European Journal of Operational Research 256 (2017) 742–756

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

A hybrid genetic algorithm with decomposition phases for the Unequal Area Facility Layout Problem

Frederico Galaxe Paes[a,*], Artur Alves Pessoa[b], Thibaut Vidal[c]

[a] Instituto Federal Fluminense – IFF, Av. Souza Mota, 350, Pq. Fundão, 28060-010 Campos dos Goytacazes, RJ, Brazil
[b] Departamento de Engenharia de Produção – Universidade Federal Fluminense, Rua Passo da Pátria 156, São Domingos 24210-240 Niterói, RJ, Brazil
[c] Departamento de Informática – PUC–Rio, Rua Marquês de São Vicente, 225, Gávea 22451-900 Rio de Janeiro, RJ, Brazil

ARTICLE INFO

Article history:
Received 22 October 2015
Accepted 11 July 2016
Available online 20 July 2016

Keywords:
Facility layout
Genetic algorithm
Hybrid metaheuristics
Decomposition strategies

ABSTRACT

We address the Unequal-Area Facility-Layout Problem (UA-FLP), which aims to dimension and locate rectangular facilities in an unlimited floor space, without overlap, while minimizing the sum of distances among facilities weighted by "material-handling" flows. We introduce two algorithmic approaches to address this problem: a basic Genetic Algorithm (GA), and a GA combined with a decomposition strategy via partial solution deconstructions and reconstructions. To efficiently decompose the problem, we impose a solution structure where no facility should cross the X or Y axis. Although this restriction can possibly deteriorate the value of the best achievable solution, it also greatly enhances the search capabilities of the method on medium and large problem instances. For most such instances, current exact methods are impracticable. As highlighted by our experiments, the resulting algorithm produces solutions of high quality for the two classic datasets of the literature, improving six out of the eight best known solutions from the first set, with up to 125 facilities, and all medium- and large-scale instances from the second set. For some of the largest instances of the second set, with 90 or 100 facilities, the average solution improvement goes as high as 6 percent or 7 percent when compared to previous algorithms, in less CPU time. We finally introduce additional instances with up to 150 facilities. On this benchmark, the decomposition method provides an average solution improvement with respect to the basic GA of about 9 percent and 1.3 percent on short and long runs, respectively.

1. Introduction

The Facility Layout Problem (FLP) is a well-researched non-linear combinatorial optimization problem, which aims to determine the locations of rectangular facilities in the floor space without overlap between them, while minimizing an objective, such as material-handling costs. The FLP is critical for production efficiency because it directly affects manufacturing costs, work in process, lead times and productivity (Drira, Pierreval, & Hajri-Gabouj, 2007). A good placement of facilities contributes to the operational efficiency and can save up to 50 percent of expenses (Tompkins, White, Bozer, & Tanchoco, 2010). FLPs can be classified as static or dynamic, with equal or unequal facility areas as well as soft or hard dimensions. The placement space can also be open or enclosed. Dedicated surveys of FLP algorithms can be found in Drira et al. (2007) and Singh and Sharma (2006).

In this paper, we focus on the static variant of the FLP with both open and continuous placement space. We also consider both hard and soft facilities with unequal areas. This class of problems is denoted as UA-FLP. Given the material-handling flow between facilities, the objective is to minimize the sum – over all facility pairs – of the distances between the centroids of the facilities weighted by the flows. The constraints of the problem include facility area requirements, shape restrictions, and non-overlap. The UA-FLP is NP-hard as a generalization of the optimal linear arrangement (Garey, Johnson, & Stockmeyer, 1976). The exact resolution of medium-scale instances of UA-FLP, e.g., with more than 30 facilities, is currently impracticable due to excessive CPU time. Therefore, analytic techniques, heuristics and meta-heuristics have been frequently used to solve this problem.

