Journal of the Operational Research Society

ISSN: 0160-5682 (Print) 1476-9360 (Online) Journal homepage: https://www.tandfonline.com/loi/tjor20

An improved model for the parallel row ordering problem

Xuhong Yang, Wenming Cheng, Alice E. Smith & André R. S. Amaral

To cite this article: Xuhong Yang, Wenming Cheng, Alice E. Smith & André R. S. Amaral (2020) An improved model for the parallel row ordering problem, Journal of the Operational Research Society, 71:3, 475-490, DOI: 10.1080/01605682.2018.1556570

To link to this article: https://doi.org/10.1080/01605682.2018.1556570

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
2020, VOL. 71, NO. 3, 475–490
https://doi.org/10.1080/01605682.2018.1556570

ORIGINAL ARTICLE

An improved model for the parallel row ordering problem

Xuhong Yang[a], Wenming Cheng[a], Alice E. Smith[b] and André R. S. Amaral[c]

[a]School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China; [b]Department of Industrial & Systems Engineering, Auburn University, Auburn, Alabama, USA; [c]Graduate School of Computer Science, Federal University of Espirito Santo (UFES), Vitória, ES, Brazil

ABSTRACT
This article studies the parallel row ordering problem (PROP), which is NP-hard. The PROP is interesting from both a theoretical and practical point of view. A new mixed-integer programming model for this problem is proposed, which presents a superior performance to that of a former mixed-integer programming model proposed for the problem. With the new model, several problem instances taken from the literature or randomly generated were efficiently solved to optimality. Moreover, it is now possible to efficiently solve problem instances of larger sizes.

ARTICLE HISTORY
Received 17 April 2018
Accepted 3 December 2018

KEYWORDS
Parallel row ordering problem; facility layout; mixed-integer programming

1. Introduction

Determining the physical organisation of facilities in a plant is defined to be the facility layout problem (FLP). Where to locate departments (the optimal arrangement) in a given layout and the efficient design of those departments are significant and elementary strategic issues facing any manufacturing industry. From 20 to 50 percent of the total operating expenses in manufacturing are attributed to materials handling costs (Tompkins, White, Bozer, & Tanchoco, 2010). Moreover, effective facilities planning could reduce these costs by 10 to 30 percent annually.

The FLP has been studied for decades. Various formulations of the different types of FLPs and the solution algorithms to this problem are presented in some relevant surveys (Anjos & Vieira, 2017; Drira, Pierreval, & Hajri-Gabouj, 2007; Hosseini-Nasab, Fereidouni, Ghomi, & Fakhrzad, 2018; Meller & Gau, 1996). One of the most important FLPs is the *Row Layout Problem* (RLP), which has drawn much attention over the years. RLPs includes problems such as the *Single Row Facility Layout Problem* (SRFLP) (Simmons, 1969), the *Double Row Layout Problem* (DRLP) (Chung & Tanchoco, 2010), the *Corridor Allocation Problem* (CAP) (Amaral, 2012), the *Multi-Row Layout Problem* (MRLP) (Gen, Ida, & Cheng, 1995), the *Checkpoint Ordering Problem* (COP) (Hungerländer, 2017), and the *k-Parallel Row Ordering Problem* (k-PROP) (Amaral, 2013b). It should be noted that these problems are NP-hard in general (Garey & Johnson, 1979), thus solving them to global optimality in reasonable time is generally difficult. k-PROP is an extension of SRFLP that considers arrangements of the n rectangular departments along more than one row (Amaral, 2013b). Given a set $N = \{1, \ldots, n\}$ of departments; the length l_i of each department $i \in N$; and the average daily traffic f_{ij} between departments i and j, $(i, j \in N, i<j)$; let $\{N_i\}_{i=1,\ldots,k}$ be a partition of N such that $\cup_{i=1}^{k} N_i = N$ and $N_i \cap N_j = \emptyset (1 \leq i<j \leq k)$. Let $R = \{1, \ldots, k\}$ be a set of rows all parallel to the x-axis. We are given a one-to-one assignment of the set $\{N_i\}_{i=1,\ldots,k}$ to the set R, so that the departments pertaining to the subset N_i (for some i, $1 \leq i \leq k$) should be arranged along row r (for some r, $1 \leq r \leq k$). A k-PROP layout should respect two main conditions: (i) No space is allowed between two adjacent departments, and (ii) the left-most point of the arrangement on each row should have zero abscissa. In addition, the departments have their lengths placed on a row along the x-axis direction. The distance between two departments is assumed to be the rectilinear distance between their centres. As the ordering of departments occurs in one dimension (x), the other components of the distance do not change between any two departments. That means we only consider the x-distances between departments. The objective of k-PROP is to find a layout that minimises the total flow cost over all possible layouts.

The k-PROP is important from a practical point of view. The k-PROP has application in multi-floor building design, where the orderings occur along the horizontal direction and, thus, only horizontal distances between departments need to be decided. Other components of the distance remain constant. Another application of k-PROP is the arrangement of departments along two or more parallel straight

CONTACT André R. S. Amaral amaral@inf.ufes.br Graduate School of Computer Science, UFES, Vitória, ES 29060-900, Brazil

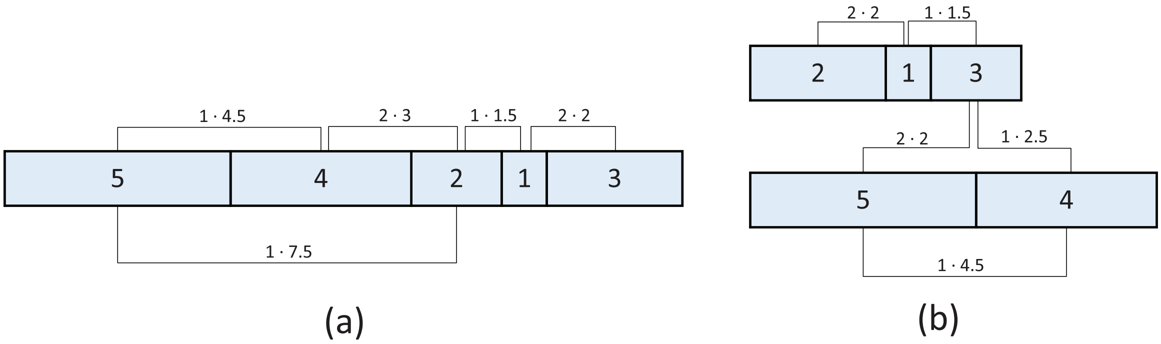

Figure 1. Arrangement of departments along two parallel straight rows on a floor plan (adapted from Amaral, 2013b).

rows on a floor plan, as shown in Figure 1, which is adapted from Amaral (2013b). In Figure 1, the distance between the greyed departments is shown. Only the x-distances between departments will have to be determined, as the ordering of departments occurs in x.

In this paper, we will focus on a new model for the special case of the k-PROP with $k = 2$, which is simply named PROP (Amaral, 2013b). The objective of PROP is to order the departments in the two rows in order to minimise a cost function of the x-distances between departments. In PROP the t departments $\{1, \ldots, t\}$ are restricted to one row (say, Row 1), while the other $\{t + 1, \ldots, n\}$ departments are restricted to a parallel row (say, Row 2). The subset of departments placed on a row does not change (just their ordering changes). Therefore, PROP seeks an ordering π^1 of the t departments at Row 1 and an ordering π^2 of the $(n - t)$ departments at Row 2.

We use a small example to show the differences between SRFLP and PROP. Consider five departments with lengths $l_1 = 1, l_2 = 2,\ l_3 = 3, l_4 = 4, l_5 = 5$ and traffic intensities $f_{12} = f_{25} = f_{45} = 1,\ f_{13} = f_{24} = 2$. Assume $t = 3$. Figure 2 displays the optimal layouts and the corresponding cost for the two problems:

- In (a) we depict an optimal layout for SRFLP. The corresponding cost is $1 \cdot 1.5 + 2 \cdot 2 + 2 \cdot 3 + 1 \cdot 7.5 + 1 \cdot 4.5 = 23.5$.
- In (b) we display an optimal layout for PROP with departments 1, 2, and 3 assigned to row 1 and departments 4 and 5 assigned to row 2. The associated cost is $1 \cdot 1.5 + 2 \cdot 2 + 2 \cdot 2 + 1 \cdot 2.5 + 1 \cdot 4.5 = 16.5$.

Obviously, the PROP reduces to the SRFLP if $t = n$. The Parallel Row Ordering Problem is NP-hard (Amaral, 2013b; Garey & Johnson, 1979), so an efficient model for this problem is of interest. In this paper, a more effective *mixed-integer programming* (MIP) model is proposed for the PROP, with which we can efficiently solve larger instances than those with a previously published mixed-integer programming model in (Amaral, 2013b) using off-the-shelf solvers such as Gurobi. We assume $n \geq 3$ throughout the paper.

The remainder of this paper is organised as follows. In the next section, we briefly review the literature on related row layout problems. A new mixed integer programming model for the PROP is provided in Section 3. In Section 4, we report the results of computational studies. Finally, conclusions and some suggestions for future research are made in Section 5.

2. Literature review

Research on FLPs is extensive. Among these problems, the SRFLP, the MRLP, the DRLP, the CAP, and the COP are row layout problems (RLP) and, thus, are the most related to PROP in the literature.

RLPs have a number of practical applications, such as supermarket layout (Simmons, 1969), campus planning (Dickey & Hopkins, 1972), typewriter keyboard design (Pollatschek, Gershoni, & Radday, 1976), balancing hydraulic turbine runners (Laporte & Mercure, 1988), the layout of machines in an automated manufacturing system (Heragu & Kusiak, 1991), numerical analysis (Brusco & Stahl, 2000), optimal digital signal processor memory layout (Wess & Zeitlhofer, 2004), room arrangements at office buildings or hospitals (Amaral, 2012), and semiconductor manufacturing (Zuo, Murray, & Smith, 2014).

