The Aperiodic Facility Layout Problem with Time-varying Demands and a Master-Slave Solution Approach

Yiyong Xiao[1], Jiaoying Huang[1], Shenghan Zhou[1*], Yuchun Xu[2], Sadan Kulturel-Konak[3], Abdullah Konak[4]

[1]School of Reliability and System Engineering, Beihang University, Beijing 100191, China

[2]School of Engineering & Applied Science, Aston University, Birmingham, B4 7ET, United Kingdom

[3]Management Information Systems, Penn State Berks, Tulpehocken Road, P.O. Box 7009, Reading, PA 19610-6009, United States

[4]Information Sciences and Technology, Penn State Berks, Tulpehocken Road, P.O. Box 7009, Reading, PA 19610-6009, United States

Abstract: In many seasonal industries, market demands are constantly changing over time and accordingly the facility layout should be re-optimized timely in order to adapt to the expected material handling patterns between manufacturing departments. This paper investigated the aperiodic facility layout problem (AFLP) that involves arranging facilities layout and re-layout aperiodically in a given planning horizon in a dynamic manufacturing environment. The AFLP is decomposed into a master problem and a slave problem without loss of optimality in this paper, and all problems are formulated as mixed-integer linear programming (MILP) models that are directly solvable to MIP solvers for small-sized problems. An exact backward dynamic programming (BDP) algorithm is developed for the master problem of large-size with a computational complexity level of O(n^2), and an improved version of problem evolution algorithm based on linear programming (PEA-LP) is developed for the slave problem of all sizes. Computational experiments are conducted on two new provisioned problems and twelve benchmarked problems in literature, and the experimental results show that the proposed solution approach is promising for solving the AFLP with all sizes of problem instances, where five benchmark problems have found new best solutions by using the improved PEA-LP algorithm.

1. Introduction

The facility layout problem (FLP) plays an important role for efficient material handling in a manufacturing system. It involves optimizing the layout of a set of given manufacturing elements (e.g., machine tools, manufacturing cells, work centers, departments, etc.) with specified requirements (e.g., specified areas and patters of spaces) within a rectangular facility floor space without overlapping, in order to increase materials handing efficiency and minimize the total material-handling cost between departments over a given planning horizon (Koopmans and Beckmann, 1957; Armour and Buffa, 1963; Sahni & Gonzalez, 1976; Montreuil, 1990; Meller et al., 1999; Sherali et al., 2003; Castillo and Westerlund, 2005; Konak et al., 2006; Kulturel-Konak, 2012; Xie et al., 2018). The dynamic FLP (DFLP), introduced first by Rosenblatt (1986), is an important extension of the static FLP, seeking to optimize the layout and re-layout of departments over multiple planning periods in which the manufacturing departments may have different material-flow patterns due to changes in product demands and others (Montreuil and Venkatadri, 1991; Urban, 1993; Lacksonen, 1997; Kulturel-Konak and Konak, 2015; Ulutas and Islier, 2015; Xiao et al., 2017; Kulturel-Konak, 2019).

The main characteristic of today's manufacturing environments is volatility, requiring manufacturing systems to be agile and flexible in order to respond rapidly to market changes (Baykasoglu et al., 2006; Kulturel-Konak et al,

*Corresponding author. Tel.: +86-010-82316003
E-mail addresses: xiaoyiyong@buaa.edu.cn (Yiyong Xiao); Huangjy@buaa.edu.cn (Jiaoying Huang); Zhoush@buaa.edu.cn (Zhou Shenghan); y.xu16@Aston.ac.uk (Yuchun Xu); sadan@psu.edu(Sadan Kulturel-Konak), auk3@psu.edu(Abdullah Konak)

2004). In many seasonal industries such as garments and footwear industry, consumer demands, both in terms of product variety and quantity, change frequently over time. These constantly changing market demands consequently requires constantly change material flows among manufacturing departments. To maintain high operational efficiency and a low material-handling cost, it is, therefore, necessary to re-optimize the layout of a production plant in a timely manner to make the system agile. However, as the re-layouts of the facilities incur additional cost, the facilities should be properly managed in both WHEN and HOW to conduct layout and re-layout optimizations, e.g. aperiodically or periodically based on the accumulated changing of material-flow over time. The problem hereby is referred to as the aperiodic facility layout problem with time-varying demands (AFLP-TVD).

The traditional models for dynamic FLP (or periodic FLP) (Rosenblatt,1986; Lacksonen, 1994; Kulturel-Konak and Konak, 2015) involves only HOW to optimize the layout of facilities in a given time of period, without considering both when and how many times the layout/re-layouts should be carried out. This study is focused on joint optimization on both WHEN and HOW the layout/re-layouts should be carried out. The contributions of this paper includes:

(1) The AFLP-TVD is described for the first time as a mixed-integer linear programming (MILP) model.
(2) An efficient solution approach based on the master-slave framework was developed. In this framework, a master problem determines WHEN the re-layouts should be arranged, and a slave problem (i.e., the traditional static FLP) determines HOW the re-layouts should be done.
(3) An exact algorithm based on dynamic programming is developed for the master problem, and an improved Problem Evolution Algorithm (PEA) is developed for the slave problem.
(4) Rigorous computational experiments were conducted to study the efficiency and effectiveness of the proposed algorithms while new best solutions were found for many of the benchmark FLP problems.

The rest of the paper is organized as follows. In Section 2, the related literature is reviewed. In Section 3, the AFLP is formally described and formulated with a general formulation, and then it is decomposed as a master problem and a slave problem. In Section 4, a new algorithm based on dynamic programming and a heuristic algorithm based on PEA are developed for the slave problem. In Section 5, experiments are conducted to test and validate the proposed algorithms. Finally, conclusions are drawn in Section 6.

2. Related work reviews

2.1 Static facility layout problems

Material handling involves short-distance movements of raw/auxiliary materials, semi-finished/finished products, non-conforming items, tools, etc., transported on a conveyor or lift truck or other types of material handling equipment among the departments of a manufacturing system. The facility layout problem (FLP) is important for the efficient and low-cost operations of most manufacturing firms. Models and formulations for the FLP are abundant in literature and can be basically categorized into two types: (1) the static models that optimize the layout design for only one period, and (2) the dynamic models that optimize concurrently the layout design for multiple periods. The static FLP was first modeled by Koopmans and Beckmann (1957) as a quadratic assignment problem (QAP), seeking to find the optimal assignment of n departments to n predetermined locations toward minimization of material handling cost (MHC) expressed as the product of distances and material flows. The QAP is NP-Complete (Sahni & Gonzalez, 1976), and exact algorithms can be found for only small-sized instances (Lawler, 1963; Bazaraa and Elshafei, 1979). Armour and Buffa (1963) proposed an extended version of the QAP model with a grid-based block FLP and the well-known CRAFT algorithm, where departments have pre-specified shape requirements (fixed heights and widths), seeking to optimize the placement of a set of n departments in a grid-based rectangular facility divided into basic squares or rectangles having a unit area.

There are various models of the static FLP beside the QAP that were developed in the literature for different practical scenarios. Typical models are those for the unequal area FLP (UE-FLP), which involves placing a set of unequal-area blocks representing departments within a continuous rectangle plane without overlapping (Bazaraa, 1975; Tam, 1992; Montreuil, 1990; Lee and Lee, 2002; Dunker et al., 2005). The block UE-FLP with fixed departmental shapes considers the situation that all departments have unequal areas but fixed shapes (e.g., given heights and weights) and their layout arrangement involves only optimizing the department coordinates and their vertical or horizontal orientations while satisfying the non-overlap requirement at the same time (Baykasoglu et al., 2006; Drira et al., 2007, Balakrishnan and Cheng, 2009; McKendall and Hakobyan, 2010; and McKendall and Liu, 2012).

Another widely studied model is the block UE-FLP that considers a more general situation of facility layout with flexible departmental shapes. In this case, the side lengths of departments (e.g., heights l_i^x and widths l_i^y), in addition to their locations, are also considered as decision variables to take any continuous values as long as the departmental area is guaranteed ($a_i = l_i^x \times l_i^y$) with some common constraints on department shape, such as the aspect-ratio or minimum side-length, being satisfied (Montreuil 1990; Bozer and Meller, 1997; Meller et al., 1999; Sherali et al., 2003; Konak et al., 2006; Ulutas and Kulturel-Konak, 2012; Kulturel-Konak and Konak, 2013, 2015; Gonçalves and Resende, 2015; Xiao et al., 2017; Xie et al., 2018). This version of the UE-FLP is categorized as the FLP on the continuous plane (Montreuil, 1990; Kulturel-Konak and Konak, 2013) and is even more difficult to solve due to two additional continuous variables (l_i^x and l_i^y) and their non-linear relations to the required areas. Relevant research on the linearization of the nonlinear relationship can be found in Montreuil (1990), Bozer and Meller (1997), Meller et al. (1999), Sherali et al. (2003), Gonçalves and Resende (2015), and Xiao et al. (2017).

The unequal area FLP with flexible bay structure (FLP-FBS) is a special case of the UE-FLP often confronted in multi-bay environments where the departments need to be arranged in parallel-bays with varying widths (Meller, 1997; Chae and Peters, 2006; Konak et al., 2006). In this situation, the departments must be aligned in parallel-bays with an equal width to the bay but flexible lengths/heights to satisfy area requirements. Several formulations and heuristic approaches for the unequal area FLP-FBS can be found in literature, such as ant system algorithm (Wong and Komarudin, 2010; Komarudin and Wong, 2010), artificial immune systems (Ulutas and Kulturel-Konak, 2012), and probabilistic tabu search (Kulturel-Konak, 2012).

The single row facility layout problem (SR-FLP) is another widely studied model in applications involving the arrangement of a set of departments along a straight line such as in supermarkets, hospitals or offices (Simmons, 1969; Amaral, 2006). It is also applicable to the arrangement of books on a shelf and manufacturing systems design as well in order to minimize transportation costs (Picard and Queyranne, 1981; Heragu and Kusiak, 1988). The SR-FLP is NP-hard (Beghin-Picavet and Hansen, 1982), and exact algorithms exist only for small-sized problems such as the branch-and-bound (Simmons, 1969) and dynamic programming (Picard and Queyranne, 1981). Recent state-of-art heuristics for RS-FLP can be found in Datta et al. (2011), Hungerländer and Rendl (2012), Guan and Lin (2016), Rubio-Sánchez et al. (2016), and Cravo and Amaral (2019).