Among the existing analytical techniques for this problem, we highlight the cluster boundary search algorithm (CBA) presented in Imam and Mir (1998). This construction and improvement algorithm locates, in turn, each facility via piecewise one-dimensional searches on the boundary formed by existing facilities. Kado (1996) developed six variants of genetic algorithms (GA) using

* Corresponding author.
E-mail addresses: fpaes@iff.edu.br (F.G. Paes), artur@producao.uff.br (A.A. Pessoa), vidalt@inf.puc-rio.br (T. Vidal).

http://dx.doi.org/10.1016/j.ejor.2016.07.022
0377-2217/

a slicing-tree structure for solution representation. In Dunker, Radons, and Westkämper (2003), a co-evolutionary approach is proposed, with improved mutation and crossover operators. The facilities are also clustered into groups to deal with larger problems. A tabu search algorithm with a slicing-tree representation and a bounding curve is proposed in Scholz, Petrick, and Domschke (2009) for the UA-FLP with fixed and flexible facilities. Komarudin (2010) proposed an Ant System to solve the UA-FLP, with a slicing-tree representation and several types of local search to improve the search performance. McKendall and Hakobyan (2010) extended CBA to solve dynamic facility layout problems (DFLP), leading to a construction heuristic called Boundary Search (BSH), which inserts facilities along the boundaries of existing facilities. After building a solution with BSH, a tabu search heuristic is used to improve the solution. A hybrid particle swarm optimization and local search approach was proposed in Kulturel-Konak and Konak (2011) for the DFLP, using a relaxed flexible bay structure (RFBS).

Finally, Gonçalves and Resende (2015) recently proposed a hybrid approach to solve both unconstrained and constrained cases of UA-FLP. This approach combines a biased random-key genetic algorithm (BRKGA) to determine the order of insertion of facilities and their dimensions, a greedy strategy to position each facility, and a linear programming model to fine-tune selected solutions (only in the constrained case). Out of 28 classical benchmark instances for the unconstrained problem, the approach improves the best known solutions (BKS) in 19 cases, with an amount of improvement ranging from 1.86 percent to 11.05 percent.

The most surprising aspect of the method proposed by Gonçalves and Resende (2015) is that it does not perform any kind of local search for the unconstrained case (addressed in this paper). This may indicate that the solution improvements of classical local searches are, for this problem, not worth the additional computational effort required by them, due to the special structure of the solutions. Indeed, the facilities should be placed in such a way that they match almost perfectly the space reserved for them. It is thus difficult to improve a small subset of such solution without creating overlap or deteriorating the cost. For this reason, simply reconstructing the entire solution – or large parts of it – using guided construction heuristics can be more successful.

We thus took inspiration from guided constructive metaheuristics such as GRASP, Ant Colony Optimization and BRKGA, as well as large neighborhood search and decompositions to propose an efficient algorithm. More specifically, we rely on the methodology of Taillard and Voss (2002), which iteratively fixes parts of a solution to efficiently improve a subset of geographically-related facilities. One main challenge, when applying this approach to the UA-FLP, is that the irregular frontier of any subset of facilities selected as a subproblem prevents most – if not all – solutions other than the current one to be feasible or profitable. To overcome this issue, we propose to force a specific solution structure, in such a way that four subproblems with a regular frontier can be defined. We show in our experiments that this imposed structure does not significantly deteriorate the solution quality, and that the re-optimization of the underlying subproblems provides worthwhile improvements on large instances.

The main contributions of this article are the following.

1. We introduce two algorithmic approaches for the UA-FLP: a basic GA, and a GA combined with a strategy of partial solution deconstruction and reconstruction (PDR).
2. These algorithms combine the construction heuristic of Gonçalves and Resende (2015) with new solution representation, crossover, population management and intensification strategies.
3. We propose to impose a specific solution structure, where no facilities cross the X or Y axis. This solution structure opens the way to successful decomposition techniques.
4. Our computational experiments on the two classic sets of benchmark instances for this problem demonstrate the good performance of the new algorithms and the contribution of the decomposition phases, leading to new best known solutions for most large-scale instances. For the large instances of the second benchmark (Gonçalves & Resende, 2015), with 90 or 100 facilities, the average solution improvement with respect to previous algorithms goes as high as 6 percent to 7 percent, in less CPU time. In contrast, for some smaller instances of the second set, the decomposition-based approach is outperformed by the basic GA. As verified in our experiments, this is explained by a special “line” structure of the flow matrix and optimal solutions for this benchmark. To complement these observations, we finally propose 25 new instances with 50–150 facilities, and different percentages of soft facilities. On this last benchmark, the decomposition approach leads again to significant improvements over the basic GA, leading to cost reductions of 9 percent and 1.3 percent in average for short and long runs, respectively.