The SRFLP, also known as the *one-dimensional space allocation problem* (ODSAP), was first formulated by Simmons (1969). The SRFLP consists of finding the most efficient arrangement of a given number of departments along one side of the corridor. One special case of the SRFLP is the *minimum linear arrangement problem* (MinLA) (Adolphson & Hu, 1973; Amaral, 2009a), where every department has unit length. It is known that the MinLA is strongly NP-hard (Garey & Johnson, 1979), so it follows that the SRFLP is strongly NP-hard. The SRFLP arranges departments in one straight row. To solve the SRFLP, some exact approaches have

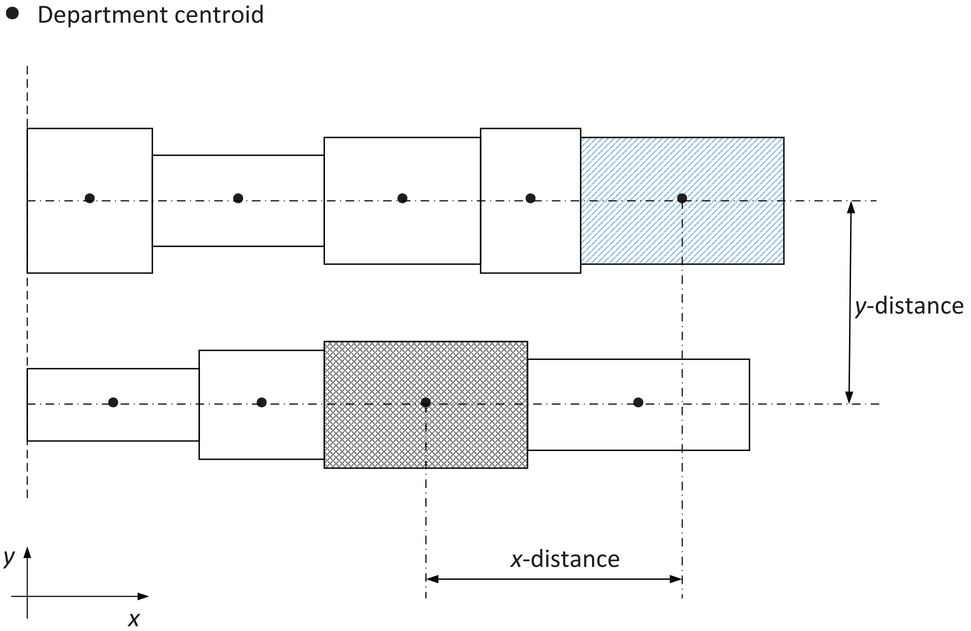

Figure 2. (a) Description of the SRFLP optimal layout; (b) Description of the PROP optimal layout.

been proposed. Simmons (1969) presented a branch and bound algorithm characterised by a stepwise build-up of feasible orderings. Some mixed-integer programming models were also proposed to solve the SRFLP. Love and Wong (1976) formulated the problem as a mixed-integer linear programming model. Amaral (2006, 2008) proposed two improved models. Amaral (2009b) proposed a cutting plane algorithm which solved to optimality instance with size up to 33. After deriving several huge classes of valid inequalities, and giving conditions for them to induce facets, Amaral and Letchford (2013) presented a full branch-and-cut algorithm, which solved instances with $n \leq 30$ to proven optimality.

Another research line with respect to the exact approach is *semidefinite programming* (SDP). Anjos, Kennings, and Vannelli (2005) used this method to give a lower bound on the SRFLP together with a heuristic procedure that obtains a feasible solution from the SDP relaxation. Anjos and Vannelli (2008) showed that a combination of a semi-definite programming relaxation with cutting planes can handle instances with up to 30 departments. Hungerländer and Rendl (2013) conducted a comprehensive comparison of the different modelling approaches for SRFLP and applied a SDP approach for this problem, where optimal solutions for instances with up to 42 departments were obtained. Recently, the work of Andrade and Ferreira (2017) improved the model of the SRFLP in (Amaral, 2006) by changing two constraints, which compute distances. The model in (Andrade & Ferreira, 2017) presented a better performance than the models in (Amaral, 2006; Love & Wong, 1976) in terms of solution quality and computational time.

The majority of the solution approaches for the SRFLP are heuristic or meta-heuristic algorithms (e.g., Keller, 2017; Ning & Li, 2017), which generate feasible solutions in a fast way and could solve instances with a larger number of departments. The reader may refer to Kothari and Ghosh (2012) and Keller and Buscher (2015) for reviews with respect to the different models and methods for the SRFLP.

Since the early 1960s, several studies on the MRLP have been carried out. The MRLP is concerned with the placement of departments in different rows such that the total transportation cost between departments is minimised. Koopmans and Beckmann (1957) presented a *quadratic assignment problem* (QAP) model for the MRLP. After that, some researchers developed linearizations of the QAP and solved it using optimal algorithms (e.g., Zhou, Love, Teo, & Luo, 2017). Singh and Sharma (2008) formulated the MRLP as a QAP, where machines are separated by minimum clearances. A survey may be found about models and algorithms for the MRLP by Zhou et al. (2017).

The DRLP, in which departments are placed in both sides of the corridor, was first discussed by Heragu and Kusiak (1988). It was first formulated as *mixed-integer linear programming* (MILP) model by Chung and Tanchoco (2010), who developed five heuristic algorithms to provide a reasonable initial solution and corresponding upper bound for the DRLP. Zhang and Murray (2012) provided corrections for the model in Chung and Tanchoco (2010). Amaral (2013a) proposed a new model of the DRLP, which performs better than the formulation proposed by Chung and Tanchoco (2010). Recently, two DRLP models have been presented. One developed by Amaral (2018) and the other by Secchin and Amaral (2018). The CAP was proposed in Amaral (2012). The CAP is a combinatorial problem, as departments are placed without space between them. The PROP resembles the DRLP and the CAP for they display departments along two rows. Usually, in the PROP, DRLP, and CAP, the distance between the two parallel rows (i.e., the corridor width) is assumed to be zero. In contrast to PROP, departments arranged along two parallel rows are not restricted to any row in DRLP or CAP. PROP and CAP assume that no space is permitted between two adjacent departments and that the departments arrangement in both rows start from a common point (i.e., zero abscissa), while in the DRLP these two conditions need not be enforced.

Although there are a great number of papers on row layout problems, to the best of our knowledge, there are only three published papers addressing the PROP. Amaral (2013b) presents the PROP and proposes a mixed-integer linear programming model, which extends a MILP model of the SRFLP and can solve instances of the PROP with up to 23 departments to optimality. The instances presented in Amaral (2013b) have been commonly used as the benchmark for the SRFLP to perform computational experiments. Using the optimisation solver CPLEX 12.4, the proposed MIP solves all instances to optimality. Maadi, Javidnia, and Jamshidi (2017) solve instances with sizes ranging from 30 to 70 departments using two strategies based on genetic algorithm (GA) and a novel population based simulated annealing algorithm (PSA). Computational results show that PSA has a superior performance in terms of objective value and computational time as the sizes of test instances increase. Hungerländer (2014) develops a SDP approach for the *k*-PROP, which extends an SDP approach for the SRFLP, by modelling inter-row distances as products of ordering variables. The proposed semidefinite programming approach yields tight global bounds for instances with size up to 23 departments within a few minutes. Moreover, for the instances with up to 100 departments the algorithm could yield reasonable global bounds.

3. The PROP mathematical model

The mathematical model presented here for the PROP is based on that in Andrade and Ferreira (2017), which in turn is based on the mathematical model for the SRFLP of Amaral (2006). The parameters and decision variables in this paper are defined in Table 1.

Inspired by the work of Andrade and Ferreira (2017), in order to improve the model, we modify the four distances constraints in Amaral (2013b) using a similar approach when two departments lie in the same row. The model of Amaral (2013b) for the PROP is called PROP1. Next, we define some polytopes, which are the same as in Amaral (2013b).

Table 1. Parameters and decision variables.

Parameters	
n	Number of departments
t	Number of departments in the first row
N_1	The set of departments in the first row; $N_1 = \{1, \ldots, t\}$
N_2	The set of departments in the second row; $N_2 = \{t+1, \ldots, n\}$
N	The set of all departments; $N = N_1 \cup N_2$
i, j, k	Department indices
l_i	The length of department i; $i \in N$
f_{ij}	The traffic intensity between departments i and j
Variables	
α^1	A vector $\alpha^1 = (\alpha^1_{ij})_{i,j \in N_1; i<j}$ such that $\alpha^1_{ij} = 1$ when department i is to the left of department j and $\alpha^1_{ij} = 0$ otherwise.
α^2	A vector $\alpha^2 = (\alpha^2_{ij})_{i,j \in N_2; i<j}$ such that $\alpha^2_{ij} = 1$ when department i is to the left of department j and $\alpha^2_{ij} = 0$ otherwise.
d^1	A vector $d^1 = (d^1_{ij})_{i,j \in N_1; i<j}$ such that d^1_{ij} stands for the horizontal-distance between the centres of two departments placed at Row 1.
d^2	A vector $d^2 = (d^2_{ij})_{i,j \in N_2; i<j}$ such that d^2_{ij} stands for the horizontal-distance between the centres of two departments placed at Row 2.
d^{12}	A vector $d^{12} = (d^{12}_{ij})_{i,j \in N_2; i<j}$ such that d^{12}_{ij} stands for the horizontal-distance between the centres of a department placed at Row 1 and one placed at Row 2.

$$H^1_t = \left\{ \alpha^1 \in \Re_+^{\binom{t}{2}} : \alpha^1_{ij} + \alpha^1_{jk} - \alpha^1_{ik} \le 1 \quad (i, j, k \in N_1; i<j<k) \right. \tag{1}$$

$$\left. -\alpha^1_{ij} - \alpha^1_{jk} + \alpha^1_{ik} \le 0 \quad (i, j, k \in N_1; i<j<k) \right\} \tag{2}$$

$$H^2_{n-t} = \left\{ \alpha^2 \in \Re_+^{\binom{n-t}{2}} : \alpha^2_{ij} + \alpha^2_{jk} - \alpha^2_{ik} \le 1 \quad (i, j, k \in N_2; i<j<k) \right. \tag{3}$$

$$\left. -\alpha^2_{ij} - \alpha^2_{jk} + \alpha^2_{ik} \le 0 \quad (i, j, k \in N_2; i<j<k) \right\} \tag{4}$$

$$D^1_t = \left\{ d^1 \in \Re_+^{\binom{t}{2}} : d^1_{ij} \ge \left(l_i + l_j\right)/2 \quad (i, j, k \in N_1; i<j<k) \right\} \tag{5}$$

$$D^2_{n-t} = \left\{ d^2 \in \Re_+^{\binom{n-t}{2}} : d^2_{ij} \ge \left(l_i + l_j\right)/2 \quad (i, j, k \in N_2; i<j<k) \right\} \tag{6}$$

Then, a mixed integer programming model of the PROP is as follows:

Objective function:

$$\text{Minimize} \sum_{i,j \in N_1; i<j} f_{ij} d^1_{ij} + \sum_{i,j \in N_2; i<j} f_{ij} d^2_{ij} + \sum_{i \in N_1, j \in N_2} f_{ij} d^{12}_{ij} \tag{7}$$