There are a number of other variants of the static FLP in literature, including the zone-based FLP where the facility needs to be divided into several zones with pre-defined relative position relations before the departments being arranged optimally in zones (Montreuil et al., 2002, 2004; Xiao et al., 2017, Kulturel-Konak 2019), the multi-objective FLP that considers multiple objective functions in addition to the material handling cost, such as the number of devices and the average WIP (Saraswat et al., 2015; Liu and Liu, 2019), the multi-floor FLP that considers the floor having been separated into multiple sub-floors by inner structure walls, passages, or fixed split lines (Lee et al., 2005; Meller and Bozer, 1997; Ahmadi et al., 2017), the FLP with input/output station design that associates each department with input (and output) stations from which the materials are received or sent to other interactive departments (Montreuil and Ratliff, 1988, 1989; Montreuil et al., 1993; Arapoglu et al., 2001; Friedrich et al., 2018), and the FLP with heterogeneous distance metrics that considers a mixed use of metrics such as Rectilinear, squared Euclidean, and Tchebychev distance, and Contour distance metrics for measuring the material moving distances between departments (Ozdemir et al., 2003; Hale et al., 2012; Niroomand and Vizvári, 2013; Paes, 2017; Xie et al., 2018).

2.2 Dynamic facility layout problems

In a time-varying market environment, the product demands/volumes of a manufacturing system may suffer from constantly changing. To adapt to this situation, the manufacturing systems have to be updated over time and the facility layout needs to be re-optimized dynamically in order to maintain high operational efficiencies and low material handling costs. This dynamic nature of the FLP was first recognized by Hitchings (1970) who pointed out that a change in a layout system is appropriate when the cost of the change is less than the savings which would accrue due to an increased efficiency resulting from the change. Rosenblatt (1986) first introduced the dynamic facility layout problem (DFLP) that was an important extension of the FLP considering optimizations of the layout manufacturing units over multiple planning periods. The proposed DFLP model is based on the QAP formulation with the objective function includes the total material handling cost in all periods and the re-layout costs between consecutive periods.

Montreuil and Venkatadri (1991) presented a proactive methodology for the dynamic layout optimization of a manufacturing system with multiple expansion (or decline) phases, including the initial layout, tactically intermediate layouts, and final strategic layout at the maturity level. They provided a linear programming model for generating optimal layouts for all expansion phases of the expansion plan of a manufacturing firm from the initial phase to the final mature scenario. Urban (1993) developed a heuristic based on the steepest-descent pairwise-interchange procedure for the QAP version of the DFLP, where the material handling cost could be optimized based on varying lengths of forecast windows and the corresponding rearrangement costs were included. Lacksonen and Enscore (1993) modeled the DFLP as a modified QAP by assuming unit department sizes and pointed out that the cutting plane was the best algorithm, outperforming other four alternative algorithms under comparing, including CRAFT, branch and bound, dynamic programming, and cut trees, based on a series of realistic test problems. Lacksonen (1994) developed a two-step algorithm for static and dynamic layout problems with unequal department areas, using a heuristic cutting plane routine for solving a QAP formulation and a mixed-integer linear programming model for optimizing the block diagram layout. Lacksonen (1997) improved the model and algorithm by using a preprocessing method developed to pre-specify certain obvious department pair orientations and therefore achieved significant cost reductions and computational efficiency improvements.

In recent decades, various solution approaches and algorithms have proposed for the DFLP. Balakrishnan et al. (2003) developed a hybrid genetic algorithm with several effective heuristics for solving the large-sized DFLP under QAP formulation. Baykasoglu et al. (2006) first formulated the budget-constrained DFLP where the accumulated cost for shifting of facilities in each period should not exceed pre-specified budget limitations and developed an ant colony optimization (ACO) algorithm to solve the model with budget constraints. Balakrishnan and Cheng (2009) investigated the DFLP with rolling planning horizons containing demand uncertainties and conducted experiments to show the differences from the traditional settings for DFLP with fixed planning horizons and no forecast errors. Kulturel-Konak and Konak (2015) studied the cyclic facility layout problem (CFLP), a special case of the DFLP assuming that the production cycle repeats itself by going back to the first period after the last one, and the authors developed a large-scale hybrid simulated annealing algorithm (LS-HSA) which is applied to various facility layout problems. Ulutas and Islier (2015) studied the DFLP of a footwear facility and several scenarios generated by the real-life data, and they proposed a clonal selection based algorithm to the real-life DFLP. Kulturel-Konak (2019) proposed a matheuristic approach combining tabu search (TS) and mathematical programming to solve the zone-based DFLP on the continuous plane with unequal area departments. Xiao et al. (2017) developed a general formulation of the zone-based unequal-area DFLP and proposed a new meta-heuristic algorithm called problem evolution algorithm (PEA) for facility layout problems. Xiao et al. (2019) modeled the unequal-area DFLP with pick-up and drop-off locations, and they developed an improved particle swarm optimization (PSO) algorithm as the solution approach. Other heuristic approaches for the DFLP existing in literature include the simulated annealing algorithm (McKendall et al., 2006), tabu search heuristics (McKendall and Liu, 2012), hybrid particle swarm optimization (Hosseini-Nasab and Emami, 2013), hybrid ant colony optimization (Chen, 2013), heuristic Wang-Landau (WL) sampling algorithm (Liu et al., 2017), and hybrid heuristic algorithm based on bacterial foraging optimization (Turanoğlu and Akkaya, 2018).

All reviewed works above aim to optimize the layouts of a facility in multiple periods, and occasionally introduced some individual settings such as resource limitation, cycling periods, and input/output stations for practical needs. Most of the existing DFLPs were still modeled based on the static FLP model by extending it from a single period to multiple periods with additional re-layout costs between consecutive periods added to the objective function. Some critic issues of dynamic layout such as when the optimization and re-optimizations should be carried out and how many times they should be arranged in a long planning horizon were still not involved. This is particularly important in an uneven time-varying manufacturing environment but could be found with few relevant types of research in literature.

3. Problem description and formulation

The AFLP-TVD is described as follows. A production plant has a set, N, of n blocks of departments (e.g., machine tools, manufacturing cells, work centers, storage units, departments, etc.) with specified area/shape requirements to be placed within a rectangular facility without overlapping. The material-flows among the departments change over time in response to the time-varying demands of products. Thus, the entire planning horizon can be divided into a set, T, of multiple short production periods, and for each period $t \in T$, the material-flow volume, f_{ijt}, between a department pair (i,j), $i, j \in N$, is constant. The production plant needs an initial layout and may require re-layouts in the following periods in order to minimize the material handling cost (MHC), which is expressed as the sum of material-flow volumes weighted by the distances between departments. However, the re-layouts also lead to production disruptions that result in the re-layout cost (RLC) including such as profit loss, fixed wages, and rental fee. Thus, re-layouts may be needed in some periods after the first one in order to reduce the total MHC, but they cannot be very frequent due to the RLC. In this paper, the RLC, C_t, in period t is considered as an overall estimated fixed cost related to the period t. Thus, the objective of the AFLP-TVD is to determine an initial layout at the first period and find out a subset of periods in which re-layouts are performed to re-optimize the material flows toward minimizing the total MHC and RLC occurred during the entire planning horizon.

3.1 A general formulation of the AFLP-TVD

Below, we provide a general formulation of the AFLP-TVD using a mixed integer linear programming model (MILP). The parameters and decision variables used in the MILP model are summarized as follows.

Parameters:

T	set of period, where m = card (T)
t, k	time period index, where $t, k \in T$
C_t	relayout cost if a re-layout occurs in period t
N	set of departments, n = card (T)
i,j	index of departments, $i, j \in N$
x, y	axis directions
e	an axis direction, $e \in \{x, y\}$
B^x, B^y	side length of the rectangle facility in the x and y axis directions
f_{ijt}	material flow between department i and j in period t.
a_i	minimum area requirement of department i
α_i	maximum aspect ratio of department i (the ratio of the longer side to the shorter)
M	a large number

Decision Variables:

s_t	binary variable indicating whether a re-layout is set in period t (s_t = 1) or not (s_t = 0) for $t \in T$
c_{it}^x, c_{it}^y	centroid coordinate of department i in period t in the x and y axis directions
l_{it}^x, l_{it}^y	side length of department i in period t in the x and y axis directions
d_{ijt}^x, d_{ijt}^y	distance between the centroids of departments i and j in period t in the x and y axis directions
z_{ijt}^x, z_{ijt}^y	binary variables denoting the relative locations of departments i and j in period t in the x and y axis directions such that

(1) z^x_{ijt} = 1 and z^y_{ijt} = 1: department i is on the left side of department j (in the x axis direction)

(2) z^x_{ijt} = 0 and z^y_{ijt} = 0: department i is on the right side of department j (in the x axis direction)

(3) z^x_{ijt} = 0 and z^y_{ijt} = 1: department i is on the bottom side of department j (in the y axis direction)

(4) z^x_{ijt} = 1 and z^y_{ijt} = 0: department i is on the top right side of department j (in the y axis direction)

v_t representing a variable in period t, e.g., c_{it}^x, c_{it}^x, l_{it}^x, l_{it}^y, d_{ijt}^x, d_{ijt}^y, z_{ijt}^x, z_{ijt}^y, where $t \in T$

Objective function:

The objective function is to minimize the total cost including the MHCs occurred in all periods and the RLCs occurred in the re-layout periods. Thus, the AFLP-TVD can be formulated as MILP model as follows.

***Problem** AFLP-TVD:*

$$\min Total\ Cost = \sum_{t \in T} (s_t C_t + \sum_{i,j \in N; i<j} f_{ijt}(d_{ijt}^x + d_{ijt}^y)) \tag{1}$$

s.t.

