between each pair of machines; (4) ensures that if machines i and j are allocated at the same row, then the distance between their centers is at least $(\ell_i + \ell_j)/2$; (5) prohibits overlap between machines; (6) avoids symmetry; (7)–(10) characterize the incidence vectors (α_{ij}) of a DRLP solution.

3 The new model

Based on the model of Amaral [6], we propose to modify Constraint (4) in order to consider the machines allocated between machines i e j . For $i < j, k \neq i, k \neq j$, we define the binary variable e_{kij} such that $e_{kij} = 1$ if machine k is between machines i and j , all at the same row; and 0 otherwise. These types of variables have been used before in [4]. Note that if there is a set $S_{ij} \subset N$ of machines allocated between machines i and j , all of them at the same row, then we can narrow the right hand side of constraint (4) to $\sum_{k \in S_{ij}} \ell_k$. Further, $d_{ij} > (\ell_i + \ell_j)/2$, when $S_{i,j} \neq \emptyset$ so that restriction (5) can also be reformulated by replacing $(\ell_i + \ell_j)/2$, the lower bound for $|x_i - x_j|$, by the tighter bound d_{ij} . Thus, we obtain the following model:

$$\begin{aligned}
 \text{M: } \min \quad & \sum_{(i,j):i < j} c_{ij}d_{ij} \\
 & (2), (3), (6)-(11) \\
 & \sum_{k;k \neq i,j} \ell_k e_{kij} \leq d_{ij} - \alpha_{ij}(\ell_i + \ell_j)/2 - \alpha_{ji}(\ell_i + \ell_j)/2, \quad i < j \quad (12) \\
 & x_i + d_{ij} \leq x_j + 2(L - \ell_i/2 - \ell_j/2)(1 - \alpha_{ij}), \quad i < j \quad (13) \\
 & x_i + d_{ji} \leq x_j + 2(L - \ell_i/2 - \ell_j/2)(1 - \alpha_{ij}), \quad i > j \quad (14) \\
 & e_{kij} \geq \alpha_{ik} + \alpha_{kj} - 1, \quad i < j, k \neq i, k \neq j \quad (15) \\
 & e_{kij} \geq \alpha_{jk} + \alpha_{ki} - 1, \quad i < j, k \neq i, k \neq j \quad (16) \\
 & e_{kij} \in \{0, 1\}, \quad i < j, k \neq i, k \neq j \quad (17)
 \end{aligned}$$

Note that Constraint (12) is the reformulation of Constraint (4). It is easy to see that $d_{ij} - 2(L - \ell_i/2 - \ell_j/2) \leq x_j - x_i$ is always valid, implying validity of constraints (13) and (14), which represent the reformulation of constraint (5). The constraints (15)–(17) together with the objective impose that the variables e_{kij} assume only values according to their definition. In Model M0 there are $\frac{n}{2}(2n^2 - n - 1) + 1$ constraints, $\frac{n}{2}(n - 1) + n$ continuous variables and $n(n - 1)$ binary variables. In the proposed Model (M), there are $n(n^2 - 3n + 2)$ additional restrictions, and $\frac{n}{2}(n^2 - 3n + 2)$ additional binary variables. However, the reformulation (12), (13) and (14) of the constraints (4) and (5) contribute to the gain of performance in relation to Model M0.

3.1 Valid inequalities

As inter-machine distance is a semi-metric, the following inequalities are valid:

$$d_{ij} \leq d_{ik} + d_{jk}, \quad d_{ik} \leq d_{ij} + d_{jk}, \quad d_{jk} \leq d_{ij} + d_{ik} \quad (18)$$

The variable e_{kij} is zero if machines k, i , and j are not all on the same side of the corridor. The following inequalities are, therefore, valid:

$$e_{kij} \leq \alpha_{ik} + \alpha_{ki}, \quad e_{kij} \leq \alpha_{ij} + \alpha_{ji}, \quad e_{kij} \leq \alpha_{jk} + \alpha_{kj} \quad (19)$$

Table 1 Comparison of times (s) required to solve problems using Models M0T, M0, M, MT, MC, MTC