Subject to:

$$\begin{aligned} d^1_{ij} \ge \left(l_i + l_j\right)/2 + \sum_{1 \le k<i} l_k \left(1 - \alpha^1_{ki}\right) + \sum_{i<k<j} l_k \alpha^1_{ik} + \sum_{j<k \le t} l_k \alpha^1_{ik} \\ - \sum_{1 \le k<i} l_k \left(1 - \alpha^1_{kj}\right) - \sum_{i<k<j} l_k \left(1 - \alpha^1_{kj}\right) - \sum_{j<k \le t} l_k \alpha^1_{jk} \quad (i, j \in N_1; i<j) \end{aligned} \tag{8}$$

$$d^1_{ij} \geq (l_i + l_j)/2 - \sum_{1\leq k<i} l_k\left(1-\alpha^1_{ki}\right) - \sum_{i<k<j} l_k\alpha^1_{ik} - \sum_{j<k\leq t} l_k\alpha^1_{ik} + \sum_{1\leq k<i} l_k\left(1-\alpha^1_{kj}\right) + \sum_{i<k<j} l_k\left(1-\alpha^1_{kj}\right) + \sum_{j<k\leq t} l_k\alpha^1_{jk} \quad (i,j\in N_1; i<j) \tag{9}$$

$$d_1 \in D^1_t \tag{10}$$

$$\alpha_1 \in H^1_t \tag{11}$$

$$\alpha^1_{ij} \in \{0,1\} \quad (i,j\in N_1; i<j) \tag{12}$$

$$d^2_{ij} \geq (l_i + l_j)/2 + \sum_{t<k<n} l_k\left(1-\alpha^2_{ki}\right) + \sum_{i<k<j} l_k\alpha^2_{ik} + \sum_{j<k\leq n} l_k\alpha^2_{ik} - \sum_{t<k<i} l_k\left(1-\alpha^2_{kj}\right) - \sum_{i<k<j} l_k\left(1-\alpha^2_{kj}\right) - \sum_{j<k\leq n} l_k\alpha^2_{jk} \quad (i,j\in N_2; i<j) \tag{13}$$

$$d^2_{ij} \geq (l_i + l_j)/2 - \sum_{t<k<n} l_k\left(1-\alpha^2_{ki}\right) - \sum_{i<k<j} l_k\alpha^2_{ik} - \sum_{j<k\leq n} l_k\alpha^2_{ik} + \sum_{t<k<i} l_k\left(1-\alpha^2_{kj}\right) + \sum_{i<k<j} l_k\left(1-\alpha^2_{kj}\right) + \sum_{j<k\leq n} l_k\alpha^2_{jk} \quad (i,j\in N_2; i<j) \tag{14}$$

$$d_2 \in D^2_{n-t} \tag{15}$$

$$\alpha_2 \in H^2_{n-t} \tag{16}$$

$$\alpha^2_{ij} \in \{0,1\} \quad (i,j\in N_2; i<j) \tag{17}$$

$$d^{12}_{ij} \geq (l_i - l_j)/2 + \sum_{1\leq k<i} l_k\alpha^1_{ki} + \sum_{i<k\leq t} l_k\left(1-\alpha^1_{ik}\right) - \sum_{t<k<j} l_k\alpha^2_{kj} - \sum_{j<k\leq n} l_k\left(1-\alpha^2_{jk}\right) \quad (i\in N_1; j\in N_2) \tag{18}$$

$$d^{12}_{ij} \geq (l_j - l_i)/2 - \sum_{1\leq k<i} l_k\alpha^1_{ki} - \sum_{i<k\leq t} l_k\left(1-\alpha^1_{ik}\right) + \sum_{t<k<j} l_k\alpha^2_{kj} + \sum_{j<k\leq n} l_k\left(1-\alpha^2_{jk}\right) \quad (i\in N_1; j\in N_2) \tag{19}$$

The objective function (7) minimises the total traffic intensity between departments; Constraints (8) and (9) compute the x-distance between any two departments of Row 1. Constraints (13) and (14) determine the x-distance between pairs of departments of Row 2. The x-distances between two departments from different rows are determined by Constraints (18) and (19). When two departments are placed at the same row, Constraints (10) and (15) reinforce that their minimum x-distance equals the sum of their half-lengths. Constraints (11) and (12) ensure that for Row 1, α^1 is an incidence vector of a linear ordering, while constraint (16) and (17) ensure that for Row 2, α^2 is an incidence vector of a linear ordering.

The PROP formulation (7)–(19) is called PROP2, and is different from PROP1 (Amaral, 2013b) in Constraints (8), (9), (13), and (14). The corresponding constraints in the model of Amaral (2013b) are as follows,

$$d^1_{ij} \geq (l_i - l_j)/2 + \sum_{1\leq k<i} l_k\alpha^1_{ki} + \sum_{i<k\leq t} l_k\left(1-\alpha^1_{ik}\right) - \sum_{1\leq k<j} l_k\alpha^1_{kj} - \sum_{j<k\leq t} l_k\left(1-\alpha^1_{jk}\right) \quad (i,j\in N_1; i<j) \tag{20}$$

$$d^1_{ij} \geq (l_j - l_i)/2 - \sum_{1\leq k<i} l_k\alpha^1_{ki} - \sum_{i<k\leq t} l_k\left(1-\alpha^1_{ik}\right) + \sum_{1\leq k<j} l_k\alpha^1_{kj} + \sum_{j<k\leq t} l_k\left(1-\alpha^1_{jk}\right) \quad (i,j\in N_1; i<j) \tag{21}$$

$$d^2_{ij} \geq (l_i - l_j)/2 + \sum_{t<k<i} l_k\alpha^2_{ki} + \sum_{i<k\leq n} l_k\left(1-\alpha^2_{ik}\right) - \sum_{t<k<j} l_k\alpha^2_{kj} - \sum_{j<k\leq n} l_k\left(1-\alpha^2_{jk}\right) \quad (i,j\in N_2; i<j) \tag{22}$$

$$d^2_{ij} \geq (l_j - l_i)/2 - \sum_{t<k<i} l_k\alpha^2_{ki} - \sum_{i<k\leq n} l_k\left(1-\alpha^2_{ik}\right) + \sum_{t<k<j} l_k\alpha^2_{kj} + \sum_{j<k\leq n} l_k\left(1-\alpha^2_{jk}\right) \quad (i,j\in N_2; i<j) \tag{23}$$

The two models have the same number of binary variables, continuous variables, and constraints (excluding non-negativity constraints and bounds on variables). The number of binary variables is $n(n-1)/2-t(n-t)$. The number of continuous variables is $n(n-1)/2$. The number of constraints is $n(n-1)(n+1)/2-t(n-t)(n-2)$. We can easily prove that the number of binary variables and constraints in the model is the smallest when $t = n/2$ if n is even; and $t = n/2+1$ or $t = n/2-1$ if n is odd.

4. Computational results

The computational experiments for the mixed integer programming models (PROP1 and PROP2) were conducted with Gurobi (Gurobi Optimization, 2015) version 7.5.1 on an Intel(R) Core (TM) i3-4130 CPU @ 3.40 GHz machine (in 64-bit mode) with 4 GB memory running the Windows 10 operating system and were implemented using the modelling language YALMIP (Löfberg, 2004) in MATLAB 2015a. The purpose of the computational results is to show the improvement of the new model PROP2 compared to the existing one.

Table 2. Experimental comparison of the solution time required by models PROP1 and PROP2 for instances with size $n \leq 23$ and $t = \lfloor n/2 \rfloor$.

			PROP2		PROP1		
Instance	n	Optimal value	T2(s)	BB nodes	T1(s)	BB nodes	T1/T2
S11	11	3895.5	0.24	199	0.57	263	2.41
LW11	11	4668.5	0.19	53	0.55	65	2.93
H12	12	9305.0	0.28	157	0.39	668	1.42
Am12a	12	1583.0	0.25	215	0.39	496	1.55
Am12b	12	1697.5	0.32	401	0.46	855	1.47
Am13a	13	2668.5	0.36	600	1.67	2409	4.65
Am13b	13	3210.0	0.27	337	0.58	781	2.12
H15	15	17,780.0	1.07	1366	4.89	10,495	4.55
Am15	15	3435.0	1.04	1556	2.29	4756	2.20
P16a	16	7630.0	10.39	33,372	22.37	61,850	2.15
P16b	16	6239.5	7.15	21,396	23.01	61,501	3.22
Am17	17	4853.0	4.82	9928	6.51	17,921	1.35
Am18	18	5742.5	12.58	22,205	29.58	60,923	2.35
H20	20	8190.0	228.03	366,883	319.66	586,398	1.40
P20_a	20	12,609.5	1043.39	1,854,851	1317.41	2,215,162	1.26
P20_b	20	12,936.0	780.26	928,369	1095.34	1,366,953	1.40
P21_a	21	7006.5	868.61	1,184,255	1207.40	1,771,741	1.39
P21_b	21	11,705.0	663.49	536,233	3246.25	2,787,329	4.89
P21_c	21	11,434.0	236.32	213,037	448.12	577,356	1.90
P21_d	21	12,289.0	1269.67	1,258,206	4059.18	5,970,919	3.20
P21_e	21	13,112.5	1853.44	1,952,819	2447.70	2,590,114	1.32
P22_a	22	8874.0	1769.20	1,662,475	5843.79	5,156,480	3.30
P22_b	22	15,714.0	707.60	704,088	6450.45	3,389,463	9.12
P22_c	22	14,693.0	4628.71	5,162,377	12,686.13	12,597,557	2.74
P22_d	22	16,355.0	4291.61	3,975,275	21,950.44	11,425,212	5.11
P22_e	22	14,815.5	3760.11	4,226,720	14,497.86	12,827,171	3.86
P23_a	23	10,242.0	2564.93	2,318,728	8952.79	5,286,191	3.49
P23_b	23	15,802.5	3081.16	2,463,071	17,202.89	9,854,623	5.58
P23_c	23	15,542.0	4242.88	4,323,965	16,198.77	11,953,233	3.82
P23_d	23	17,174.0	6572.39	6,411,266	21,031.40	14,722,798	3.20
P23_e	23	16,481.5	3011.07	2,054,404	8918.51	5,137,087	2.96

An instance of the PROP, looks like an SRFLP instance with an additional parameter t, including the following data:

- Positive integer number: n.
- Vector of positive integers: l_i, $(i = 1, \ldots, n)$.
- Matrix of non-negative integers: f_{ij}, $(i = 1, \ldots, n; j = 1, \ldots, n)$.
- Integer number: $t \in [1, n-1]$.

We ran experiments on all instances discussed in Amaral (2013b). In addition, in order to further compare the models, we carried out experiments on some benchmark instances commonly used in the literature, such as LW11 introduced in Love and Wong (1976), Am12a, Am12b, Am13a and Am13b from Amaral (2012), Am17 and Am18 from Amaral (2008), H12, H15 and H20 introduced in Heragu and Kusiak (1991), and N25, N30 from Anjos and Vannelli (2008). Moreover, five new random PROP instances with size $n = 24$ were generated in this paper (data for these instances is available from the authors) according to the rule introduced in Amaral (2013b). We will characterise each random instance generated, which is done in the next following subsections. The total number of PROP instances used for testing in this paper is 184.

4.1. Evaluation of PROP1 and PROP2 on instances with sizes $11 \leq n \leq 23$

At first, we compare the performance of models PROP1 and PROP2 on instances with sizes $11 \leq n \leq 23$ and $t \in \{\lfloor n/2 \rfloor, \lfloor n/3 \rfloor, \lfloor n/4 \rfloor, \lfloor n/5 \rfloor\}$. The computational results obtained are displayed in Tables 2–5.