(1) $s_1 = 1$

(2) $|v_t - v_{t-1}| \le M \cdot s_t \qquad \forall t > 1$

(3) $$\begin{cases} c_{it}^x + 0.5 l_{it}^x \le c_{jt}^x - 0.5 l_{jt}^x + M(2 - z_{ijt}^x - z_{ijt}^y) \\ c_{jt}^x + 0.5 l_{jt}^x \le c_{it}^x - 0.5 l_{it}^x + M(z_{ijt}^x + z_{ijt}^y) \\ c_{it}^y + 0.5 l_{it}^y \le c_{jt}^y - 0.5 l_{jt}^y + M(1 + z_{ijt}^x - z_{ijt}^y) \\ c_{jt}^y + 0.5 l_{jt}^y \le c_{it}^y - 0.5 l_{it}^y + M(1 - z_{ijt}^x + z_{ijt}^y) \end{cases} \qquad \forall i \ne j, t$$

(4) $d_{ijt}^e \ge |c_{it}^e - c_{jt}^e| \qquad \forall i, j, t, e$

(5) $a_i = l_{it}^y \cdot l_{it}^x \qquad \forall i, t$

(6) $$\begin{cases} l_{it}^x \le \alpha_i \cdot l_{it}^y \\ l_{it}^y \le \alpha_i \cdot l_{it}^x \end{cases} \qquad \forall i, t$$

(7) $$\begin{cases} c_{it}^e - 0.5 l_{it}^e \ge 0 \\ c_{it}^e + 0.5 l_{it}^e \le B^e \end{cases} \qquad \forall i, t, e$$

(8) $c_{it}^e \ge 0, l_{it}^e \ge 0, d_{ijt}^e \ge 0, s_{ijt}^e \in \{0,1\} \qquad \forall i, j, t, e$

In the above model, Constraint (1) ensures that an initial layout must be arranged in the first period. Constraints (2) guarantee that for period $t \in T$ and t >1, if s_t = 1, then all other decision variables for period t, including c_{it}^x, c_{it}^x, l_{it}^x, l_{it}^y, d_{ijt}^x, d_{ijt}^y, z_{ijt}^x, z_{ijt}^y and represented by notation v_t, must take the same value with the previous period, t-1. Constraints (3) prevent overlapping of departments which were first introduced by Montreuil (1990). Constraints (4) determine the rectilinear distance between departments i and j. Constraint (5) represents the department area requirements, which is nonlinearly expressed for the brevity of the model presentation. This constraint can be linearized through the tangent-line based approximation method of Castillo et al. (2005) or the secant-line based approximation method of Xiao et al. (2017). Constraint (6) restricts the aspect-ratio of the departments not to exceeding the given values. Constraint (7) ensures all departments are arranged within the rectangle facility. Constraint (8) defines the value domains of decision variables.

Solving the AFLP-TVD model formulated in Eq.(1) and Constraints (1)-(8) with optimality is extremely difficult because of the large number of the binary decision variables involved. In the following sections, we decompose the problem as a master-problem and slave-problem such that the computational complexity is significantly reduced without losing the optimality.

3.2 Formulation of the master-problem

Let F_{kt} denote the optimal total MHC for the periods between t and k such that a new layout is determined in period t and used in the following periods from t+1 to k without layout change. In the master-problem, F_{kt} is considered an input parameter, and its values are pre-calculated by the static FLP which acts as the slave-problem formulated in Section 3.3. Thus, the master problem only determines the re-layout settings, which is formulated as a master-MILP model as follows.

Parameters:

- T a set of periods, where m = card (T)
- t, k index of time periods, where $t,k \in T$
- C_t cost of production if the facility is gone through a re-layout in period t
- F_{tk} the optimal total MHC of the periods from t and k, where $t \leq k$, when the facility layout is rearranged in period t and used in periods from t+1 to k without layout change.

Decision variables:

- s_t binary variable representing the re-layout decisions during the planning horizon such that s_t = 1 if a re-layout takes place in period t, and s_t = 0 otherwise.
- u_{tk} binary variable indicating the last period k covered by the re-layout in period t such that u_{tk} = 1 if period k is the last period covered by the re-layout in period t, and u_{tk} = 0 otherwise.

Master-problem AFLP-TVD:

$$\text{Min. } Total\ Cost = \sum_{t\in T} C_t s_t + \sum_{t,k\in T; t\leq k}^{n} F_{tk} u_{tk} \tag{2}$$

s.t.

(9) $s_1 = 1$

(10) $\sum_{k=t}^{n} u_{tk} = s_t \qquad \forall t \in T$

(11) $u_{t,k-1} \leq s_k \qquad \forall t,k \in T; k > 1$

(12) $\sum_{t\in T} u_{tn} = 1$

(13) $s_t \in \{0,1\}; u_{t,k} \in \{0,1\} \qquad \forall t,k \in T$

In the above formulation, the first term of the objective function is the re-layout cost, and the second term is the material-handling cost. Constraint (9) ensures that an initial layout must be set in the first time period. Constraint (10) ensures that the last period covered by a re-layout in period t (t>1) should be t or after t. Constraint (11) ensures that the last period covered by a re-layout in period t (t>1) must be immediately followed by a re-layout. Constraint (12) ensures the last period (i.e., period m) must be covered by a re-layout optimization. Constraint (13) defines the value domains of decision variables. The following constraints are not necessary for the integrity of the master-problem AFLP-TVD, but they are valid and may help to improve the computational efficiency or develop heuristic rules.

(14) $\sum_{k=1}^{t-1} u_{tk} = 0 \qquad \forall t \in T, t > 1$

(15) $u_{tk} \le s_t \qquad \forall t, k \in T$

(16) $u_{tk} = 0 \qquad \forall t, k \in T; k < t$

Constraint (14) forces that the last period covered by a re-layout in period t (t>1) should not be before t, Constraint (15) forces u_{ik} to be zero if $s_t = 0$, and Constraint (16) allows variable u_{ik} to take positive values only at the right-top half of its value matrix, which plays the same role as Constraint (15) but with simpler expression.

Since the master-problem involves determining a set of optimal re-layout settings in m periods through the decision variable s_t, and variable u_{tk} depends on s_t, the problem complexity can be recognized as exponential, i.e., $O(2^{m-1})$. In Fig. 1, we provide an example of a feasible solution to illustrate the relationship between decision variables s_t and u_{tk}. In this example, the planning horizon has 9 periods (e.g.,9 weeks), and an initial layout is set in period 1 and re-layouts are set in periods 4, 5, and 8, as shown by variable s_t. The values for variable u_{tk} are accordingly determined by variable s_t. The objective value includes the re-layout cost occurred in periods 1, 4, 5, and 8, and the material-handling costs indicating by F_{13}, F_{44}, F_{57}, and F_{89}.

Variable s_t:

t	1	2	3	4	5	6	7	8	9
s_t	1	0	0	1	1	0	0	1	0

Variable u_{tk}:

t \ k	1	2	3	4	5	6	7	8	9
1	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	0

Fig. 1 A feasible solution example of the AFLP-TVD problem

3.3 Formulation of the slave-problem

The slave-problem is a static FLP to calculate the optimal values of F_{tk}. For each pair of (t, k), where $t, k \in T$ and $t \le k$, the material flows in all periods from t to k are combined to find the best layout that minimizes the total material cost between periods t and k. The static FLP is formulated as a mixed-integer linear programming (MILP) model as follows (Montreuil 1990, Konak et al., 2006; Kulturel-Konak & Konak, 2013).

Parameters:

N	set of departments
i,j	index of department and $i, j \in N$
x, y	axis directions
e	an axis direction, $e \in \{x, y\}$
B^x, B^y	side length of the rectangle facility building floor in the x and y axis directions
f_{ij}^{tk}	material flow between departments i and j during periods from t to k, $f_{ij}^{tk} = \sum_{t'=t}^{k} f_{ijt'}$
a_i	minimum area requirement of department i
α_i	maximum aspect ratio of department i (the ratio of the longer side to the shorter)
M	a large number

Decision Variables:

c_i^x, c_i^y	centroid coordinates of department i in the x and y axis directions
l_i^x, l_i^y	side length of department i in the x and y axis directions
d_{ij}^x, d_{ij}^y	distance between the centroids of departments i and j in the x and y axis directions
z_{ij}^x, z_{ij}^y	binary variables denoting the relative locations of departments i and j in the x and y axis direction, indicating that

(1) $z^x_{ij} = 1$ and $z^y_{ij} = 1$: department i is on the left side of department j (in the x axis direction)

(2) $z^x_{ij} = 0$ and $z^y_{ij} = 0$: department i is on the right side of department j (in the x axis direction)

(3) $z^x_{ij} = 0$ and $z^y_{ij} = 1$: department i is on the bottom side of department j (in the y axis direction)

(4) $z^x_{ij} = 1$ and $z^y_{ij} = 0$: department i is on the top right side of department j (in the y axis direction)

Sub-problem to calculate F_{tk}:

$$\min F_{tk} = \sum_{i,j \in N; i<j} f_{ij}^{tk} \cdot (d_{ij}^x + d_{ij}^y) \tag{3}$$

s.t.

$$(17) \quad \begin{cases} c_i^x + 0.5 l_i^x \le c_j^x - 0.5 l_j^x + M(2 - z_{ij}^x - z_{ij}^y) \\ c_j^x + 0.5 l_j^x \le c_i^x - 0.5 l_i^x + M(z_{ij}^x + z_{ij}^y) \\ c_i^y + 0.5 l_i^y \le c_j^y - 0.5 l_j^y + M(1 + z_{ij}^x - z_{ij}^y) \\ c_j^y + 0.5 l_j^y \le c_i^y - 0.5 l_i^y + M(1 - z_{ij}^x + z_{ij}^y) \end{cases} \qquad \forall i < j$$

$$(18) \quad d_{ij}^e \ge |c_i^e - c_j^e| \qquad \forall i < j, e$$

$$(19) \quad a_i = l_i^y \cdot l_i^x \qquad \forall i$$

$$(20) \quad \begin{cases} l_i^x \le \alpha_i \cdot l_i^y \\ l_i^y \le \alpha_i \cdot l_i^x \end{cases} \qquad \forall i$$

$$(21)\quad \begin{cases} c_i^e - 0.5 l_i^e \geq 0 \\ c_i^e + 0.5 l_i^e \leq B^e \end{cases} \qquad \forall i, e$$

$$(22)\quad c_i^e \geq 0, l_i^e \geq 0, d_{ij}^e \geq 0, z_{ij}^e \in \{0,1\} \qquad \forall i, j, e$$

In the above formulations, the objective is to minimize the total MHC expressed as a product of the material flow and rectilinear distance among the departments. Constraints (17) prevent the departments from overlapping, which was first introduced by Montreuil (1990). Constraint (18) bounds the rectilinear distance between departments i and j. Constraint (19) represents the department area requirements, which are nonlinear for the brevity of model presentation. In this paper, the nonlinear area requirement constraints are linearized through the tangent-line based approximation method of Castillo et al. (2005) or the secant-line based approximation method of Xiao et al. (2017). Constraint (20) restricts the aspect-ratio of the department side lengths within the given requirement. Constraint (21) ensures that all departments are located within the boundaries of the facility. Constraint (22) defines the domains of the decision variables.