The remainder of the paper is organized as follows. Section 2 formally defines the problem. Section 3 presents the proposed GA for the UA-FLP, including the greedy construction method used to determine the facility positions and dimensions, the population management, crossover and other genetic operators, as well as the deconstruction and reconstruction strategy. Section 4 describes our computational experiments, and Section 5 concludes.

2. Problem statement

The UA-FLP is the problem of placing, without overlap, the centroids $C_i = (x_i, y_i),\ i = 1, \dots, n,$ of n rectangular facilities with unequal area on an unconstrained rectangular floor space. Each facility i is defined by its width w_i and its height h_i along the X and Y axis, respectively, having its area $A_i = w_i \times h_i$, and its aspect ratio $AR_i = w_i/h_i$. Due to practical reasons, it is assumed that $AR_i^{min} \le AR_i \le AR_i^{max}$. As such, a layout is determined by the coordinates of the centroids and the dimension of each facility i, while the area and the minimum and maximum aspect ratios are input data for the problem. A facility i is considered as *soft* when $AR_i^{min} < AR_i^{max}$. Otherwise, when $AR_i^{min} = AR_i^{max}$ the facility i is said to be *hard*. The cost function to be minimized is:

$$z(C_1, \dots, C_n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{ij} d(C_i, C_j). \tag{1}$$

In Eq. (1), f_{ij} represents the flow of material between the facilities i e j and $d(C_i, C_j)$ is the measure of distance between their centroids. In accordance with the used metric, this distance can be measured using one of the following equations:

1. Rectilinear Distance (R):
$$d(C_i, C_j) = |x_i - x_j| + |y_i - y_j|, \tag{2}$$

2. Squared Euclidean Distance (SE):
$$d(C_i, C_j) = (x_i - x_j)^2 + (y_i - y_j)^2, \tag{3}$$

3. Euclidean Distance (E):
$$d(C_i, C_j) = [(x_i - x_j)^2 + (y_i - y_j)^2]^{\frac{1}{2}}. \tag{4}$$

We assume, w.l.o.g., that the flow matrix is symmetric ($f_{ij} = f_{ji}$) since f_{ij} can always be replaced by $f_{ij} + f_{ji}$, for all $(i, j) \in \{1, \dots, n\}^2$, due to the symmetry of the distance functions. The

Table 1
Classical benchmark datasets and their distance metrics.

Datasets	Metric	Aspect ratio	Rotation	Description	Source
L020, L100	R	No	No	Instances with 20, 28, 50, 100 and	Imam and Mir (1993)
L125B				125 facilities, respectively	Mir and Imam (1996)
L028	SE	No	No		Imam and Mir (1998)
L050	E	No	No		Mir and Imam (2001) and
L075	R	Yes	No	Instances with 75 and 125 facilities,	VIP-PLANOPT (2006)
L125A				respectively	http://www.planopt.com
L062	R	No	Yes	Instance with 62 facilities	Dunker et al. (2003)
RND10-100	R	No	No	100 random generated datasets. Ten datasets for facility sizes equal to 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. The optimal solution for each dataset is known	Gonçalves and Resende (2015)
F050-150	R	Yes	No	25 random generated datasets composed by five groups with facility sizes equal to 50, 75, 100, 125 and 150 Each group in turn, is composed by datasets with 0, 10, 25, 50 and 100 percent of soft facilities	Available from the authors upon request

overlap area S_{ij} between two facilities i and j is calculated as follows (Mir & Imam, 2001):

$$S_{ij} = \lambda_{ij}(\Delta X_{ij})(\Delta Y_{ij}), \tag{5}$$

where,

$$\Delta X_{ij} = \left(\frac{w_i + w_j}{2}\right) - |x_i - x_j| \tag{6}$$

$$\Delta Y_{ij} = \left(\frac{h_i + h_j}{2}\right) - |y_i - y_j| \tag{7}$$

$$\lambda_{ij} = \begin{cases} -1 & \text{for} \quad \Delta X_{ij} \le 0 \quad \text{and} \quad \Delta Y_{ij} \le 0 \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