Table 2 displays a comparison of computational times and number of nodes required with models PROP1 and PROP2 on instances with $t = \lfloor n/2 \rfloor$. For each problem instance, the first three columns report: the name of the instance, the number n of departments, and the optimal value calculated for the instance. The following two columns present results from model PROP2: the computational time spent by Gurobi and the number of branch-and-bound nodes explored when the problem was solved to optimality. The next two columns display the same information for model PROP1. The last column shows the ratio between the time T1 required by PROP1 and the time T2 required by PROP2. Tables 3–5 display similar results for the PROP instances with $t = \lfloor n/2 \rfloor, \lfloor n/3 \rfloor$, $\lfloor n/4 \rfloor, \lfloor n/5 \rfloor$, respectively.

It can be seen that the average time consumed by the solver generally increases by a large amount as n increases. We observe that the time required to

Table 3. Experimental comparison of the solution time required by models PROP1 and PROP2 for instances with size $n \leq 23$ and $t = \lfloor n/3 \rfloor$.

			PROP2		PROP1		
Instance	n	Optimal value	T2(s)	BB nodes	T1(s)	BB nodes	T1/T2
S11	11	5404.5	0.16	86	0.21	235	1.31
LW11	11	6880.5	0.18	52	0.17	193	0.98
H12	12	11,725.0	0.18	103	0.30	565	1.65
Am12a	12	1928.0	0.30	651	0.43	1549	1.42
Am12b	12	2226.5	0.25	275	0.41	947	1.62
Am13a	13	3009.5	0.74	1447	3.01	7314	4.09
Am13b	13	3811.0	0.35	544	1.76	2563	5.04
H15	15	24,110.0	0.48	601	2.10	4436	4.38
Am15	15	3754.0	2.09	2489	4.99	10,826	2.39
P16a	16	9813.0	2.50	12,671	6.51	21,590	2.60
P16b	16	9091.5	1.90	2678	4.33	8760	2.28
Am17	17	5542.0	6.93	49,920	12.52	23,246	1.81
Am18	18	6402.5	24.40	155,917	53.68	74,563	2.20
H20	20	10,398.0	173.67	168,926	388.10	501,416	2.23
P20_a	20	15,874.5	152.45	196,199	363.47	508,943	2.38
P20_b	20	19,167.0	66.78	57,461	160.35	216,084	2.40
P21_a	21	9141.5	107.05	136,368	244.25	358,459	2.28
P21_b	21	13,887.0	277.19	275,751	450.70	629,107	1.63
P21_c	21	12,758.0	2644.37	2,808,386	7012.59	6,476,044	2.65
P21_d	21	14,988.0	224.72	276,565	524.22	555,583	2.33
P21_e	21	15,711.5	442.99	440,196	956.05	1,433,494	2.16
P22_a	22	12,238.0	277.81	298,214	362.22	424,371	1.30
P22_b	22	19,183.0	1221.22	1,188,456	2134.62	2,722,740	1.75
P22_c	22	19,963.0	492.68	386,854	1592.04	1,528,020	3.23
P22_d	22	19,981.0	962.91	1,055,536	2859.17	2,820,466	2.97
P22_e	22	20,112.5	703.94	687,176	1128.96	1,451,716	1.60
P23_a	23	14,294.0	575.63	463,762	10,647.84	8,028,793	18.50
P23_b	23	21,116.5	689.46	480,980	1393.27	1,334,876	2.02
P23_c	23	21,511.0	166.20	101,912	437.40	329,796	2.63
P23_d	23	23,522.0	921.34	631,914	6308.06	4,436,370	6.85
P23_e	23	20,798.5	224.96	137,327	992.98	617,944	4.41

Table 4. Experimental comparison of the solution time required by models PROP1 and PROP2 for instances with size $n \leq 23$ and $t = \lfloor n/4 \rfloor$.

			PROP2		PROP1		
Instance	n	Optimal value	T2(s)	BB nodes	T1(s)	BB nodes	T1/T2
S11	11	5852.5	0.30	127	1.69	557	5.62
LW11	11	7483.5	0.27	229	0.31	903	1.14
H12	12	13,685.0	0.23	140	0.27	609	1.16
Am12a	12	2054.0	0.51	1303	1.80	4137	3.51
Am12b	12	2490.5	0.29	501	0.44	1312	1.52
Am13a	13	3083.5	0.72	1241	2.11	4316	2.93
Am13b	13	4401.0	0.39	405	0.58	1239	1.48
H15	15	28,970.0	1.48	1166	4.00	7842	2.70
Am15	15	4537.0	2.81	2837	6.15	9119	2.19
P16a	16	11,409.0	5.52	5660	34.69	47,251	6.28
P16b	16	9636.5	2.15	2102	31.58	41,798	14.71
Am17	17	6409.0	11.29	12,553	29.36	45,691	2.60
Am18	18	7680.5	61.92	79,934	121.05	177,565	1.95
H20	20	10,973.0	226.54	226,050	432.75	578,633	1.91
P20_a	20	18,185.5	91.51	70,586	343.61	437,851	3.75
P20_b	20	22,801.0	42.55	32,376	373.02	546,354	8.77
P21_a	21	11,765.5	182.80	382,500	256.62	171,967	1.40
P21_b	21	18,564.0	33.99	16,967	421.76	439,916	12.41
P21_c	21	16,888.0	257.29	187,579	755.89	561,804	2.94
P21_d	21	19,471.0	129.55	87,340	288.98	209,425	2.23
P21_e	21	19,865.5	124.07	66,345	980.74	949,343	7.90
P22_a	22	15,385.0	1074.74	734,016	10,664.39	6,085,517	9.92
P22_b	22	23,534.0	236.56	80,495	3626.19	1,964,992	15.33
P22_c	22	24,221.0	487.55	305,024	3601.17	3,192,989	7.39
P22_d	22	25,180.0	216.53	86,833	2376.64	1,511,561	10.98
P22_e	22	24,515.5	4584.46	3,199,628	6889.76	7,003,631	1.50
P23_a	23	17,812.0	5835.35	2,507,638	24,548.15	9,140,147	4.21
P23_b	23	26,004.5	607.11	269,602	4192.95	2,540,395	6.91
P23_c	23	26,040.0	1011.29	1,175,501	2988.78	1,353,035	2.96
P23_d	23	27,922.0	1257.50	350,519	13,830.70	5,383,380	11.00
P23_e	23	27,574.5	306.61	94,927	2203.22	1,092,649	7.19

solve PROP2 is smaller than the time required to solve PROP1 on 122 instances out of 124. Only for two instances, LW11 with $t = \lfloor n/3 \rfloor$ and S11 with $t = \lfloor n/5 \rfloor$, did the two models take similar time to be solved (with the ratio T1/T2 of 0.98 and 1, respectively). This result indicates that PROP2 can

Table 5. Experimental comparison of the solution time required by models PROP1 and PROP2 for instances with size $n \leq 23$ and $t = \lfloor n/5 \rfloor$.

			PROP2		PROP1		
Instance	n	Optimal value	T2(s)	BB nodes	T1(s)	BB nodes	T1/T2
S11	11	5852.5	0.29	127	0.29	557	1.00
LW11	11	7483.5	0.28	229	0.31	903	1.10
H12	12	16,545.0	0.31	243	0.44	1249	1.43
Am12a	12	2410.0	0.49	243	1.85	2979	3.80
Am12b	12	3172.5	0.42	538	0.71	2465	1.67
Am13a	13	3821.5	1.21	1571	4.17	9921	3.45
Am13b	13	4656.0	0.57	433	2.05	2128	3.59
H15	15	28,970.0	1.45	1166	3.78	7842	2.60
Am15	15	4537.0	2.64	2837	5.37	9119	2.03
P16a	16	12,279.0	8.06	7295	12.58	15,918	1.56
P16b	16	11,256.5	5.05	4347	52.04	44,730	10.30
Am17	17	6892.0	36.33	33,905	86.16	101,096	2.37
Am18	18	8245.5	96.92	95,296	234.03	337,016	2.41
H20	20	12,272.0	505.95	538,488	1212.99	1,435,470	2.40
P20_a	20	21,215.5	158.96	116,031	713.11	920,100	4.49
P20_b	20	23,902.0	219.41	151,975	469.76	361,400	2.14
P21_a	21	12,382.5	271.37	146,291	1247.99	763,530	4.60
P21_b	21	20,825.0	44.85	16,947	432.13	274,178	9.63
P21_c	21	19,481.0	210.25	74,960	1581.94	973,940	7.52
P21_d	21	20,685.0	49.86	17,218	917.47	853,540	18.40
P21_e	21	22,423.5	223.66	126,600	889.91	691,789	3.98
P22_a	22	16,114.0	2332.03	1,562,648	5596.91	3,180,513	2.40
P22_b	22	25,044.0	329.04	98,963	3182.88	2,106,493	9.67
P22_c	22	25,545.0	3373.12	1,514,437	5331.07	3,101,605	1.58
P22_d	22	26,796.0	904.80	427,357	5149.41	3,286,619	5.69
P22_e	22	27,161.5	5389.16	2,683,013	11,151.26	6,647,403	2.07
P23_a	23	18,619.0	7774.35	3,173,236	258,044.31	43,696,609	33.19
P23_b	23	29,892.5	1191.59	343,791	119,347.75	28,595,785	100.16
P23_c	23	27,553.0	3660.21	1,012,708	44,612.69	13,911,818	12.19
P23_d	23	30,694.0	6251.13	1,682,855	53,527.09	9,706,334	8.56
P23_e	23	29,810.5	1094.86	371,266	9128.33	1,553,980	8.34

Table 6. Minimum, average, standard deviation, and maximum solution times over five instances by the model PROP2 and PROP1.

		PROP2				PROP1			
	n	Min.	Avg.	SD	Max.	Min.	Avg.	SD	Max.
$t = \lfloor n/2 \rfloor$	21	236.32	978.30	614.91	1853.44	448.12	2281.73	1469.14	4059.18
	22	707.60	3031.44	1707.61	4628.71	5843.79	12,285.73	6595.63	21,950.44
	23	2564.93	3894.48	1620.35	6572.39	8918.51	14,460.87	5356.48	21,031.40
$t = \lfloor n/3 \rfloor$	21	107.05	739.26	1071.81	2644.37	244.25	1837.56	2904.51	7012.59
	22	277.81	731.71	373.16	1221.22	362.22	1615.40	951.86	2859.17
	23	166.20	515.51	318.20	921.34	437.40	3955.91	4417.02	10,647.84
$t = \lfloor n/4 \rfloor$	21	33.99	145.54	82.21	257.29	256.62	540.79	315.49	980.74
	22	216.53	1319.96	1857.50	4584.46	2376.64	5431.63	3370.87	10,664.39
	23	306.61	1803.57	2283.28	5835.35	2203.22	9552.76	9604.57	24,548.15
$t = \lfloor n/5 \rfloor$	21	44.85	159.99	105.32	271.37	432.13	1013.88	430.28	1581.94
	22	329.04	2465.63	2024.20	5389.16	3182.88	6082.31	2990.45	11,151.26
	23	1094.86	3994.42	2989.80	7774.35	9128.33	96,932.03	98,476.42	258,044.30

be solved faster than PROP1 for almost all of the instances.