In the literature, there are several algorithms that were developed for solving the static FLP. However, calculating all F_{tk} values is computationally expensive especially for problems with a large number of planning periods. Fig. 2 provides a fast algorithm for calculating F_{tk} efficiently. In the algorithm, parameter R_t represents the nearest period after period t such that the next re-layout must be between $[t+1, R_t]$ if a re-layout has been performed in period t. Initially, R_t is set to the default value of m. In Step 7), the algorithm assigns R_t with a value k such that the total cost after setting two re-layouts in periods l and k (one for re-layout and one for restoring) is smaller than F_{tk}. That means there must be a re-layout between periods t-1 and k in the optimal solution. Thus, for each period t, we just need to calculate F_{tk} for all $k \in [t, R_t]$ instead of for all $k \in [t, m]$. After calculating F_{tk} values once, they are used by the dynamic programming algorithm described in Fig. 3 in Section 3.4.

1) **For** $t = m, m\text{-}1, m\text{-}2, \ldots, 1$ **Do Begin**
2) Let $R_t \leftarrow m$
3) Solve the MILP in Eq.(3) and Constraints (17)-(22) to get F_{tm}
4) **For** $k = t, t+1, \ldots, m\text{-}1$ **Do Begin**
5) Solve the MILP in Eq.(3) and Constraints (17)-(22) to get F_{tk}
6) **For** $l = t+1, \ldots, k\text{-}1$ **Do Begin**
7) **If** $F_{tk} > F_{t,l-1} + C_l + F_{l,k} + C_k$ **Then** Let $R_t \leftarrow k$
8) **If** $R_t < m$ **Then** Break
9) **End For**
10) **If** $R_t < m$ **Then** Break
11) **End For**
12) **End For**

Fig. 2 A fast procedure for calculating F_{tk} and R_t

4. Solution approaches

4.1 A dynamic programming approach to the master problem

The master-problem defined by Eq.(2) and Constraints (10)-(14) has an exponential complexity of $O(2^m)$ since the main decision variable is s_t. In this sub-section, we present an exact backward dynamic programming (BDP) procedure, which has only a polynomial complexity of $O(m^2)$ in the worst case, to solve the problem. In BDP, P_t represents the sub-problem of the original master problem that covers only the periods from t to m for all $t \in T$, and h_t denotes the optimal objective function value of P_t. Thus, the original master problem is represented by P_1 and the last FLP problem for period m is represented by P_m. Obviously, we have $h_m = C_m + F_{mm}$, and h_1 can be obtained by the BDP recursive equations described as follows.

$$h_t = \begin{cases} C_m + F_{mm} & \forall t = m \\ \min\{C_t + F_{tm}, \min\{C_t + F_{t,t'} + h_{t'} \mid t' \in [t+1, R_t]\}\} & \forall t = m-1, m-2, \ldots, 1 \end{cases} \tag{4}$$

The above BDP formulation determines the values of h_t backwardly from $t = m$ to $t = 1$, and for each period t, the lowest value of $C_t + F_{t,t'-1} + h_{t'}$ for $t' \in [t+1, R_t]$ is calculated for h_t. Thus, the BDP guarantees that the final value, i.e., h_1, is an optimal solution of the original master-problem. The DBP algorithm has a computational complexity of $O(m^2)$ for the worst case. A detailed description of the BDP algorithm is shown in Fig. 3 as follows.

```
Call the fast procedure in Fig.2 to calculate F_tk and R_t for t ∈ T, k ∈ T, and t ≤ k
Let h_m ← C_m + F_mm                    //for last period m
Let g_m ← null                          //no re-lay is needed after period m
For t = m-1, m-2, …, 1 Do Begin         // determine h_t and g_t reversely starting from t = m-1
   Let h_t ← C_t + F_t,m
   Let g_t ← null
   For t' = t+1, t+2, …, R_t Do Begin
      If C_t + F_t,t'-1 + h_t' < h_t Then
         Let h_t ← C_t + F_t,t'-1 + h_t'
         Let g_t ← t'
      End If
   End
End
Output h_1                              //Output the optimal solution
Let t_1 ← 1, t_2 ← g_1
Do while t_2 is not null Begin
   Output t_1                           //Output the re-layout period that covers [t_1, t_2 -1]
   Let t_1 ←t_2,  t_2 ← g_t             //get next period for re-layout
End
Output t_1                              //Output the last re-layout period
```

Fig. 3 The detailed BDP algorithm

4.2 An improved problem evolution algorithm with linear programming for the static FLP

The *Problem Evolution Algorithm* (PEA), introduced in Xiao el al. (2017), is a new general approach for good quality solutions to a complex problem through the control of a designed evolutionary process. The PEA is a mimic of human's learning process where a person is always first trained with simple problems, and then based on the learned experiences, he/she will gradually be able to solve problems with increasing complexity and finally solve the most difficult problem. The general framework of the PEA is shown in Fig. 4 as follows.

1) Define a serial of evolution problems $\{P_1, P_2, \ldots, P_n\}$ with gradually increasing complexity, where P_1 is the simplest problem and P_n is the original problem.
2) Solve P_1 to find the optimal solution π_1
3) For $x = 2$ to n Begin
4) Based on π_{x-1}, construct a feasible solution π_x for P_x
5) Implement a local search to improve π_x
6) End For
7) Return π_n as the final solution for problem P_n

Fig. 4 A general framework of the PEA

Xiao et al. (2017) defined the PEA-LP algorithm that combines the PEA with linear programming (LP) to solve the dynamic facility layout problem. In their approach, sub-problem P_x includes only x departments to be placed in the layout, and the PEA-LP algorithm solves a series of sub-problems by increasing the department number x gradually from 1 to n one at a time. The departments to be included in successive sub-problems are selected randomly based on an evolution index. Xiao et al. (2017) used randomized material-flows (RMF), i.e. $\rho_i \sum_j f_{ij}, \forall i \in N$ where ρ_i is a random number, as the evolution index to construct a series of evolutionary sub-problems with an increasing number of departments. In this paper, we improve the PEA-LP algorithm on two aspects: (1) the weighted RMF that considers the square-root of department areas, i.e., $\rho_i \sqrt{a_i} \sum_j f_{ij}, \forall i \in N$, instead of the original evolution index, and (2) a dynamic iterative neighborhood search (DINS), as described in Fig.5, for a more efficient local search procedure in the PEA.

//The local search procedure for problem P_x

Procedure DINS ($P_{\max}$)

1) Let $P \leftarrow 0, \Delta \leftarrow 6$
2) **Repeat**
3) Calculate the selection probabilities of the departments in terms of the selection policy
4) Select a number Δ of the departments in terms of their selection probabilities
5) Fix RPRs of all departments
6) Unfix RPRs of the selected Δ departments
7) Call a MIP solver to optimize the unfixed RPRs
8) **If** the CPU time used in Step (7) is greater than 10 seconds **Then** let $\Delta \leftarrow \Delta$-1
9) **If** the CPU time used in Step (7) is less than 0.2 seconds **Then** let $\Delta \leftarrow \Delta$+1
10) **If** the solution is improved **Then** let $P \leftarrow 0$
11) **Else** let $P \leftarrow P$+1
12) **Until** $P = P_{\max}$

Fig. 5 The dynamic iterative neighborhood search

In the above DINS procedure, a partial solution π_x is repeatedly improved by solving the problem only for a subset of the departments included in P_x. The parameter $P_{\max}$ acts as the stopping condition indicating that the DINS procedure stops after a maximum $P_{\max}$ of successive attempts without improvement on the incumbent solution. The parameter Δ is an integer number representing the number of departments to be selected for each round of partial optimization. We first let $\Delta \leftarrow 6$ (in Step 1) such that only six departments (out of n departments) will be selected. However, the parameter Δ will be dynamically adjusted by $\Delta \leftarrow \Delta$-1 and by $\Delta \leftarrow \Delta$+1 in Steps 8 and 9, respectively, if the CPU time used in Step 7 is larger than a maximum threshold, e.g., more than 10 seconds, or smaller than a minimum threshold, e.g., less than 0.2 seconds. Such dynamic adjustment of parameter Δ limits the CPU time consumed for each round of partial optimization within a proper range. Therefore, the local search has an overall high computational efficiency. In Step 3, a *selection policy* is adopted in advance for generating the selection probabilities of the departments. Three candidate policies are introduced, named as (1) Random Selection (RS), (2) Frequency-Priority (FP), and (3) Material-Flow-Priority (MFP). The RS policy is the simplest one under which all departments are assigned with equal probability to be selected. The FP policy considers the number of times that the departments have been already selected and gives higher probabilities to those departments that have been selected with fewer times. The MFP policy assumes that the departments with a larger amount of material-flow (with other departments) are more important and therefore should be assigned to larger selection probabilities. The selection probality is calculated by Eq.(5) as follows.

$$p_i = \upsilon_i / \Gamma \qquad \forall i \in N, \tag{5}$$

where N is the set of elements from which one element is to be selected, υ_i is the value for element i (i.e., 1, reciprocal of selected times, and weighted material flow for policies RS, FP, and MFP, respectively), and $\Gamma = \sum_{j \in N} \upsilon_j$.