The value of S_{ij} is positive if and only if there is an overlap between facilities i and j. A complete mathematical model for the UA-FLP can be expressed as follows:

$$\text{Minimize} \quad z(C_1, \ldots, C_n) \tag{9}$$

$$\text{s.t.} \quad S_{ij} \le 0 \quad i = 1, \ldots, n-1; \quad j = i+1, \ldots, n; \tag{10}$$

$$w_i \times h_i = A_i \quad i = 1, \ldots, n. \tag{11}$$

$$AR_{min} \le \frac{w_i}{h_i} \le AR_{max} \quad i = 1, \ldots, n; \tag{12}$$

If rotation is allowed then (12) is replaced by

$$AR_{min} \le \frac{w_i}{h_i} \le AR_{max} \quad \text{or} \quad AR_{min} \le \frac{h_i}{w_i} \le AR_{max} \quad i = 1, \ldots, n. \tag{13}$$

The proposed algorithms can solve problem instances with any metric in (2)–(4). Table 1 displays a summary of the existing benchmark instances and their metrics.

3. Proposed methodology

In this section, we introduce a basic genetic algorithm (BGA) for the UA-FLP as well as a variant that prohibits the facilities to cross the X and Y axis (QGA), and its combination with deconstruction and reconstruction of solutions, called DRQGA. These algorithms use some successful strategies of Gonçalves and Resende (2015), as well as new elements described in the following.

3.1. Chromosome representation and facility placement strategy

In both the BGA and the QGA, a chromosome is represented as a permutation of the facilities $\{1, \ldots, n\}$, which represent the order of construction for the solution. Similarly to Gonçalves and Resende (2015) a constructive procedure acts as *solution decoder* to produce the final layout for each permutation. There are three key differences between our constructive approach and that of Gonçalves and Resende (2015).

First, our solution representation does not specify the information about facility dimensions, as these decision variables will be determined during the greedy construction procedure.

The second difference concerns only the QGA variant. In this algorithm, the facilities are inserted in one of the four quadrants of the Cartesian plane that represents the floor space, as depicted in Fig. 1(b). Hence, they are not allowed to span two quadrants. The acronym QGA derives from this additional *quadrant* constraint, which forces some regular solution frontiers, enhances convergence and opens the way for efficient deconstruction and reconstruction phases.

Finally, we propose an alternative criterion for the construction algorithm. The cost function of previous works (to be minimized when determining the position of a given facility i) is given by Eq. (14), where M is the set of facilities already placed. Our alternative criterion uses a cost function that combines Eq. (14) with the distance of the facility i to the origin of the floor space, as shown in Eq. (15).

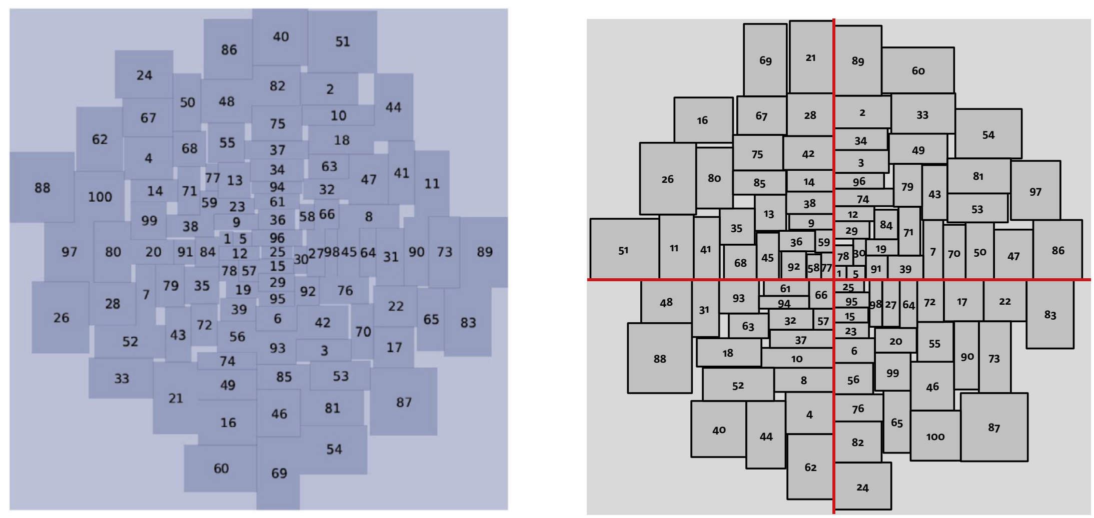
$$z_i = \sum_{j \in M} f_{ij} d(C_i, C_j) \tag{14}$$