For the instances with size $n \leq 20$, the largest ratio, 14.71, was found for the instance $\{P16, \lfloor n/5 \rfloor\}$. In this case, the computational times consumed by PROP1 and PROP2 are 31.58 s and 2.15 s, respectively. For instances with sizes $n = 21$, 22, and 23, Table 6 shows, for a given size n, the comparison between the two models presenting the minimum (Min.), average (Avg.), standard deviation (SD) and maximum (Max.) computational times over five instances with $t = \lfloor n/2 \rfloor, t = \lfloor n/3 \rfloor, t = \lfloor n/4 \rfloor$, and $t = \lfloor n/5 \rfloor$, respectively. For each model, it can be noticed that the average time consumed generally increases by a large amount as n increases for each model. It also can be seen that the average consumed time to solve PROP2 is much less than the average time to solve PROP1. In Table 6 with $t = \lfloor n/2 \rfloor$, for example, when solving PROP2, the average time for the instances with $n = 21$ is about 16 min and for the instances with $n = 22$ is around 50 min, while the instance with $n = 23$ required an average time of about 65 min. However, the corresponding average times when solving PROP1 are about 38 min, 3.5 h and 4 h, respectively. Among all of the instances, the largest running time with model PROP1 is of about 71.5 h, corresponding to instance $\{\text{P23\_a}, \lfloor n/5 \rfloor\}$, however, with model PROP2 the same instance needs 2.2 h.

Table 7. Experimental comparison of the performance of PROP1 with PROP2 for instances with size $n > 23$ and $t = \lfloor n/2 \rfloor$ within a time limit of 28,800 s.

		PROP2					PROP1				
Instance	n	UB	LB	Gap(%)	T2(s)	BB nodes	UB	LB	Gap(%)	T1(s)	BB nodes
P24_a	24	11,778.0	11,778.0	0	17,580.62	13,010,009	11,778.0	10,531.9	10.57	28,803.16	13,265,997
P24_b	24	16,178.5	16,178.5	0	19,177.99	1,335,2356	16,179.0	14,188.4	12.30	28,800.98	1,048,0451
P24_c	24	17,033.0	17,033.0	0	17,132.04	8,259,471	17,033.0	15,305.7	10.14	28,802.61	12,663,895
P24_d	24	15,817.0	15,817.0	0	10,766.44	6,541,030	16,001.0	13,638.6	14.76	28,802.70	8,264,903
P24_e	24	18,530.0	18,530.0	0	27,138.77	16,485,398	18,530.0	15,800.8	14.73	28,802.81	8,824,509
N25_1	25	2349.0	2010.8	14.39	28,806.03	11,479,336	2349.0	1731.3	26.27	28,803.54	10,637,895
N25_2	25	19,138.5	19,138.5	0	24,284.51	9,795,116	19,181.0	15,943.7	16.87	28,802.29	8,304,060
N25_3	25	12,604.0	11,109.9	11.90	28,802.72	9,202,252	12,624.0	10,693.1	15.30	28,802.03	9,461,839
N25_4	25	25,026.5	22,621.1	9.60	28,801.76	8,983,839	25,079.5	20,472.8	18.40	28,801.03	6,101,055
N25_5	25	8011.0	7356.3	8.20	28,802.67	12,381,851	8011.0	6531.5	18.50	28,801.74	9,279,850
N30_1	30	4174.0	2436.5	41.61	28,800.01	1,351,433	4174.0	1995.5	52.18	28,806.94	957,703
N30_2	30	11,250.5	7375.5	34.44	28,810.09	2,500,029	11,242.0	6239.0	44.50	28,800.48	1,314,987
N30_3	30	23,410.0	16,705.2	28.60	28,800.74	1,563,823	23,558.0	13,909.2	41.00	28,800.08	1,887,287
N30_4	30	32,756.5	27,750.9	15.30	28,800.25	740,614	32,745.5	26,165.5	20.10	28,800.40	795,234
N30_5	30	60,628.0	46,650.6	23.10	28,800.45	1,086,341	60,603.0	42,598.0	29.70	28,800.06	2,060,438

The ratio between the time required by PROP1 and by PROP2 reaches 100.16 for instance $\{\text{P23\_b}, \lfloor n/5 \rfloor\}$, which is the largest observed ratio considering Tables 2–5. For this instance, the times when using model PROP2 and PROP1 are 20 min and 33.2 h.

It can be seen from the tables, for almost all the instances considered, that PROP2 requires a smaller number of branch-and-bound nodes than PROP1, except for three instances, $\{Am18, \lfloor n/3 \rfloor\}, \{Am18, \lfloor n/5 \rfloor\}$ and $\{\text{P21\_a}, \lfloor n/4 \rfloor\}$. This does not necessarily reflect on solution time. For example, for instance $\{Am18, \lfloor n/3 \rfloor\}$, PROP1 requires 74,563 nodes and was solved in 53.68 seconds, while PROP2 needs 155,917 nodes and was solved in 24.4 seconds. The optimal solution and corresponding cost for the instances that have not been considered in Amaral (2013b) are given in Appendix A, see Tables A1–A4.

4.2. Evaluation of PROP1 and PROP2 on instances with sizes $24 \le n \le 30$

In this subsection, we consider larger PROP instances with sizes $24 \le n \le 30$, and evaluate the difference between two models in terms of upper bound, lower bound, gap, running time, and branch-and-bound nodes. Random instances with sizes $n = 24$ are introduced by following the rule in Amaral (2013b). Instance P24_a was constructed by deleting the last facilities of the SRFLP instance with size $n = 30$ given by Heragu and Kusiak (1991). Instances P24_b, P24_c, P24_d, and P24_e are characterised by $\{n = 24, L_n(5, 14), NUG(30-6)\}$. We consider $\{\text{P24\_a}, t\}$, $\{\text{P24\_b}, t\}, \ldots, \{\text{P24\_e}, t\}$, for each $t \in \lfloor n/2 \rfloor, \lfloor n/3 \rfloor, \lfloor n/4 \rfloor, \lfloor n/5 \rfloor$ which involves 20 PROP instances. With the addition of the instances with sizes 25 and 30, we have a total of 60 PROP instances in this subsection. For the larger instances in this subsection, the Gurobi solver may require a very long time. Thus, a time limit of eight hours was set for the solver and the two models are evaluated in terms of upper bound, lower bound, gap, running time, and number of branch-and-bound nodes.

The results obtained for the two models on the instances with sizes from 24 to 30 are displayed in Tables 7–10. Consider Table 7 when $t = \lfloor n/2 \rfloor$, for example. The first column refers to the instance name. The second column gives the number n of the departments. The following five columns report results for the PROP2 model. The first two columns of these five columns present the upper bounds and lower bounds when Gurobi terminated, following by a column relative to the corresponding gap which is calculated as $[(UB-LB)/LB] \times 100\%$. The last two of these five columns give the computational time and the number of branch-and-bound nodes explored. The next five columns report the same information for Model PROP1. Tables 8–10 display similar results for PROP instances with $t = \lfloor n/3 \rfloor$, $\lfloor n/4 \rfloor$, $\lfloor n/5 \rfloor$, respectively.

Tables 7–10 show that model PROP2 performs better than model PROP1 in terms of computational times and gap. With model PROP2, 33 instances are solved to optimality, however only 10 instances are solved with model PROP1. For the instances for which none of the models can obtain an optimal solution within the time limit of 8 hours, the gap is much lower for model PROP2. For the 20 instances with $n = 24$, 17 optimal solutions are found using PROP2, while using model PROP1 only 5 optimal solutions are found (within the time limit); and for the three instances for which PROP2 cannot find an optimal solution, $\{\text{P24\_a}, \lfloor n/4 \rfloor\}, \{\text{P24\_b}, \lfloor n/5 \rfloor\}$, $\{\text{P24\_d}, \lfloor n/5 \rfloor\}$, the gaps obtained are 3.33%, 8.03%, and 6.27%, respectively, which are much lower than the respective gaps 14.32%, 26.32%, and 24.27% with model PROP1. In addition, another observation is that none of the instances can be solved to optimality within the time limit with the model PROP1 when $t = \lfloor n/2 \rfloor$ or $\lfloor n/5 \rfloor$, nevertheless, 13 instances can be solved to optimality with model

Table 8. Experimental comparison of the performance of PROP1 with PROP2 for instances with size $n > 23$ and $t = \lfloor n/3 \rfloor$ within a time limit of 28,800 s.

		PROP2					PROP1				
Instance	n	UB	LB	Gap(%)	T2(s)	BB nodes	UB	LB	Gap(%)	T1(s)	BB nodes
P24_a	24	14,730.0	14,730.0	0	2565.00	1,838,606	14,730.0	14,730.0	0	9510.81	5,378,976
P24_b	24	19,113.5	19,113.5	0	26,703.12	17,536,440	19,312.5	16,345.1	15.36	28,800.61	12,714,355
P24_c	24	20,495.0	20,495.0	0	4362.23	2,635,974	20,601.0	18,803.7	8.27	28,805.38	13,391,348
P24_d	24	20,655.0	20,655.0	0	7104.45	4,388,887	20,655.0	20,655.0	0	18,595.57	9,764,865
P24_e	24	22,243.0	22,243.0	0	11,862.98	8,623,181	22,243.0	22,243.0	0	26,421.28	12,165,903
N25_1	25	3077.0	3077.0	0	19,584.21	9,039,328	3081.0	2754.3	10.58	28,802.10	9,937,602
N25_2	25	23,826.5	23,826.5	0	914.02	358,261	23,826.5	23,765.5	0	17,218.45	5,648,068
N25_3	25	18,714.0	18,714.0	0	701.34	320,875	18,714.0	18,714.0	0	3788.55	2,108,602
N25_4	25	30,647.5	30,647.5	0	1018.80	389,416	30,647.5	30,647.5	0	5832.28	1,928,258
N25_5	25	10,126.0	10,126.0	0	1054.79	481,235	10,126.0	10,126.0	0	6100.88	2,233,569
N30_1	30	5319.0	4240.3	20.27	28,800.85	3,019,499	5326.0	3652.0	31.41	28,802.85	1,499,785
N30_2	30	14,922.5	13,191.2	11.60	28,800.65	2,992,265	15,079.5	11,564.8	23.31	28,800.35	1,342,796
N30_3	30	27,652.0	481,235.0	20.50	28,803.32	3,160,967	27,586.0	18,160.7	34.20	28,800.39	2,188,483
N30_4	30	44,498.5	44,498.5	0	15,355.64	1,779,856	44,498.5	18,160.7	10.60	28,800.60	1,723,976
N30_5	30	69,022.0	55,014.8	20.30	28,800.57	1,136,895	68,998.0	49,912.6	27.70	28,800.75	1,442,851

Table 9. Experimental comparison of the performance of PROP1 with PROP2 for instances with size $n > 23$ and $t = \lfloor n/4 \rfloor$ within a time limit of 28,800 s.