In the computational experiment section, we compared the performances of these three policies using different problem instances.

5. Computational experiments

In this section, we tested the proposed models and algorithms of the problem AFLP-TVD on several benchmark problems from the literature and compare our results with the existing best-known solutions. The MIP solver CPLEX (version 12.6.0.1) was used to solve benchmark problem instances, and the algorithms were coded in the AMPL environment. Computational experiments were conducted on a PC server with two 2.30 GHz Intel@ Xeon(R) CPUs (32 cores) and 110 GB memory.

5.1 Tested problem instances

The problem instances for testing the AFLP-TVD model are listed in Table 1, where M4 and M5 were generated in this paper, O7, O8, and O9 were originally from Meller et al. (1999), and F10 were from Montreuil et al. (2004), respectively. We associated each of the small-sized instances M4, M5, O7, O8, O9, and F10 with 48 periods, in which the first period has the original material-flow patterns and the following periods have randomly generated time-varying material-flow patterns. The problem instances BA12, BA13, BA14, M11*, M14*, M25*, SC30, and SC35, which were originally from Bazaraa (1975), Meller (1992) and Liu and Meller (2007), were used to test the performance of the improved PEA-LP algorithm for larger-sized static FLP problems.

Table 1. Summary of parameters of test problems.

Problem ($n \times T$)	Period Number	Dimensions	Number of Departments	Shape Constraint	Data reference
M4	48	10×12	4	$\alpha = 4$	This study
M5	48	11.5×12	5	$\alpha = 4$	
O7	48	13×8.54	7	$\alpha = 4$	Meller et al. (1999)
O8	48	13×11.31	8	$\alpha = 4$	
O9	48	13×12	9	$\alpha = 4$	
F10	48	90×95	10	$\alpha = 3$	Montreuil et al. (2004)
BA12	1	10×6	12	l^{min}=1	Bazaraa (1975)
BA13	1	9×7	13	l^{min}=1	
BA14	1	9×7	14	l^{min}=1	
M11*	1	6×6	11	α=4	Meller (1992) and Liu and Meller (2007)
M15*	1	15×15	15	α=5	
M25*	1	15×5	25	α=5	
SC30	1	15×12	30	α=5	Liu and Meller (2007)
SC35	1	16×15	35	α=4	

5.2 Testing the optimality of the AFLP model and the DBP procedure

First, we solved the MILP model formulated for the sub-problem described by Eq.(3) and Constraints (17)-(22) using CPLEX to calculate F_{tk} for t = 1, 2, …, n, and k = t+1, t+2, …, n for each of the small-sized instances as given in Table 2. After that, the BDP algorithm described in Fig.3 was used to find the optimal solutions of the test instances. The computational results were listed in Table 2 in the last column (i.e., T=48). To illustrate the effect of T on solution time, for each instance we also provide the optimal results for sub-problems containing only the first 3, 6, 12, 24, and 36 periods, indicated by T=3, T=6, T=12, T=24, and T=36, respectively, in Table 2. It can be observed that the computational efficiency dropped dramatically as the numbers of periods or departments increase. For all test instances, the major part of the computational time was due to CPLEX for computing F_{tk} values while the BDP procedure used less than 1 second.

Table 3 presents the extent to which the fast procedure described in Fig.2 can reduce the computational burden for calculating F_{tk} by comparing the actual number of times that the static FLP model was solved to the total times of full combinations. For problem instance F10 with 48 periods requires solving the static FLP model for a total of 1176 times, but the fast procedure was able to calculate F_{tk} value using only 232 MIP solutions, a reduction of 80.3% in computational cost without losing optimality. It can be generally recognized that a higher rate of computational time saving could be observed for larger numbers of planning periods. The average time saving rates of all tested instances for T=6, T=12, T = 24 and T=48 are respectively 4.0%, 26.3%, 53.6%, 67.0%, and 73.8%. In Table 2, the column *AVG time* indicates the average CPU time used to solve the static FLP model with optimal solutions using CPLEX. Notably, since the values of F_{tk} do not have any interdependency on one another, they can be computed in a parallel way, and thus the time span can significantly be reduced.

Table 2. Computational results of the BDP algorithm

	T=3		T=6		T=12		T=24		T=36		T=48	
Problem	Obj.	T.	Obj.	T.	Obj.	T.	Obj.	T.	Obj.	T.	Obj.	T.
M4	**1580.15**	<1s	**2813.97**	3s	**6122.22**	7s	**12832.74**	20s	**19646.20**	32s	**25778.27**	42s
M5	**3398.78**	2s	**6578.22**	11s	**13347.12**	31s	**27045.97**	82s	**40043.93**	2.2m	**53792.92**	3.0ms
O7	**617.20**	5.4m	**1216.20**	28m	**2348.71**	38m	**4594.83**	1.4h	**6749.28**	2.1h	**9031.18**	3h
O8	**869.63**	24m	**1811.23**	2.4h	**3648.53**	2.9h	**7592.65**	8.1h	**11239.17**	13h	**14779.40**	16h
O9	**804.36**	0.4h	**1654.35**	1.8h	**3312.49**	3.4	**6833.09**	7.9h	**10109.55**	12h	**13730.80**	15h
F10	**24019.78**	0.4h	**47230.44**	2.2h	**95259.88**	6.0h	**197850.46**	21h	**294935.67**	44h	**390545.18**	96h

Note: Boldface font indicates the optimal value.

Table 3. Effect of the fast procedure (in Fig.2) for calculating F_{tk}

Problem	T=3	T=6	T=12	T=24	T=36	T=48	AVG time
M4	6/6	19/21	51/78	139/300	251/666	277/1176	0.2s
M5	6/6	21/21	44/78	92/300	139/666	199/1176	0.9s
O7	6/6	21/21	64/78	161/300	244/666	352/1176	30s
O8	6/6	18/21	68/78	185/300	289/666	449/1176	128s
O9	6/6	21/21	71/78	151/300	244/666	337/1176	160s
F10	6/6	21/21	47/78	108/300	166/666	232/1176	1489s
AVG	100%	96.0%	73.7%	46.4%	33.4%	26.2%	

Next, we solved the general model for Problem *AFLP-TVD* formulated in Eq.(1) and Constraints (1)-(7) with the tested instances and with respect to period numbers T=1, 3, 6, 12, 24 and 48, respectively. A two-hour time limit was set for all computations. The results were compared to the optimal solutions (from Table 1) in Table 4, where columns *T./D.%* indicate either the computational time (in seconds) if the result was optimal or the deviation (%) from the optimal value if the result was not optimal. The notation "--" indicates no feasible solution was found. It could be observed that only a few optimal solutions could be found in the given time limit, e.g., T=1 for all instances, T=3 for M4 and M5, and T=6 for M4. Most of the solutions were non-optimal and showed quite large deviations from the optimal ones especially for large numbers of periods. This comparative experiment hinted that the AFLP-TVD, as a whole general model, is very difficult to be solved with good solutions even for very small-sized problem instances. Therefore, the BDP algorithm with pre-calculated F_{tk} values would be a practically feasible approach for obtaining optimal solutions for small-sized problem instances. For large-sized problem instances, we adopted the improved PEA-LP algorithm as an efficient solution approach for calculating F_{tk} with the excellent results given in Section 5.3.

Table 4. Computational results of solving the general model with CPLEX (Time limit = 2 hours)

	T=1		T=3		T=6		T=12		T=24		T=36		T=48	
Problem	Obj.	T./D.%	Obj.	T./D.%	Obj.	T./D.%	Obj.	T./D.%	Obj.	T./D.%	Obj.	T./D.%	Obj.	T./D.%
M4	**511.38**	<1s	**1580.15**	7s	**2813.97**	7200s	6234.70	1.8%	13109.35	2.2%	20159.39	2.6%	26458.99	2.6%
M5	**1417.61**	1s	**3398.78**	7200s	6632.80	0.8%	13421.57	0.6%	27387.88	1.3%	--	--	--	--
O7	**131.64**	63s	638.83	3.5%	1291.80	6.2%	2541.87	8.2%	5959.88	29.7%	7856.45	16.4%	12751.02	41.2%
O8	**243.06**	382s	935.69	7.6%	1980.64	9.4%	--	--	9578.13	26.2%	16340.28	45.4%	20076.43	35.8%
O9	**236.13**	558s	879.79	9.4%	1949.17	17.8%	--	---	--	---	14429.36	42.7%	21969.73	60.0%
F10	**7649.99**	1638s	32544.08	35.5%	57532.50	21.8%	137531.67	44.4%	--	--	553081.22	87.5%	699335.94	79.1%

Note: Boldface font indicates the optimal value.

5.3 Testing the improved PEA-LP algorithm for the static FLP

As the computational results demonstrated BDP algorithm is quite efficient given that F_{tk} values are available. Therefore, the main computational challenge is to calculate optimal or near optimal F_{tk} values as efficient as possible. In this section, we tested the improved PEA-LP algorithm using the test instances BA12, BA13, and BA14 (Bazaraa, 1975), the instances M11*, M15*, and M25* (Meller 1992; Liu and Meller, 2007), and the instances SC 30 and SC35 (Liu and Meller, 2007). We ran the improved PEA-LP algorithm 10 times for each of the test instances with respect to the three selection policies, namely, RS, FP, and MFP, respectively. The parameter setting weres $P_{\max} = 50$ and $\Delta = 6$. The first sub-problem solved in the PEA-LP algorithm is P_6, indicating that the first six departments were initially selected and optimized. The parameter β is the accuracy setting introduced in Xiao et al. (2017) for controlling the maximum error range in the area constraint linearization. We used $\beta = 0.1\%$ during the optimization process and $\beta = 0.001\%$ in the post-optimization to refine final solutions within a very accurate level. Thus, our final solutions will be very close to the actual optimal solution. The computational results were listed and compared in Table 5, where column *Selection policy* indicates the department selection policy adopted for partial optimizations, column *FS* how many feasible solutions where found in 10 runs, columns *Average* and *Best* indicate respectively the average and best objective values obtained in 10 runs, column *Dev.%* indicates the deviation (%) from the previous best-known solutions in the literature, column *AVG Time (s)* indicates the average CPU time used in 10 runs, column *Best policy* indicates if the selection policy is the best one out of three policies under comparison, and column *Prev. Best Known* gives the previous best solutions provided in Kulturel-Konak and Konak (2013), Kulturel-Konak and Konak (2015), Gonçalves and Resende (2015), and Xiao et al. (2017).