$$z_i^D = \left(\frac{n - |M| - 1}{n - 1}\right) d(C_i, O) \sum_{j \in M} f_{ij} + \left(1 - \left(\frac{n - |M| - 1}{n - 1}\right)\right) \sum_{j \in M} f_{ij} d(C_i, C_j), \tag{15}$$

Note that z_i^D is equivalent to the distance to the origin in the beginning of the construction, and becomes nearly equivalent to z_i as more facilities are being placed. The modified criterion contributes to better pack together the first few facilities during construction, even if they do not have mutual flows.

3.1.1. Efficient construction

The greedy construction procedures of both BGA and QGA use the same data structures as Gonçalves and Resende (2015). To



(a) BRKGA: Cost=478910.09; Gonçalves and Resende (2015).

(b) Proposed method: Cost=473097.90.

Fig. 1. Solutions for the instance L100, without (a) or with (b) quadrant constraints.

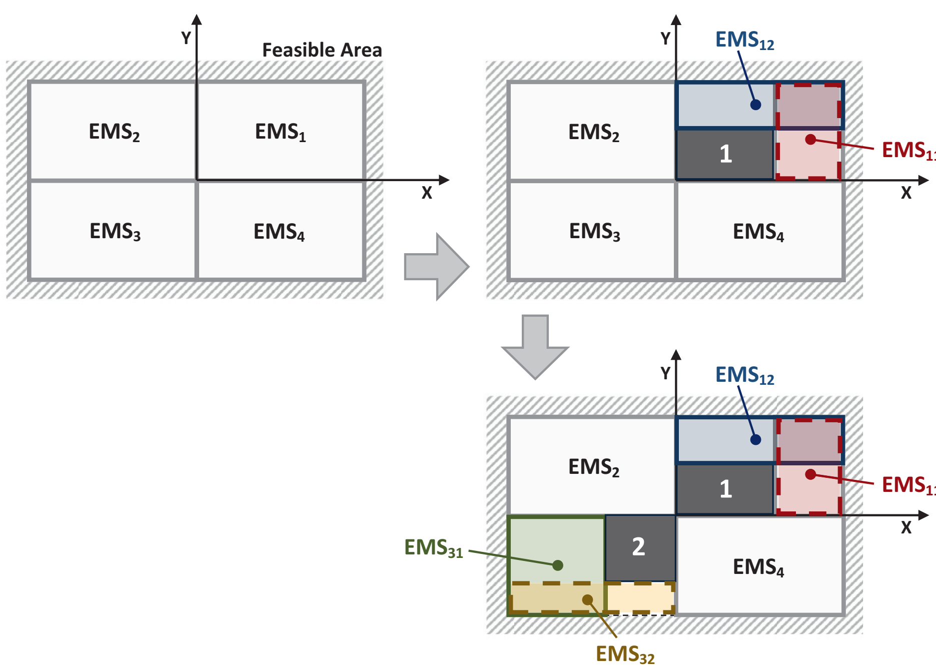


Fig. 2. Example of generation of new EMSs by the difference process (DP).

place the facilities on the floor space, a list $\mathcal{L}$ of empty maximal-spaces (EMS) is generated and maintained. Each EMS is a rectangular space with maximal area that has no overlap with the facilities already placed. An EMS e belonging to $\mathcal{L}$ is represented by its vertices (x^e_{min}, y^e_{min}) and (x^e_{max}, y^e_{max}). The construction procedure tests all the EMSs where the facility can be feasibly inserted, and chooses the best one according to Eq. (15). We use the difference process (DP) technique of Lai and Chan (1997) to update $\mathcal{L}$ after each insertion.