		PROP2					PROP1				
Instance	n	UB	LB	Gap(%)	T2(s)	BB nodes	UB	LB	Gap(%)	T1(s)	BB nodes
P24_a	24	18,757.0	18,757.0	0	4325.84	1,012,450	18,757.0	18,126.3	3.36	28,801.34	11,530,798
P24_b	24	24,439.5	23,625.4	3.33	28,808.83	9,237,981	24,439.5	20,939.2	14.32	28,801.40	10,077,985
P24_c	24	24,826.0	24,826.0	0	1141.53	400,215	24,826.0	24,826.0	0	26,047.14	14,873,125
P24_d	24	23,908.0	23,908.0	0	12,911.15	6,291,269	23,942.0	20,821.6	13.03	28,809.07	8,861,789
P24_e	24	28,411.0	28,411.0	0	2313.52	920,497	28,411.0	28,411.0	0	15,886.58	7,801,552
N25_1	25	3705.0	3705.0	0	23,957.70	9,337,747	3705.0	3221.6	13.04	28,801.50	5,123,851
N25_2	25	26,229.5	26,229.5	0	880.46	224,756	26,229.5	26,229.5	0	24,190.54	5,023,026
N25_3	25	23,081.0	23,081.0	0	2036.71	574,772	23,081.0	21,033.2	8.90	28,800.94	3,576,816
N25_4	25	33,584.5	33,584.5	0	1097.97	267,596	33,584.5	28,942.2	13.80	28,800.13	4,746,585
N25_5	25	11,289.0	11,289.0	0	1998.17	752,992	11,289.0	9976.7	11.60	28,800.49	5,389,657
N30_1	30	6864.0	5260.3	23.35	28,807.52	117,4380	6849.0	4514.4	34.08	28,800.42	1,091,407
N30_2	30	18,928.5	16,277.8	14.00	28,805.83	1,917,497	18,928.5	14,521.5	23.28	28,800.17	807,124
N30_3	30	34,676.0	28,271.1	18.50	28,801.03	1,725,900	34,557.0	26,583.9	23.10	28,800.69	1,277,332
N30_4	30	52,710.5	50,467.8	4.30	28,800.36	155,5002	52,764.5	43,504.0	17.50	28,800.44	719,696
N30_5	30	89,923.0	77,430.8	13.90	28,800.54	124,4379	90,241.0	57,769.9	36.00	28,800.38	481,051

Table 10. Experimental comparison of the performance of PROP1 with PROP2 for instances with size $n > 23$ and $t = \lfloor n/5 \rfloor$ within a time limit of 28,800 s.

		PROP2					PROP1				
Instance	n	UB	LB	Gap(%)	T2(s)	BB nodes	UB	LB	Gap(%)	T1(s)	BB nodes
P24_a	24	21,729.0	21,729.0	0	25,972.32	6,712,709	21,729.0	18,533.5	14.70	28,803.33	6,294,801
P24_b	24	28,789.5	26,477.5	8.03	28,805.74	6,959,256	28,789.5	21,209.7	26.32	28,803.66	4,132,514
P24_c	24	30,347.0	30,347.0	0	27,485.23	5,080,227	30,347.0	25,302.3	16.62	28,804.51	4,251,872
P24_d	24	29,170.0	27,341.7	6.27	28,808.50	7,621,173	29,170.0	22,089.6	24.27	28,803.00	4,255,903
P24_e	24	33,061.0	33,061.0	0	13,100.83	2,545,725	33,061.0	26,855.4	18.77	28,801.90	3,249,248
N25_1	25	4039.0	3659.2	9.38	28,805.39	5,011,466	4039.0	3163.5	21.66	28,801.11	2,795,971
N25_2	25	30,193.5	30,193.5	0	2120.83	342,211	30,223.5	28,467.1	5.81	28,800.19	3,199,128
N25_3	25	23,167.0	23,167.0	0	10,750.42	186,4118	23,167.0	19,607.7	15.40	28,800.03	2,852,565
N25_4	25	38,689.5	38,689.5	0	2611.43	404,986	38,878.5	32,236.4	17.10	28,800.04	3,253,211
N25_5	25	12,951.0	12,951.0	0	2333.92	372,932	12,977.0	10,506.5	19.00	28,800.90	2,348,853
N30_1	30	7330.0	5269.3	28.10	28,800.43	973,332	7347.0	4592.5	37.48	28,800.12	616,010
N30_2	30	20,020.5	16,337.0	18.40	28,807.33	1,124,145	19,944.5	14,021.0	29.70	28,800.11	677,428
N30_3	30	39,524.0	32,803.3	17.00	28,800.48	963,067	39,544.0	26,328.9	33.40	28,800.51	692,366
N30_4	30	59,587.5	52,087.1	12.60	28,800.30	731,412	59,587.5	43,334.0	27.30	28,800.50	566,397
N30_5	30	105,103.0	87,300.9	16.90	28,800.57	882,046	104,825.0	69,101.1	34.10	28,800.47	595,013

PROP2. The instances that were solved to optimality with PROP2, their optimal solutions and their costs are given in Appendix A, see Tables A1–A4.

4.3. Pairwise comparisons

Statistical tests, known as pairwise comparisons, can be used to analyse multiple population means in pairs for determining whether they significantly differ from one another. Such tests can be used to compare the performance of two methods when solving the same set of instances (Derrac, García, Molina, & Herrera, 2011). To observe whether PROP2 significantly outperforms PROP1, we used a Sign test and a Wilcoxon signed ranks test to compare PROP2 and PROP1 for instances with size $n \leq 23$ and $n > 23$, respectively. The Sign test is quite simple for it only requires counting the number of wins achieved by model PROP2. A Wilcoxon signed ranks test is analogous to the paired t-test in non-parametric statistical procedures. It is a pairwise test that is used to detect significant differences between two sample means, that is, the behaviour of two models (Derrac et al., 2011). To do so, we defined two hypotheses, the null hypothesis H_0 which means no difference between PROP2 and PROP1, and the alternative hypothesis H_1 which means significant difference between two models. The significance level α is usually set to 0.05 and we used the same value here.

The results of the tests are displayed in Table 11 for four instance sets ($t = \lfloor n/5 \rfloor, \lfloor n/3 \rfloor, \lfloor n/4 \rfloor$, and $\lfloor n/5 \rfloor$, respectively) with size $n \leq 23$ and $n > 23$. The number of instances on which PROP2 is a winner or loser was counted to calculate p-values in the Sign test. The solution time required for PROP2 and PROP1 in Tables 2–5 is used to produced p-values in the Wilcoxon signed ranks test for instances with size $n \leq 23$. Moreover, the Paired sample t-test is processed for instance set $t = \lfloor n/3 \rfloor$ with size $n \leq 23$ for only this instance set with the differences are normally distributed. The p-value is 0.025 in this situation. On the other hand, the gap of PROP2 and PROP1 in Tables 8–10 is used to produce p-values in the Wilcoxon signed ranks test for instances with size $n > 23$. We obtained p-values for these two tests using the well-known statistical software package SPSS. All p-values produced by both tests are less than the significance level of 0.05 for each problem instance set. Thus we reject the null hypothesis H_0 for the test results take statistical significance. Furthermore, we conclude that PROP2 shows a significant improvement over PROP1.

Table 11. Statistical comparison of PROP1 and PROP2.

		PROP2 versus PROP1				Wilcoxon signed
					Sign test	ranks test
n		Wins	Equals	Loses	(p-value)	(p-value)
$n \leq 23$	$t = \lfloor n/2 \rfloor$	31	0	0	<.001	<.001
	$t = \lfloor n/3 \rfloor$	30	0	1	<.001	<.001
	$t = \lfloor n/4 \rfloor$	31	0	0	<.001	<.001
	$t = \lfloor n/5 \rfloor$	30	1	0	<.001	<.001
$n > 23$	$t = \lfloor n/2 \rfloor$	15	0	0	<.001	<.001
	$t = \lfloor n/3 \rfloor$	15	0	0	<.001	.014
	$t = \lfloor n/4 \rfloor$	15	0	0	<.001	.003
	$t = \lfloor n/5 \rfloor$	15	0	0	<.001	<.001

4.4. Comparing with the PSA metaheuristic of Maadi et al. (2017)

We define:

$$\mathrm{RPD} = \frac{\mathrm{OFV_w} - \mathrm{OFV_b}}{\mathrm{OFV_b}} \times 100\% \tag{24}$$

where $\mathrm{OFV_w}$ is the worst OFV, and $\mathrm{OFV_b}$ is the best OFV between two methods. If the OFVs are the same, $\mathrm{RPD} = 0$.

In order to further validate the performance of the PROP2 in this article, we now compare the results derived by solving PROP2 with a time limit of 28,800 sec. with one metaheuristic, a population based simulated annealing algorithm (PSA) (Maadi et al., 2017) for instances with size $n = 30$, as shown in Table 12. The first two columns report the value of t and the name of the instance. The next two columns display results for model PROP2: the objective function value (OFV) and the corresponding computational time. For each instance, PSA was run 20 times. The best OFV found among the 20 runs, their corresponding computational times are showed in the following two columns. The time required by PSA is noted as T3. The last column shows the

Table 12. Comparing the performance of solving PROP2 with a time limit of 28,800 sec. with the best of 20 runs of PSA (Maadi, Javidnia, & Jamshidi,2017).

		PROP2		PSA		
	Instance	OFV	T2(s)	OFV	T3(s)	RPD
$t = \lfloor n/2 \rfloor$	N30_1	4174.0	28,800.01	4174.0	2.83	0
	N30_2	11,250.5	28,810.09	11,154.5	2.99	0.0086
	N30_3	23,410.0	28,800.74	23,127.0	2.92	0.0122
	N30_4	32,756.5	28,800.25	32,651.5	2.94	0.0032
	N30_5	60,628.0	28,800.45	60,353.0	3.00	0.0046
$t = \lfloor n/3 \rfloor$	N30_1	5319.0	28,800.85	5310.0	3.92	0.0017
	N30_2	14,922.5	28,800.65	14,894.5	3.62	0.0019
	N30_3	27,652.0	28,803.32	27,306.0	3.20	0.0127
	N30_4	44,498.5	15,355.64	44,498.5	3.11	0
	N30_5	69,022.0	28,800.57	68,998.0	3.14	0.0003
$t = \lfloor n/4 \rfloor$	N30_1	6864.0	28,807.52	6791.0	3.98	0.0107
	N30_2	18,928.5	28,805.83	18,928.5	4.00	0
	N30_3	34,676.0	28,801.03	34,523.0	3.99	0.0044
	N30_4	52,710.5	28,800.36	52,710.5	3.99	0.0000
	N30_5	89,923.0	28,800.54	89,548.0	3.99	0.0042
$t = \lfloor n/5 \rfloor$	N30_1	7330.0	28,800.43	7289.0	4.25	0.0056
	N30_2	20,020.5	28,807.33	19,785.5	4.40	0.0119
	N30_3	39,524.0	28,800.48	39,524.0	4.41	0
	N30_4	59,587.5	28,800.30	59,587.5	4.30	0
	N30_5	105,103.0	28,800.57	104,449.0	4.33	0.0063
	#Better	0		15		
	Average		28,129.85		3.67	0.0044

relative percentage deviation (RPD) of the OFVs with the two approaches, which is calculated by Eq. 24. The row "#Better" indicates the number of instances for which one method gives better OFV than the other. The row "Average" shows the average value of computational time and RPD.