Table 5. Overall performances of the improved PEA-LP algorithm ($\beta = 0.1\%$)

Problem	Section policy	The improved PEA-LP (10 runs) *Fs*	Average	Dev.%	Best	Dev.%	AVG T.(s)	Best policy	Prev. Best Known	Sources
BA12	RS	10	8092.67	0.89%	**8020.98**	0.00%	722	Yes	**8020.97**	Gonçalves and Resende (2015)
	FP	10	8100.48	0.99%	**8020.98**	0.00%	643	Yes		
	MFP	10	8089.74	0.86%	**8020.98**	0.00%	685	Yes		
BA13	RS	10	4654.71	0.57%	**4628.23**	0.00%	723	Yes	**4628.32**	Kulturel-Konak & Konak (2013)
	FP	10	4746.69	2.56%	**4628.23**	0.00%	656	Yes		
	MFP	10	4710.36	1.77%	**4628.23**	0.00%	629	Yes		
BA14	RS	9	4677.92	1.06%	**4628.23**	-0.01%	860	Yes	**4628.79**	Gonçalves & Resende (2015)
	FP	10	4702.93	1.60%	**4628.23**	-0.01%	946	Yes		
	MFP	10	4752.36	2.67%	**4628.23**	-0.01%	830	Yes		
M11*	RS	8	1177.73	2.61%	***1136.23***	-1.00%	920	Yes	**1147.75**	Xiao et al. (2017)

	FP	10	1191.46	3.81%	***1136.23***	-1.00%	831	Yes		
	MFP	10	1182.56	3.03%	***1136.23***	-1.00%	750	Yes		
M15*	RS	9	25281.50	1.88%	***23118.53***	-6.84%	629	Yes	**24815.22**	Xiao et al. (2017)
	FP	10	25888.73	4.33%	*24411.54*	-1.63%	745			
	MFP	10	25784.15	3.90%	***23118.53***	-6.84%	632	Yes		
M25*	RS	8	1367.26	14.91%	1268.04	6.57%	861		**1189.88**	Kulturel-Konak & Konak (2013)
	FP	8	1328.98	11.69%	***1138.35***	-4.33%	860	Yes		
	MFP	10	1381.04	16.07%	1218.03	2.37%	982			
SC30	RS	10	3300.44	-0.55%	***3240.06***	-2.37%	3634	Yes	**3318.76**	Kulturel-Konak & Konak (2015)
	FP	10	3333.77	0.45%	*3299.10*	-0.59%	3231			
	MFP	10	3326.37	0.23%	*3284.61*	-1.03%	3126			
SC35	RS	10	3390.20	2.21%	*3310.92*	-0.18%	992		**3316.77**	Gonçalves & Resende (2015)
	FP	10	3366.37	1.50%	***3304.77***	-0.36%	1076	Yes		
	MFP	10	3410.69	2.83%	*3312.04*	-0.14%	813			

Note: The Bold & Italic font indicates the new best-known value.

As seen in Table 5, the improved PEA-LP algorithm was able to find the same best solutions reported in Kulturel-Konak and Konak (2013) and Gonçalves and Resende (2015) for problems BA12, BA13, and BA14. The slight differences in the solutions are due to numeric errors incurred by the area linearization as their departmental layouts are exactly the same. New best-known solutions were found for all the rest problems including M11*, M15*, M25*, SC30, and SC35, with improvements of -1.00%, -6.84%, -4.33%, -2.37%, and -0.36%, respectively, and in comparison to the previous best known solutions provided in Xiao et al. (2017), Kulturel-Konak and Konak (2013, 2015), and Gonçalves and Resende (2015). This experimental comparison indicated that the improved PEA-LP algorithm can solve the static FLP efficiently with good solution qualities.

In Tables 6 and 7, the performance comparisons of three selection policies (i.e., RS, FP, and MFP) are provided in terms of the average solutions (in Table 6) and the best solutions (in Table 7) found in the experiments. The numbers 1, 2, and 3 in the tables indicate the ranking positions of the policies, e.g., number 1 is the best, number 3 represents the worst, and number 2 represents the middle. It can be observed that in terms of the ranking position of the average solutions found, the RS policy performed the best, followed by the FP policy and the MFP policy. In terms of the ranking position of the best solution found, the three policies showed an equal performance (all have 11 summarized points). The RS policy performed better than other two policies on problems M15* and SC30, but it was worse than the FP policy on problems M25* and SC35, and is worse than the MFP policy on problem M25*. This means that all three tested policies showed strength on a part of the test problems but may be weak on other problems. Therefore, hybrid use of them may be a promising approach for various types of problems, and interested readers are suggested to refer to the competition strategy for hybrid use of multiple operators (Dellaert and Jeunet, 2000; Xiao et al., 2016, 2019).

Table 6. Ranking positions of three selection policies with respect to the average solution

Ranking Position	BA12	BA13	BA14	M11*	M15*	M25*	SC30	SC35	Sum.

RS	2	1	1	1	1	2	1	2	11
FP	3	3	2	3	3	1	3	1	19
MFP	1	2	3	2	2	3	2	3	18

Table7. Ranking positions of three selection policies with respect to the best solution

Ranking Position	BA12	BA13	BA14	M11*	M15*	M25*	SC30	SC35	Sum.
RS	1	1	1	1	1	3	1	2	11
FP	1	1	1	1	2	1	3	1	11
MFP	1	1	1	1	1	2	2	3	11

6. Conclusion

The aperiodic facility layout problem (AFLP) studied in this paper is strongly NP-hard, as it combines concurrently two optimizations: (1) the combinatorial optimization for WHEN to carry out a re-layout (aperiodically or periodically) in pre-specified multiple periods, and (2) the static optimization for HOW to design optimally the facility layout in selected periods. Existing exact algorithms, e.g., the branch & bound algorithm in CPLEX, can only solve very small-sized problems, e.g., less than 5 departments over 6 periods, in acceptable CPU time. The decomposition of the AFLP into a master problem and a slave problem is a promising way that has reduced the problem complexity significantly. The proposed BDP algorithm in this paper is an exact algorithm that can solve the master problem with a complexity level of O (n^2). The slave problem is a static FLP that has been extensively studied in the literature, and there are many algorithms available to solve it. The PEA-LP algorithm is a proven solution approach that is among the best ones for solving FLPs. The improved PEA-LP developed in this paper showed even better performances on solving all sizes of FLP problem instances, with five the known benchmark problems being updated with new best-known solutions in the experiments. Future studies may be conducted on two aspects: (1) to develop a more efficient approach for calculating F_{tk} that is a time-consuming part of the proposed solution approach, and (2) to extend the AFLP model by considering more practical factors that cause the re-layout cost such as in zone-based or multi-floor situations.

Acknowledgments

This study was partly supported by the National Natural Science Foundation of China under Grant Nos. 71871003 and 71571004.

Reference

[1]. Ahmadi A., Pishvaee S. M., Jokar A. R. M., 2017. A survey on multi-floor facility layout problems. Computers & Industrial Engineering 107, 158-170.

[2]. Amaral A. R. S., 2006. On the exact solution of a facility layout problem. European Journal of Operational Research, 173(2), 508-518.

[3]. Arapoglu A. R., Norman A. B., Smith E. A., 2001. Locating Input and Output Points in Facilities Design—A Comparison of Constructive,Evolutionary, and Exact Methods. IEEE Transactions on Evolutionary Computation 5(3), 192-203.

[4]. Balakrishnan J., Cheng C. H., 2009. The dynamic plant layout problem: incorporating rolling horizons and forecast uncertainty. Omega 37, 165–177 .