Depending on the type of facility location problem, a special treatment is applied to the first facility inserted. In the beginning, there are four EMSs available due to the division of space into quadrants. We discern the following cases:

1. If the first facility i is hard ($AR^{min}_i = AR^{max}_i$) and rotation is not admitted, then i is always inserted in the EMS1 with its lower left vertex in the origin.
2. If the first facility is soft ($AR^{min}_i < AR^{max}_i$) and rotation is not admitted, then we consider the placement of the second facility j. Two configurations are tested: 1) i and j are either placed side by side, with their centroids horizontally aligned, and having minimum aspect ratios, or 2) i and j are placed one on the top of the other with their centroids vertically aligned, and having maximum aspect ratios. The option that produces the smaller distance between centroids is selected.
3. Finally, if rotations are permitted, then one of the two orientations is arbitrarily chosen for i, and we proceed as in Case 1.

Fig. 2 illustrates the EMS and their updating mechanism during the insertion of two facilities. According to Case 1, the first facility is placed in EMS1 (Case 1), leading to two new EMSs. Note that the list $\mathcal{L}$ containing the EMSs is updated to eliminate those EMSs which cannot accommodate any facility, as well as those that are totally inscribed in other EMS. After this filtering, five EMS remain.

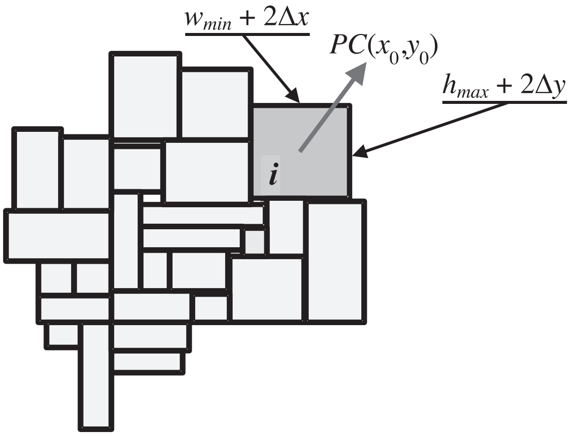


Fig. 3. Facility i being inserted with dimensions adjusted by the model.

Facility 2 is then placed in EMS3 and the list $\mathcal{L}$ is updated, leading to six EMSs, illustrated on the bottom of the figure. This process continues until all the facilities are placed.

3.1.2. Optimization of facility dimensions

The placement of each facility inside each EMS follows the same strategy as the one described in Gonçalves and Resende (2015) when all facilities are hard, i.e., each facility i is placed in the position that minimizes Eq. (14), being as close as possible to the geometric center of the placed facilities if i has no flow to them.

If the facility i is soft, then its dimensions are also optimized to minimize Eq. (14). To that extent, we divide this placement problem into three cases. If the facility i, when in the optimal position with minimum aspect ratio, does not touch the top and the bottom borders of the EMS, then it should be placed in this position with this aspect ratio. Analogously, if the facility i, when in the optimal position with maximum aspect ratio, does not touch the left and the right borders of the EMS, then it should be placed in this position with this aspect ratio. If none of the previous conditions occur, then the facility should be placed in one of the four corners of the EMS. For the sake of brevity, we describe only the case where it is placed in the bottom-left corner. The other cases are symmetric. In this case, the optimal dimensions of facility i are computed by solving the non-linear model of Eqs. (18)–(20). There, the cost function given by (14) is approximated by its first-order Taylor series expansion around the point (x_0, y_0), which corresponds to the temporary position of the facility i, obtained assuming that it has the minimum aspect ratio. If z_i is not differentiable in the point (x_0, y_0) (which may be the case for the rectilinear metric), then a sub-gradient is used for the approximation.

In the models (18)–(20), the continuous variables Δx and Δy denote, respectively, the variation undergone by x_0 and y_0 to optimize the partial objective function. Since the facility is necessarily at the bottom-left corner of the EMS with its centroid being at the point (x_0, y_0) when its aspect ratio is minimum, its dimensions w_i and h_i can be calculated as a function of Δx and Δy,

$$w_i = w_{min} + 2\Delta x; \tag{16}$$

$$h_i = h_{max} + 2\Delta y; \tag{17}$$

where w_{min} and h_{max} are the facility's width and height, respectively, when its aspect ratio is minimum. Fig. 3 illustrates a facility i indicating its width, height, and the subgradient vector of the partial cost function. Let w_{max} be the width of facility i when its aspect ratio is maximum. Then, the model to be solved is the following:

$$\text{Min} \quad PC_x(x_0, y_0)\Delta x + PC_y(x_0, y_0)\Delta y \tag{18}$$

$$\text{s.t.} \ (w_{min} + 2\Delta x)(h_{max} + 2\Delta y) = A \tag{19}$$

$$0 \le 2\Delta x \le w_{max} - w_{min}. \tag{20}$$

The objective function of Eq. (18) is a first-order approximation of the partial cost function (14) at (x_0, y_0). Constraint (19) ensures that the area of the inserted facility will not be changed. Constraint (20) ensures that the dimensions will respect the limits imposed by the aspect ratio. This model can be solved analytically using basic Lagrangian relaxation techniques, with optimal solution (Δx^*, Δy^*) given by

$$\Delta x^* = \frac{\sqrt{PC_x\, PC_y\, A}}{2\, PC_x} - \frac{w_{min}}{2}, \tag{21}$$

$$\Delta y^* = \frac{\sqrt{PC_x\, PC_y\, A}}{2\, PC_y} - \frac{h_{max}}{2}, \tag{22}$$

when Δx^* satisfies the constraints (20). Otherwise, the aspect ratio of facility i is set to its minimum (maximum) value when $\Delta x^* < 0$ ($2\Delta x^* > w_{max} - w_{min}$).

3.2. Initial population

The greedy construction tends to fill the floor space around the position of the first facility (say i) of the sequence in such a way that i is in the center of the final solution and the distance from i to each facility j tends to grow larger as the position of j in the sequence increases. As a result, good solutions usually have smaller facilities with more incident flow placed before larger facilities with less incident flow. Based on this observation, we promoted such features in the initial population of the GA, by sorting the facilities in an increasing order of the ratio between their areas and their total incident flow, denoted by $r_i = A_i / \sum_{j=1;\, j\neq i}^{n} f_{ij}$, and generating initial sequences using this order and some controlled randomization.

We use for this purpose a restricted list of candidates (RCL), whose size is governed by a parameter α. Iteratively, for $k = 1, \ldots, n$, the kth element of the sequence will be randomly selected from the RCL with uniform distribution. At any iteration k, let $\mathcal{S}_k$ be the set of facilities which have already been inserted, let $\bar{\mathcal{S}}_k$ be the set of facilities which have not been inserted, and let $\mathcal{S}_k^{\text{FLOW}}$ be the subset of facilities from $\bar{\mathcal{S}}_k$ which receive some flow from at least one facility in $\mathcal{S}_k$. The RCL is initialized and updated to contain the following elements:

If $\mathcal{S}_k^{\text{FLOW}} \neq \varnothing$, **then** RCL contains the $\ell = \min\{\alpha, |\mathcal{S}_k^{\text{FLOW}}|\}$ facilities with smallest r_i in $\mathcal{S}_k^{\text{FLOW}}$.

Else, the RCL contains the $\ell = \min\{\alpha, |\bar{\mathcal{S}}_k|\}$ facilities with smallest r_i in $\bar{\mathcal{S}}_k$.

Note that larger values of α lead to more randomness, and thus a larger number of possible initial sequences. When $\alpha = \infty$, any permutation of the n facilities can be generated with equal probability. Fig. 4 illustrates the generation of a chromosome for the initial population, with $\alpha = 5$.

The flow matrix is displayed on the left of the figure. On the top-right side, the sequence of facilities is sorted by increasing r_i. The next lines represent the intermediary RCLs and randomized sequences. Since no facility is placed at the first iteration, no facility has flow from a placed facility and RCL contains the first five facilities of the sorted sequence. As a possible outcome of the algorithm, the facility 6 is randomly selected and inserted in the first position. For the second RCL, only the facilities 1, 2, 3, 4 and 10 have flow from the facility 6 (already placed). The facility 4 is randomly selected and placed in the second position. The process is

$$Flow_{ij} = \begin{pmatrix} 0 & 0 & 2 & 1 & 0 & 4 & 3 & 0 & 0 & 5 \\ 0 & 0 & 3 & 0 & 2 & 2 & 8 & 0 & 10 & 0 \\ 2 & 3 & 0 & 0 & 0 & 6 & 3 & 0 & 5 & 0 \\ 1 & 0 & 0 & 0 & 7 & 1 & 0 & 5 & 0 & 0 \\ 0 & 2 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 2 & 6 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 3 & 8 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 10 & 5 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Sorted chromosome with all facilities