Table 12 shows that among the 20 instances, the best of 20 PSA runs yields a better OFV for 15 instances. For five instances PROP2 and PSA produced the same OFV. On the other hand, as we can see from the last column of Table 12, the average value of RPD is 0.0044, which indicates that the gaps between the OFVs produced by the two methods are quite small. Note that for one instance with size $n = 30$, $\{\text{N30\_4}, t = \lfloor n/3 \rfloor\}$ PROP2 produced an optimal solution with a computational time of 15,355.64 s < 28,800 s. The PSA was coded in C# and run on an Intel (R) core (TM) i5-3210 CPU @ 2.5 GHz with 4 GB memory running the Windows 8.1 operating system.

It should be noted that the implementation and purposes of the exact solution method and the meta-heuristic algorithm are different. While the PSA is a very good heuristic, there is considerable variability in its performance and the tuning of its parameters for the problem instances at hand. Moreover, designing and implementing a successful metaheuristic requires special expertise, problem specific algorithm coding, and often a great deal of trial and error. The exact method as presented here does not require such resources. Nor does the exact method require running multiple iterations as the method by Maadi et al. (2017) did (their results are the best among 20 runs). Also, simulated annealing is a highly stochastic algorithm that can be quite dependent on cooling schedule parameter values and on the initial starting solution. Our exact method can be used in a straightforward and dependable manner.

5. Conclusions

In this article, the mathematical model given by Amaral (2013b) for the parallel ordering problem (PROP) has been revisited. We proposed a new mixed integer programming model for this problem, which modifies some constraints of the model in Amaral (2013b). Computational results showed that for the instances with sizes less than or equal to 23, the new model (PROP2) leads to much faster optimal solutions than those obtained with the previous model (PROP1). For the larger instances with sizes equal to 24, 25, or 30, the performance of the new model was better than the former model in terms of computational time or gap. In fact, the computational results confirmed the superiority of PROP2 over PROP1. We believe that the new model can provide some inspirations for future work with respect to the parallel row ordering problem and relevant row layout problem. The computational results provided in this paper can be used as a baseline for comparison of heuristic or meta-heuristic algorithms for the problem.

As future research, one could consider a study of valid inequalities that are useful for the new model. One could incorporate these valid inequalities in a branch and cut framework, which may further improve the results given here. In addition, the development of effective heuristic and meta-heuristic algorithms for dealing with larger PROP instances is another interesting direction.

Acknowledgments

The authors would like to thank the two anonymous referees for their valuable feedback and suggestions which contributed to the improvement of the original version of this paper.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This research was supported in part by the Sichuan Province Miaozi Innovation Project of China [grant number 2017119]; National Natural Science Foundation of China [grant number 51705436]; This study was financed in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

Adolphson, D., & Hu, T. C. (1973). Optimal linear ordering. *SIAM Journal on Applied Mathematics*, *25*(3), 403–423. doi:10.1137/0125042

Amaral, A. R. S. (2006). On the exact solution of a facility layout problem. *European Journal of Operational Research*, *173*(2), 508–518. doi:10.1016/j.ejor.2004.12.021

Amaral, A. R. S. (2008). An exact approach to the one-dimensional facility layout problem. *Operations Research*, *56*(4), 1026–1033. doi:10.1287/opre.1080.0548

Amaral, A. R. S. (2009a). A mixed 0-1 linear programming formulation for the exact solution of the minimum linear arrangement problem. *Optimization Letters*, *3*(4), 513–520. doi:10.1007/s11590-009-0130-0

Amaral, A. R. S. (2009b). A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics*, *157*(1), 183–190. doi:10.1016/j.dam.2008.06.002

Amaral, A. R. S. (2012). The corridor allocation problem. *Computers & Operations Research*, *39*(12), 3325–3330. doi:10.1016/j.cor.2012.04.016

Amaral, A. R. S. (2013a). Optimal solutions for the double row layout problem. *Optimization Letters*, *7*(2), 407–413. doi:10.1007/s11590-011-0426-8

Amaral, A. R. S. (2013b). A parallel ordering problem in facilities layout. *Computers & Operations Research*, *40*(12), 2930–2939. doi:10.1016/j.cor.2013.07.003

Amaral, A. R. S. (2018). A mixed-integer programming formulation for the double row layout of machines in manufacturing systems. *International Journal of Production Research*, 1–14. doi:10.1080/00207543.2018.1457811

Amaral, A. R. S., & Letchford, A. N. (2013). A polyhedral approach to the single row facility layout problem. *Mathematical Programming*, *141*(1–2), 453–477. doi:10.1007/s10107-012-0533-z

Andrade, R. C., & Ferreira, M. D S. (2017). Problema de ordenação de departamentos em linhas simples. In: *XLIX SBPO-Simpósio Brasileiro de Pesquisa Operacional* (pp. 2874–2879). Blumenau/SC, Brazil. http://www.sbpo2017.iltc.br/pdf/168752.pdf

Anjos, M. F., Kennings, A., & Vannelli, A. (2005). A semidefinite optimization approach for the single-row layout problem with unequal dimensions. *Discrete Optimization*, *2*(2), 113–122. doi:10.1016/j.disopt.2005.03.001

Anjos, M. F., & Vannelli, A. (2008). Computing globally optimal solutions for single-row layout problems using semidefinite programming and cutting planes. *INFORMS Journal on Computing*, *20*(4), 611–617. doi:10.1287/ijoc.1080.0270

Anjos, M. F., & Vieira, M. V. (2017). Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions. *European Journal of Operational Research*, *261*(1), 1–16. doi:10.1016/j.ejor.2017.01.049

Brusco, M. J., & Stahl, S. (2000). Using quadratic assignment methods to generate initial permutations for least-squares unidimensional scaling of symmetric proximity matrices. *Journal of Classification*, *17*(2), 197–223. doi:10.1007/s003570000019

Chung, J., & Tanchoco, J. (2010). The double row layout problem. *International Journal of Production Research*, *48*(3), 709–727. doi:10.1080/00207540802192126

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3–18. doi:10.1016/j.swevo.2011.02.002

Dickey, J., & Hopkins, J. (1972). Campus building arrangement using topaz. *Transportation Research*, *6*(1), 59–68. doi:10.1016/0041-1647(72)90111-6

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, *31*(2), 255–267. doi:10.1016/j.arcontrol.2007.04.001

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York, NY: W. H. Freeman & Co.

Gen, M., Ida, K., & Cheng, C. (1995). Multirow machine layout problem in fuzzy environment using genetic algorithms. *Computers & Industrial Engineering*, *29*(1–4), 519–523. doi:10.1016/0360-8352(95)00127-M

Gurobi Optimization, I. (2015). *Gurobi Optimizer Reference Manual.* Retrieved from http://www.gurobi.com

Heragu, S. S., & Kusiak, A. (1988). Machine layout problem in flexible manufacturing systems. *Operations Research*, *36*(2), 258–268. doi:10.1287/opre.36.2.258

Heragu, S. S., & Kusiak, A. (1991). Efficient models for the facility layout problem. *European Journal of Operational Research*, *53*(1), 1–13. doi:10.1016/0377-2217(91)90088-D

Hosseini-Nasab, H., Fereidouni, S., Ghomi, S. M. T. F., & Fakhrzad, M. B. (2018). Classification of facility layout problems: a review study. *The International Journal of Advanced Manufacturing Technology*, *94*(1–4), 957–977. doi:10.1007/s00170-017-0895-8

Hungerländer, P. (2014). *A semidefinite optimization approach to the parallel row ordering problem* (Tech. Rep.). Technical report, Alpen-Adria Universität Klagenfurt, Mathematics, Optimization Group, TR-ARUK-MO-14-05.

Hungerländer, P. (2017). The checkpoint ordering problem. *Optimization*, *66*(10), 1699–1712.

Hungerländer, P., & Rendl, F. (2013). A computational study and survey of methods for the single-row facility layout problem. *Computational Optimization and Applications*, *55*(1), 1–20. doi:10.1007/s10589-012-9505-8

Keller, B. (2017). Construction heuristics for the single row layout problem with machine-spanning clearances. *INFOR: Information Systems and Operational Research*, 1–24. doi:10.1080/03155986.2017.1393729

Keller, B., & Buscher, U. (2015). Single row layout models. *European Journal of Operational Research*, *245*(3), 629–644. doi:10.1016/j.ejor.2015.03.016

Koopmans, T. C., & Beckmann, M. (1957). Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, *25*(1), 53–76. doi:10.2307/1907742

Kothari, R., & Ghosh, D. (2012). The single row facility layout problem: state of the art. *Opsearch*, *49*(4), 442–462. doi:10.1007/s12597-012-0091-4

Laporte, G., & Mercure, H. (1988). Balancing hydraulic turbine runners: A quadratic assignment problem. *European Journal of Operational Research*, *35*(3), 378–381. doi:10.1016/0377-2217(88)90227-5

Löfberg, J. (2004). YALMIP : a toolbox for modeling and optimization in MATLAB. In *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)* (pp. 284–289). New Orleans, LA: IEEE. doi:10.1109/CACSD.2004.1393890

Love, R., & Wong, J. (1976). On solving a one-dimensional space allocation problem with integer programming. *INFOR: Information Systems and Operational Research*, *14*(2), 139–143. doi:10.1080/03155986.1976.11731633

Maadi, M., Javidnia, M., & Jamshidi, R. (2017). Two strategies based on meta-heuristic algorithms for parallel row ordering problem (prop.). *Iranian Journal of Management Studies*, *10*(2), 467–498.

Meller, R. D., & Gau, K.-Y. (1996). The facility layout problem: recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, *15*(5), 351–366.

Ning, X., & Li, P. (2018). A cross-entropy approach to the single row facility layout problem. *International Journal of Production Research*, *56*(11), 3781–3794. doi:10.1080/00207543.2017.1399221

Pollatschek, M. A., Gershoni, N., & Radday, Y. T. (1976). Optimization of the typewriter keyboard by simulation. *Angewandte Informatik*, *17*, 438–439.

Secchin, L., & Amaral, A. R. S. (2018). An improved mixed-integer programming model for the double row layout of facilities. *Optimization Letters*, 1–7. doi:10.1007/s11590-018-1263-9

Simmons, D. M. (1969). One-dimensional space allocation: an ordering algorithm. *Operations Research*, *17*(5), 812–826. doi:10.1287/opre.17.5.812

Singh, S., & Sharma, R. (2008). Two-level modified simulated annealing based approach for solving facility layout problem. *International Journal of Production Research*, *46*(13), 3563–3582. doi:10.1080/00207540601178557

Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). *Facilities planning*. New York: John Wiley & Sons.