[5]. Balakrishnan J., Cheng C. H., Conway D. G., Lau C. M., 2003. A hybrid genetic algo- rithm for the dynamic plant layout problem. Int. J. Prod. Econ. 86 (2), 107–120.
[6]. Baykasoglu A., Dereli T., Sabuncu I., 2006. An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. Omega 34, 385–396.
[7]. Bazaraa M. S., Elshafei A. N., 1979. An exact branch and bound procedure for quadratic assignment problems. Naval Res. Logistics Q. 26, 109–121.
[8]. Bazaraa M.S., 1975. Computerized layout design: a branch and bound approach. AIIE Trans. 7, 432–438.
[9]. Beghin-Picavet M., Hansen P., 1982. Deux problemes d'affectation non linéaires. RAIRO, Recherche Opérationelle 16, 263-276.
[10]. Bozer Y. A., Meller R. D., 1997. A re-examination of the distance based facility layout problem. IIE Trans 29 (7), 549–560.
[11]. Castillo I., Westerlund T., 2005. An ε-accurate model for optimal unequal area facility layout. Computers and Operations Research, 32, 429–447.
[12]. Chen G. Y. H., 2013. A new data structure of solution representation in hybrid ant colony optimization for large dynamic facility layout problems. Int. J. Prod. Econ. 142, 362–371.
[13]. Cravo G. L., Amaral A. R. S., 2019. A GRASP algorithm for solving large-scale single row facility layout problems. Computers & Operations Research 106, 49-61
[14]. Datta D., Amaral A. R. S., Figueira J. R., 2011. Single row facility layout problem using a permutation-based genetic algorithm. European Journal of Operational Research 213(2), 388-394.
[15]. Dellaert N., Jeunet J., 2000. Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm. Int. J. Prod. Res. 38, 1083-1099.
[16]. Drira A., Pierreval H., Hajri-Gabouj S., 2007. Facility layout problems: a survey. Annu. Rev. Control 31, 255–267.
[17]. Dunker T., Radons G., Westkamper E., 2005. Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. Eur. J. Oper. Res. 165 (1), 55–69.
[18]. Friedrich C., Klausnitzer A., Lasch R., 2018. Integrated slicing tree approach for solving the facility layout problem with input and output locations based on contour distance. European Journal of Operational Research 270(3), 837-851.
[19]. Gonçalves J. F., Resende M. G. C., 2015. A biased random-key genetic algorithm for the unequal area facility layout problem. Eur. J. Oper. Res. 246 (1), 86–107.
[20]. Guan J., Lin G., 2016. Hybridizing variable neighborhood search with ant colony optimization for solving the single row facility layout problem. European Journal of Operational Research, 248 (3), 899-909.
[21]. Hale T. S., Huq F., Hipkin I., 2012. An improved facility layout construction method. International Journal of Production Research, 50 (15), 4271–4278.
[22]. Heragu S. S., Kusiak A., 1988. Machine layout problem in flexible manufacturing systems. Operations Research 36(2), 258-268.
[23]. Hitchings G.G., 1970. Control, redundancy and change in layout systems. AIIE Trans. 2, 253–260.
[24]. Hosseini-Nasab H., Emami L., 2013. A hybrid particle swarm optimization for dynamic facility layout problem. Int. J. Prod. Res. 51 (4), 4325–4335 .
[25]. Hungerländer P., Rendl F., 2012. A computational study and survey of methods for the single-row facility layout problem. Computational Optimization and Applications 55(1), 1-20
[26]. Komarudin, Wong K. Y., 2010. Applying ant system for solving unequal area facility layout problems. Eur. J. Oper. Res. 202, 730–746.
[27]. Konak A., Kulturel-Konak S., Norman B.-A., Smith A.-E., 2006. A new mixed integer programming formulation for facility layout design using flexible bays. Operations Research Letters, 34 , 660–672 .
[28]. Koopmans T.C., Beckmann M., 1957. Assignment problems and the location of economic activities. Econometrica 25, 53–76.
[29]. Kulturel-Konak, S., 2019. The Zone-based Dynamic Facility Layout Problem. INFOR: Information Systems and Operational Research, 57, 1-31.

[30]. Kulturel-Konak S., Konak A., 2015. A large scale hybrid simulated annealing algorithm for cyclic facility layout problem. Eng. Optim. 47 (7), 963–973.

[31]. Kulturel-Konak S., Konak A., 2013. Linear programming based genetic algorithm for the unequal area facility layout problem. Int. J. Prod. Res. 51 (14), 4302–4324.

[32]. Kulturel-Konak S., 2012. A linear programming embedded probabilistic tabu search for the unequal-area facility layout problem with flexible bays. European Journal of Operational Research, 223 (3), 614–625.

[33]. Lacksonen T. A., 1997. Preprocessing for static and dynamic facility layout problems. Int. J. Prod. Res. 35 (4), 1095–1106.

[34]. Lacksonen T. A., 1994. Static and dynamic layout with varying areas. J. Oper. Res. Soc. 45, 59–69.

[35]. Lacksonen T. A., Emory Enscore Jr, E., 1993. Quadratic assignment algorithms for the dynamic layout problem. Int. J. Prod. Res. 31 (3), 503–517.

[36]. Lawler E.L., 1963. The quadratic assignment problem. Manage. Sci. 9 (4), 586–599.

[37]. Lee K.-Y., Roh M., Jeong H.-S., 2005. An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. Comput. Oper. Res. 32 (4), 879–899.

[38]. Lee Y. H., Lee M. H., 2002. A shape-based block layout approach to facility layout problems using hybrid genetic algorithm. Comput. Ind. Eng. 42, 237–248.

[39]. Liu J., Wang D., He K., Xue Y., 2017. Combining Wang–Landau sampling algorithm and heuristics for solving the unequal-area dynamic facility layout problem. European Journal of Operational Research 262(3), 1052-1063.

[40]. Liu Jiafa, Liu Jun, 2019. Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems. Applied Soft Computing 74, 167-189.

[41]. Liu Q., Meller R. D., 2007. A sequence-pair representation and MIP-model-based heuristic for the facility layout problem with rectangular departments. IIE Trans. 39, 377–394.

[42]. McKendall A. R., Liu, W-H., 2012. New Tabu search heuristics for the dynamic facility layout problem. Int. J. Prod. Res. 50 (3), 867–878.

[43]. McKendall A. R., Hakobyan A., 2010. Heuristics for the dynamic facility layout problem with unequal-area departments. Eur. J. Oper. Res. 201 (1), 171–182.

[44]. McKendall R. A., Shang J., Kuppusamy S., 2006. Simulated annealing heuristics for the dynamic facility layout problem. Computers & Operations Research 33(8), 2431-2444.

[45]. Meller R. D., Narayanan V., Vance P. H., 1999. Optimal facility layout design. OR Letters, 23, 117–127.

[46]. Meller R. D., 1997. The multi-bay manufacturing facility layout problem. Int. J. Prod. Res. 35 (5), 1229–1237.

[47]. Meller R. D., Bozer Y. A., 1997. Alternative approaches to solve the multi-floor facility layout problem. J. Manufact. Syst. 16 (3), 192–203.

[48]. Meller R. D., 1992. Layout Algorithms for Single and Multiple Floor Facilities. University of Michigan.

[49]. Montreuil B., 1990. A modeling framework for integrating layout design and flow network design. In: Proceedings of Materials Handling Research Colloq.. Hebron, KY, pp. 43–58.

[50]. Montreuil B., Ouazzani N., Brotherton E., Nourelfath M., 2004. Antzone layout metaheuristic: coupling zone-based layout optimization, ant colony system and domain knowledge. In: Progress in Material Handling Research. Material Handling Institute of America, Charlotte, pp. 301–331.

[51]. Montreuil B., Brotherton E., Marcotte S., 2002. Zone based layout optimization. In: Proceedings of 2002 Industrial Engineering Research Conference. Orlando, USA, pp. 19–22 .

[52]. Montreuil B., Venkatadri U., Ratliff H. D.,1993. Generating a layout design from a design skeleton. IIE Trans. 25 (1), 3–15.

[53]. Montreuil B., Venkatadri U., 1991. Strategic interpolative design of dynamic manufacturing systems layouts. Manage. Sci. 37 (6), 682–694.

[54]. Montreuil B., Ratliff H. D., 1989. Utilizing cut trees as design skeletons for facility layout. IIE Trans. 21 (2), 136–143.

[55]. Montreuil B., Ratliff D. H., 1988. Optimizing the location of input/output stations within facilities layout. Engineering Costs and Production Economics 14(3), 177-187.

[56]. Niroomand S., Vizvári B., 2013. A mixed integer linear programming formulation of closed loop layout with exact distances. Journal of Industrial and Production Engineering 30(3), 190–201.

[57]. Ozdemir G., Smith A. E., Norman, B. A., 2003. Incorporating heterogeneous distance metrics within block layout design. International Journal of Production Research 41(5), 1045–1056.

[58]. Paes F. G., Pessoa A. A., Vidal T., 2017. A hybrid genetic algorithm with decomposition phases for the Unequal Area Facility Layout Problem. European Journal of Operational Research 256, 742–756.

[59]. Picard J.-C., Queyranne M., 1981. On the one-dimensional space allocation problem. Operations Research 29 (2), 371-391.

[60]. Rubio-Sánchez M., Gallego M., Gortázar F., Duarte A., 2016. GRASP with path relinking for the single row facility layout problem. Knowledge-Based Systems 106(15), 1-13.

[61]. Sahni S., Gonzalez T., 1976. P-complete approximation problems. J. Assoc. Comput. Mach. 23, 555–565.

[62]. Saraswat A., Venkatadri U., Castillo I., 2015. A framework for multi-objective facil- ity layout design. Comput. Ind. Eng. 90, 167–176.

[63]. Sherali H. D., Fraticelli B. M. P., Meller R. D., 2003. Enhanced model formulations for optimal facility layout. Operational Research, 51, 629–644.

[64]. Simmons D. M., 1969. One-dimensional space allocation: an ordering algorithm. Operations Research, 17(5), 812-826.

[65]. Tam K. Y., 1992. A simulated annealing algorithm for allocating space to manufac- turing cells. Int. J. Prod. Res. 30, 63–87.

[66]. Turanoğlu B., Akkaya G., 2018. A new hybrid heuristic algorithm based on bacterial foraging optimization for the dynamic facility layout problem. Expert Systems with Applications 98, 93-104.

[67]. Ulutas B. H., Kulturel-Konak S., 2012. An artificial immune system based algorithm to solve unequal area facility layout problem. Expert Syst. Appl. 39, 5384–5395 .

[68]. Ulutas B., Islier A., 2015. Dynamic facility layout problem in footwear industry. Journal of Manufacturing Systems 35, 55–61.

[69]. Urban T. L., 1993. A heuristic for the dynamic facility layout problem. IIE Trans. 25 (4), 57–63.

[70]. Wong K. Y., Komarudin, 2010. Solving facility layout problems using flexible bay structure representation and ant system algorithm. Expert Syst. Appl. 37, 5523–5527.

[71]. Xiao X., Hu Y., Wang W., Ren W., 2019. A robust optimization approach for unequal-area dynamic facility layout with demand uncertainty. Procedia CIRP 81, 594-599.