Reference	Instance	Time (s)					
		M0T	M0 [6]	M	MT	MC	MTC
Simmons [17]	S9	5.75	2.80	1.94	2.42	1.81	2.47
	S9H	52.98	36.80	28.81	19.41	19.81	35.62
	S10	16.48	14.72	17.25	17.41	11.14	17.21
	S11	150.44	96.94	72.76	76.49	72.22	66.27
Amaral [5]	12a	237.20	207.30	152.78	131.99	125.04	133.13
	12b	632.97	283.66	178.49	182.07	123.58	143.58
	13a	2,532.69	1,562.53	694.26	988.08	648.57	617.03
	13b	2,037.00	984.78	449.23	377.06	440.93	355.50
This paper	14a	35,816.78	22,389.72	7,115.91	3,929.42	6,404.35	4,286.17
	14b	26,830.52	9,151.56	7,768.00	7,936.44	7,355.03	9,299.20
Amaral [1]	P15	130,932.25	89,923.20	53,131.25	15,585.71	24,651.60	32,177.24

4 Computational experiments

All tests were performed using CPLEX 12.7.1.0 on an Intel Core i7-3770 CPU 3.40 GHz with 8 GB of RAM with the Windows 8 operating system. The problem instances tested include instances (S9, S9H, S10, S11) introduced by Simmons [17], instances (12a, 12b, 13a, 13b) of Amaral [5] and the instance P15 of Amaral [1]. Besides, there are two new instances (14a, 14b), which are available from the authors. For all of the instances tested here, we assume that clearances have already been included in the lengths of the machines.

In order to analyze the effects of inequalities (18) and (19), we tested Model M and its following variants:

- MC: Model M with the inequalities (18) used as cuts, inserted on demand;
- MT: Model M with the inequalities (18) inserted as constraints;
- MTC: Model MT with the inequalities (19) used as cuts.

These are compared against models:

- M0: the model, which was given in [6];
- M0T: Model M0 with the inequalities (18) inserted as constraints.

Computational results are presented in Tables 1 and 2. For each problem instance, the first two columns of Table 1 specify the problem instance. The next six columns show a comparison of each model (M0T, M0, M, MC, MT, MTC) in terms of time required to solve the problems. Table 2 displays for each model (M0T, M0, M, MC, MT, MTC), the number of enumeration tree nodes consumed to solve each instance to optimality. All tests ended with optimality gap equal to zero.

From Table 1 it can be seen that for every instance the solution process with Model M is often much faster than with Model M0. Moreover, each of the models MT, MC or MTC provides some performance improvement over Model M and hence, over Model

Table 2 Comparison of number of nodes required to solve problems using Models M0T, M0, M, MT, MC, MTC

Reference	Instance	Number of nodes					
		M0T	M0 [6]	M	MT	MC	MTC
Simmons [17]	S9	20,733	18,631	6,251	4,083	4,919	4,059
	S9H	391,004	430,581	143,394	44,710	72,030	91,369
	S10	37,200	94,010	27,194	11,481	13,699	11,940
	S11	395,103	589,937	124,742	50,686	87,484	40,860
Amaral [5]	12a	463,950	839,329	131,937	43,234	72,793	41,988
	12b	595,913	1,095,778	149,957	69,734	67,043	46,149
	13a	4,268,227	5,322,187	541,996	199,717	247,128	112,350
	13b	2,372,332	3,443,922	245,466	77,708	175,468	71,176
This paper	14a	23,233,242	33,372,367	3,637,525	584,340	1,373,130	517,542
	14b	18,360,617	19,022,364	3,777,966	1,446,657	1,542,187	1,386,147
Amaral [1]	P15	56,973,552	86,151,296	13,349,760	933,887	3,085,948	3,056,060

M0. Among the six models shown in the table, the smallest processing time for each instance was obtained either with MTC, MT or MC.

The solution times for Models MC and MT indicate that for the larger instances it may be better to insert the triangular inequalities (18) as constraints in Model M. Note that MT was specially fast for the largest instance (P15). However, the inclusion of Inequalities (18) in M0 does not help with speed up the solution process, as M0T is the slowest model for every instance, but S10.

Regarding the use of cuts (19) in MT or MTC, it has been observed that MT was particularly fast for solving Instance P15. However, for the instances with $n < 15$, there is a similar time performance between MTC and MT.

From Table 2 it is seen that for all instances the number of nodes consumed with the models M, MC, MT or MTC is smaller than with M0, which indicates that the proposed models are tighter than M0. Particularly, with MT or MTC, the number of nodes has been drastically reduced in relation to M0, for all instances. Therefore, with the presented models we can obtain optimal solutions for larger instances (with 14 and 15 departments) more efficiently than before.