Wess, B., & Zeitlhofer, T. (2004). On the Phase Coupling Problem Between Data Memory Layout Generation and Address Pointer Assignment. In H. Schepers (Eds.), *Software and Compilers for Embedded Systems. SCOPES 2004* (LNCS, vol. 3199). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-30113-4_12

Zhang, Z., & Murray, C. C. (2012). A corrected formulation for the double row layout problem. *International Journal of Production Research*, *50*(15), 4220–4223. doi:10.1080/00207543.2011.603371

Zhou, J., Love, P. E. D., Teo, K. L., & Luo, H. (2017). An exact penalty function method for optimising qap formulation in facility layout problem. *International Journal of Production Research*, *55*(10), 2913–2929. doi:10.1080/00207543.2016.1229068

Zuo, X., Murray, C. C., & Smith, A. E. (2014). Solving an extended double row layout problem using multiobjective tabu search and linear programming. *IEEE Transactions on Automation Science and Engineering*, *11*(4), 1122–1132. doi:10.1109/TASE.2014.2304471

Appendix: A optimal solution and corresponding cost

See Tables A1–A4.

Table A1. Optimal solution and corresponding cost for instances with $t = \lfloor n/2 \rfloor$.

Instance	*n*	Optimal value	Optimal solution												
LW11	11	4668.5	4	5	1	3	2								
			6	9	7	8	10	11							
H12	12	9305.0	3	1	4	2	5	6							
			9	12	11	8	10	7							
Am12a	12	1583.0	3	2	5	6	4	1							
			7	12	11	10	8	9							
Am12b	12	1697.5	1	4	5	6	2	3							
			11	10	8	9	12	7							
Am13a	13	2668.5	5	6	3	2	4	1							
			9	7	11	12	10	8	13						
Am13b	13	3210.0	1	2	4	3	5	6							
			12	8	13	11	10	9	7						
H15	15	17,780.0	6	3	4	5	2	1	7						
			10	15	14	13	12	9	8	11					
Am17	17	4853.0	3	6	5	7	2	4	8	1					
			9	13	10	14	12	11	16	15	17				
Am18	18	5742.5	9	6	3	2	4	8	7	5	1				
			13	10	18	14	16	11	12	15	17				
H20	20	8190.0	9	5	6	7	2	4	8	10	3	1			
			13	19	20	15	12	14	11	16	18	17			
P24_a	24	11,778.0	4	8	9	3	7	11	10	1	6	12	5	2	
			14	20	16	19	21	23	22	13	18	15	17	24	
P24_b	24	16,178.5	1	12	6	10	7	8	11	4	9	3	2	5	
			15	17	24	13	22	16	19	18	23	14	20	21	
P24_c	24	17,033.0	1	12	6	10	7	8	11	9	4	3	2	5	
			24	17	15	13	22	16	19	18	23	21	20	14	
P24_d	24	15,817.0	4	11	3	9	8	7	1	10	12	6	5	2	
			20	14	23	18	16	19	22	15	17	24	13	21	
P24_e	24	18,530.0	8	11	4	1	3	7	9	10	12	6	5	2	
			17	15	23	18	22	19	16	14	20	24	21	13	
N25_2	25	19,138.5	1	9	10	7	6	8	11	2	4	3	12	5	
			13	23	19	22	15	16	21	24	14	25	20	18	17

Table A2. Optimal solution and corresponding cost for instances with $t = \lfloor n/3 \rfloor$.

Instance	n	Optimal cost	Optimal solution																			
LW11	11	6880.5	3	2	1																	
			10	5	7	8	9	4	6	11												
H12	12	11,725.0	3	1	4	2																
			9	12	11	8	10	5	6	7												
Am12a	12	1928.0	1	4	2	3																
			5	8	6	10	11	12	9	7												
Am12b	12	2226.5	3	1	2	4																
			7	12	11	10	5	6	8	9												
Am13a	13	3009.5	1	4	2	3																
			13	8	10	12	11	5	6	7	9											
Am13b	13	3811.0	1	2	4	3																
			12	8	13	11	5	6	9	10	7											
H15	15	24,110.0	1	4	3	2	5															
			10	6	15	14	13	8	12	9	11	7										
Am17	17	5542.0	1	5	2	4	3															
			17	7	15	8	12	11	14	16	10	6	13	9								
Am18	18	6402.5	1	5	4	2	3	6														
			17	7	15	8	12	11	16	14	18	10	13	9								
H20	20	10,398.0	1	3	2	4	5	6														
			17	18	10	7	12	14	11	16	8	15	20	19	13	9						
P24_a	24	14,730.0	4	2	5	3	7	8	6	1												
			16	9	21	13	10	11	19	23	22	18	15	17	24	12	20	14				
P24_b	24	19,113.5	5	2	3	4	6	7	8	1												
			13	21	9	16	11	10	19	22	18	23	17	24	12	14	20	15				
P24_c	24	20,495.0	5	2	6	1	7	3	8	4												
			24	12	13	10	9	21	22	19	11	16	15	17	23	18	20	14				
P24_d	24	20,655.0	2	5	6	4	3	7	8	1												
			21	13	9	10	16	11	19	22	18	12	24	15	17	23	14	20				
P24_e	24	22,243.0	2	5	6	3	4	7	8	1												
			13	21	16	9	11	10	19	22	18	23	20	14	12	24	15	17				
N25_1	25	3077.0	5	2	3	8	4	6	7	1												
			24	18	11	25	14	21	16	9	19	10	20	15	22	23	17	12	13			
N25_2	25	23,826.5	5	1	6	7	3	8	2	4												
			20	22	15	10	19	16	21	9	24	25	11	14	18	23	17	12	13			
N25_3	25	18,714.0	3	5	2	7	1	6	8	4												
			18	16	24	21	11	14	25	17	19	10	23	9	20	22	15	13	12			
N25_4	25	30,647.5	5	2	1	3	8	6	7	4												
			18	11	9	22	15	25	16	21	24	19	10	20	14	23	17	12	13			
N25_5	25	10,126.0	5	1	6	7	3	8	2	4												
			20	22	15	10	19	16	21	9	25	24	11	14	18	23	17	12	13			
N30_4	30	44,498.5	10	1	7	6	9	5	2	3	8	4										
			26	17	24	12	22	13	25	21	16	23	18	11	29	14	19	30	27	20	28	15

Table A3. Optimal solution and corresponding cost for instances with $t = \lfloor n/4 \rfloor$.

Instance	n	Optimal cost	Optimal solution																			
LW11	11	7483.5	2	1																		
			10	7	5	3	8	9	4	6	11											
H12	12	13,685.0	3	1	2																	
			9	12	11	8	4	10	5	6	7											
Am12a	12	2054.0	1	2	3																	
			5	8	4	6	10	11	12	9	7											
Am12b	12	2490.5	1	3	2																	
			11	12	10	4	6	8	5	9	7											
Am13a	13	3083.5	1	2	3																	
			13	8	4	12	11	5	6	10	7	9										
Am13b	13	4401.0	1	2	3																	
			8	4	13	6	5	10	9	7	11	12										
H15	15	28,970.0	1	3	2																	
			10	4	15	6	14	13	5	8	12	9	11	7								
Am17	17	6409.0	1	3	2	4																
			17	7	15	8	12	14	11	16	10	6	5	13	9							
Am18	18	7680.5	1	3	4	2																
			17	7	15	8	12	11	14	16	18	10	5	6	13	9						
H20	20	10,973.0	1	3	2	4	5															
			17	18	10	7	12	14	11	16	8	15	20	6	19	13	9					
P24_a	24	18,757.0	4	2	5	6	3	1														
			16	9	21	13	10	11	7	22	23	18	19	8	15	17	24	12	20	14		
P24_c	24	23,908.0	5	2	6	3	4	1														
			9	21	13	16	10	11	7	22	19	8	12	24	15	17	23	18	20	14		
P24_d	24	28,411.0	2	5	6	4	3	1														
			21	13	9	10	16	12	7	24	22	18	19	8	11	15	17	23	14	20		
P24_e	24	3705.0	2	5	6	1	3	4														
			13	12	24	21	9	10	7	22	19	18	16	11	23	14	20	8	15	17		
N25_1	25	26,229.5	5	2	3	4	6	1														
			18	11	8	25	14	16	9	19	23	7	10	21	20	15	22	24	17	12	13	
N25_2	25	23,081.0	5	1	3	6	4	2														
			20	22	15	10	19	7	16	21	24	8	25	9	11	14	18	23	17	12	13	
N25_3	25	33,584.5	3	5	2	4	1	6														
			18	11	8	14	25	16	24	21	7	17	19	10	23	9	20	22	15	13	12	
N25_4	25	11,289.0	5	1	3	6	2	4														
			20	22	15	10	19	7	16	21	24	8	25	9	11	14	18	23	17	12	13	

Table A4. Optimal solution and corresponding cost for instances with $t = \lfloor n/5 \rfloor$.

Instance	n	Optimal cost	Optimal solution																			
LW11	11	7483.5	2	1																		
			10	7	5	3	8	9	4	6	11											
H12	12	16,545.0	1	2																		
			3	9	12	11	8	4	10	5	6	7										
Am12a	12	2410.0	1	2																		
			7	3	12	11	10	4	5	6	8	9										
Am12b	12	3172.5	1	2																		
			9	6	8	4	5	10	11	7	12	3										
Am13a	13	3821.5	1	2																		
			13	8	4	3	12	11	5	6	10	7	9									
Am13b	13	4656.0	1	2																		
			8	4	13	3	5	6	10	9	7	11	12									
H15	15	28,970.0	1	3	2																	
			10	4	15	6	14	13	5	8	12	9	11	7								
Am17	17	6892.0	1	3	2																	
			17	7	15	8	12	4	14	11	16	10	6	5	13							
Am18	18	8245.5	1	3	2																	
			17	7	18	15	4	14	12	8	11	16	10	5	6	13	9					
H20	20	12,272.0	1	3	2	4																
			19	10	18	15	14	16	11	12	8	20	7	6	5	17	13	9				
P24_a	24	21,729.0	2	4	3	1																
			5	18	22	23	11	8	19	21	7	9	10	13	16	6	15	17	24	12	20	14
P24_c	24	30,347.0	2	3	4	1																
			5	23	18	11	22	21	19	8	7	9	10	16	13	6	12	17	15	24	20	14
P24_e	24	33,061.0	2	3	4	1																
			5	21	20	14	18	23	22	19	8	11	7	9	10	16	24	12	6	13	15	17
N25_2	25	30,193.5	5	2	1	3	4															
			18	11	14	6	8	25	16	24	21	7	19	10	9	15	22	20	23	17	12	13
N25_3	25	23,167.0	3	5	2	4	1															
			12	18	16	25	11	24	17	21	14	6	8	7	19	23	10	9	20	22	15	13
N25_4	25	38,689.5	5	2	1	3	4															
			18	11	6	19	16	25	8	21	24	7	10	15	9	22	20	14	23	17	12	13
N25_5	25	12,951.0	5	2	1	3	4															
			18	11	14	6	8	25	16	21	24	7	19	10	9	15	22	20	23	17	12	13