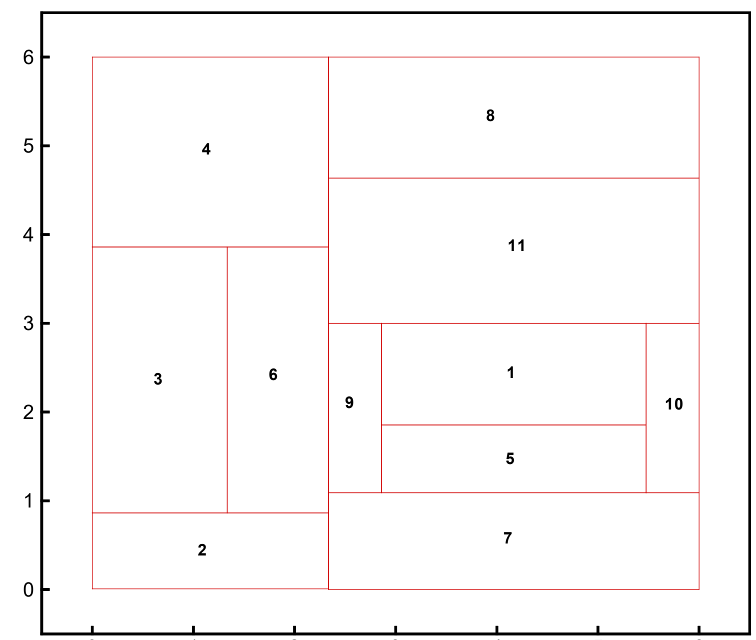
[72]. Xiao Y., Konak A., 2016. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. Transportation Research Part E 88, 146–166.

[73]. Xiao Y., Xie Y., Kulturel-Konak S., Konak A., 2017. A problem evolution algorithm with linear programming for the dynamic facility layout problem—A general layout formulation. Computers and Operations Research 88, 187–207.

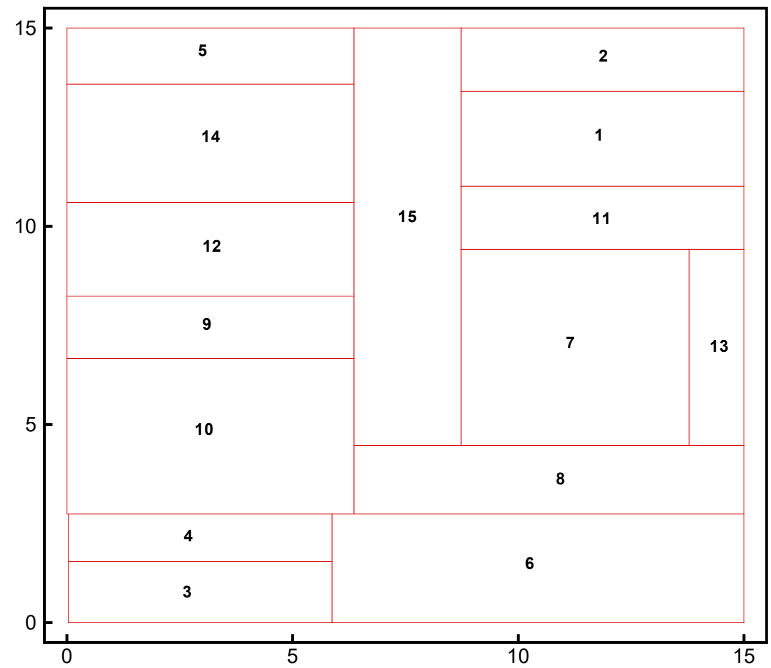
[74]. Xiao Y., Zuo X., Kaku I., Zhou S., Pan X., 2019. Development of energy consumption optimization model for the electric vehicle routing problem with time windows. Journal of Cleaner Production 225, 647-663.

[75]. Xie Y., Zhou S., Xiao Y., Kulturel-Konak S., Konak A., 2018. A β-accurate linearization method of Euclidean distance for the facility layout problem with heterogeneous distance metrics. European Journal of Operational Research 265, 26–38.

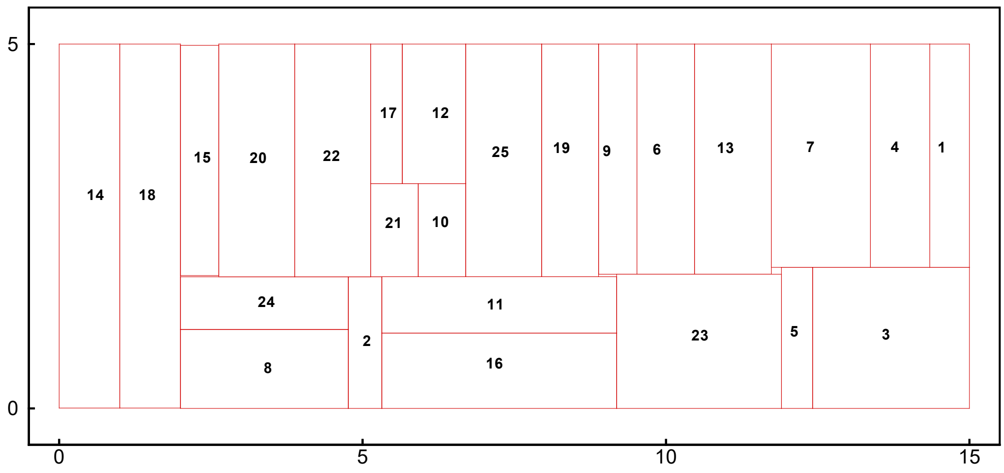
Appendix: New best solutions



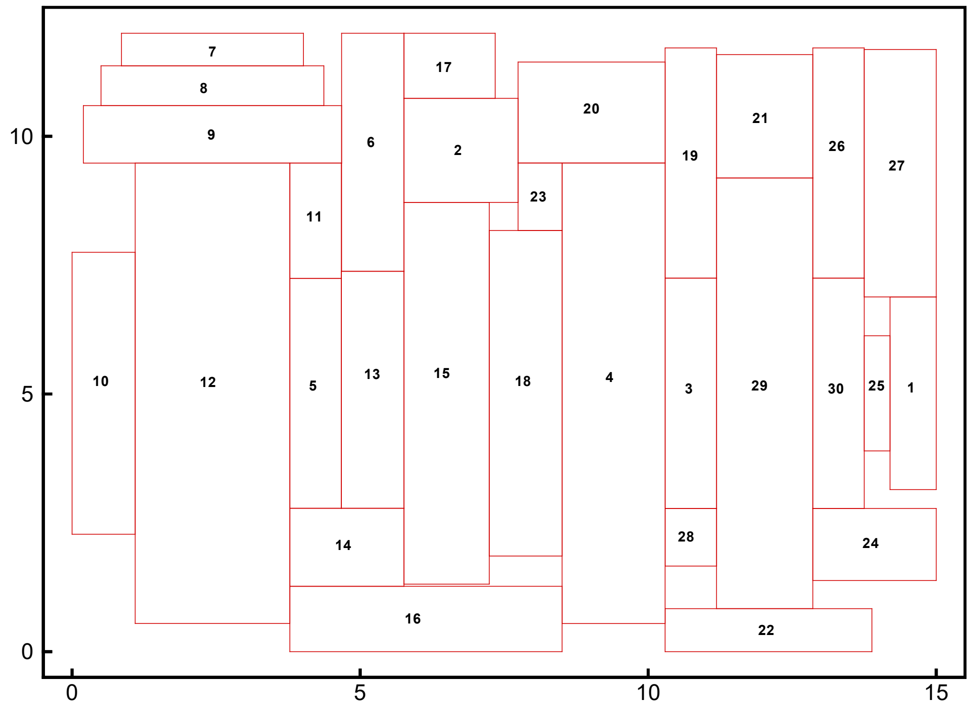
M11* (1136.23)



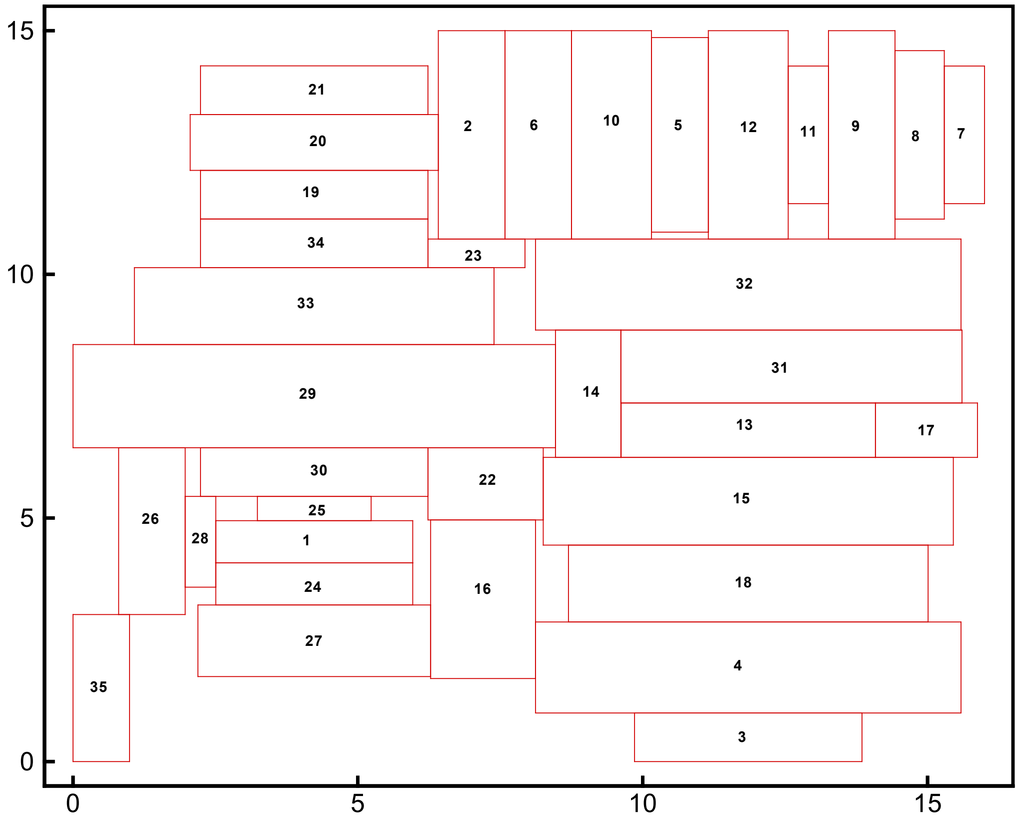
M15*(23118.53)



M25* (1138.35)



SC30 (3240.06)


15
10
5
0
0
5
10
15
21
20
19
34
33
29
30
25
1
24
27
26
28
35
2
6
10
5
12
11
9
8
7
23
32
14
31
13
17
22
15
16
18
4
3

SC35* (3304.77)