5 Conclusions

In this article, we considered the double row layout problem (DRLP), which has large applicability, particularly in the design of manufacturing systems. We proposed modifications to the mixed-integer programming formulation of the DRLP given by Amaral [6] (denoted here as M0), thus obtaining a model M, which is more efficient than M0. Moreover, we described three variants of the model M, which were called: MC, MT and MTC. These variants are tighter than M, and hence tighter than M0. Computational tests with the proposed model M and its variants have shown that they

perform considerably better than M0 leading to both smaller solution times and smaller number of nodes.

We believe that improvements on the proposed formulation are possible. In future works, valid inequalities other than those used here can be applied to the proposed model leading to a further performance gain.

Acknowledgements The second author was supported by FAP/UFES and CAPES (Grant Number 99999.002643/2015-04).

References

1. Amaral, A.R.S.: On the exact solution of a facility layout problem. *Eur. J. Oper. Res.* **173**(2), 508–518 (2006). <https://doi.org/10.1016/j.ejor.2004.12.021>
2. Amaral, A.R.S.: An exact approach to the one-dimensional facility layout problem. *Oper. Res.* **56**(4), 1026–1033 (2008). <https://doi.org/10.1287/opre.1080.0548>
3. Amaral, A.R.S.: A mixed 0–1 linear programming formulation for the exact solution of the minimum linear arrangement problem. *Optim. Lett.* **3**(4), 513–520 (2009a). <https://doi.org/10.1007/s11590-009-0130-0>
4. Amaral, A.R.S.: A new lower bound for the single row facility layout problem. *Discrete Appl. Math.* **157**(1), 183–190 (2009). <https://doi.org/10.1016/j.dam.2008.06.002>
5. Amaral, A.R.S.: The corridor allocation problem. *Comput. Oper. Res.* **39**(12), 3325–3330 (2012). <https://doi.org/10.1016/j.cor.2012.04.016>
6. Amaral, A.R.S.: Optimal solutions for the double row layout problem. *Optim. Lett.* **7**, 407–413 (2013a). <https://doi.org/10.1007/s11590-011-0426-8>
7. Amaral, A.R.S.: A parallel ordering problem in facilities layout. *Comput. Oper. Res.* **40**(12), 2930–2939 (2013). <https://doi.org/10.1016/j.cor.2013.07.003>
8. Amaral, A.R.S., Letchford, A.N.: A polyhedral approach to the single row facility layout problem. *Math. Program.* **141**(1–2), 453–477 (2013). <https://doi.org/10.1007/s10107-012-0533-z>
9. Bukchin, Y., Tzur, M.: A new milp approach for the facility process–layout design problem with rectangular and l/t shape departments. *Int. J. Prod. Res.* **52**(24), 7339–7359 (2014). <https://doi.org/10.1080/00207543.2014.930534>
10. Chae, J., Regan, A.C.: Layout design problems with heterogeneous area constraints. *Comput. Ind. Eng.* **102**, 198–207 (2016). <https://doi.org/10.1016/j.cie.2016.10.016>
11. Chung, J., Tanchoco, J.M.A.: The double row layout problem. *Int. J. Prod. Res.* **48**(3), 709–727 (2010). <https://doi.org/10.1080/00207540802192126>
12. Hathhorn, J., Sisikoglu, E., Sir, M.Y.: A multi-objective mixed-integer programming model for a multi-floor facility layout. *Int. J. Prod. Res.* **51**(14), 4223–4239 (2013). <https://doi.org/10.1080/00207543.2012.753486>
13. Heragu, S.S., Kusiak, A.: Machine layout problem in flexible manufacturing systems. *Oper. Res.* **36**(2), 258–268 (1988). <https://doi.org/10.1287/opre.36.2.258>
14. Javadi, B., Jolai, F., Slomp, J., Rabbani, M., Tavakkoli-Moghaddam, R.: An integrated approach for the cell formation and layout design in cellular manufacturing systems. *Int. J. Prod. Res.* **51**(20), 6017–6044 (2013). <https://doi.org/10.1080/00207543.2013.791755>
15. Klausnitzer, A., Lasch, R.: Extended Model Formulation of the Facility Layout Problem with Aisle Structure, pp. 89–101. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-20863-3_7
16. Niroomand, S., Vizvári, B.: A mixed integer linear programming formulation of closed loop layout with exact distances. *J. Ind. Prod. Eng.* **30**(3), 190–201 (2013). <https://doi.org/10.1080/21681015.2013.805699>
17. Simmons, D.M.: One-dimensional space allocation: an ordering algorithm. *Oper. Res.* **17**(5), 812–826 (1969). <https://doi.org/10.1287/opre.17.5.812>