Sequence	1	2	7	6	4	8	3	5	10	9

First RCL with *α=5*

RCL	1	2	7	6	4

index randomly selected for the 1st position

First facility placed and the remaining sorted

Sequence	6	1	2	7	4	8	3	5	10	9

Second RCL with *α=5*

RCL	1	2	4	3	10

index randomly selected for the 2nd position

First two facilities placed and the remaining sorted

Sequence	6	4	1	2	7	8	3	5	10	9

Perturbed chromosome after *n* selections

Sequence	6	4	1	5	8	9	7	2	10	3

Fig. 4. Perturbation of chromosome for $\alpha = 5$.

Parent 1	**1**	2	**7**	**6**	4	5	**3**	8	10	**9**
Parent 2	2	1	3	4	8	9	6	7	5	10

selected parent

Child generated by *PX* operator

Child	**1**	2	**7**	**6**	4	8	**3**	5	10	**9**

Fig. 5. Example of position-based crossover (PX).

iterated until reaching the initial chromosome, illustrated on the bottom of the figure.

3.3. Selection and crossover

The selection of the parents is done randomly with a uniform distribution. Then, we apply the position-based crossover (PX – Syswerda & Palmucci, 1991), a variation of order crossover where the selected genes are not contiguous. This crossover randomly selects one parent and $N_{inherit}$ random genes of this parent. These genes are copied to the offspring, preserving their original positions. Finally, the remaining positions of the offspring are filled with the genes that have not been copied from the first parent, following their relative order in the second parent. This process is illustrated in Fig. 5, where the gray-colored genes have been chosen from parent 1.

In our implementation, we set $N_{inherit} = \frac{n}{2} + \frac{N_{dif}}{4}$, where N_{dif} is the number of unequal alleles (facilities) in the two parents. This strategy is based on the observation that, at the start of the algorithm, individuals are still very different and generating a child with 75 percent of its genes from one of the parents helps to avoid an excessive time up to convergence. On the other hand, when the population begins to converge and the individuals become similar, the value of $N_{inherit}$ approaches 50 percent of n, and selecting 50 percent of each parent avoids the generation of many duplicated children.

3.4. Phase I – complete problem

Algorithm 1 represents a pseudo-code for the phase I of both BGA and QGA. In Line 3, an initial population $P(0)$ of size *nPop* is initialized in the same way as described in Section 3.2. For each generated chromosome, a solution is constructed and evaluated in Line 4, based on the placement strategy described in Section 3.1 and on the greedy criterion parameter *GC* (Eqs. 14 or 15). The main loop at Lines 5–13 controls the number of generations, stopping only when the population has converged. For each iteration of this loop, the algorithm randomly selects two parents from population $P(t-1)$ in line 8 and applies the crossover

Algorithm 1 Genetic Algorithm for UA-FLP.

1: **procedure** QGA/BGA(α, *GC*, *nPop*)
2: $t \leftarrow 0$;
3: Initialize population $P(t)$;
4: Construct and evaluate each individual of $P(t)$;
5: **while** ($P(t)$ did not converge) **do** {Main loop. Control the number of generations }
6: $t \leftarrow t+1$;
7: **for** $i = 1, \ldots, nPop$ **do** {Generate offsprings by crossover}
8: Randomly select two parents $p_1, p_2 \in P(t-1)$;
9: $d \leftarrow$ Crossover(p_1, p_2) ;
10: Construct and evaluate individual d ;
11: **end for**
12: Select survivor individuals to form $P(t)$;
13: **end while**
14: **return** the best individual found so far S^*;
15: **end procedure**

Table 2
Comparison of DRQGA and the best literature results for the instances of the first group (new BKSs are marked with $*$)

Instance	VIP-PLANOPT		BRKGA		DRQGA ($nPop = 150$; $\alpha = 5$; $GC = z$)			BKS
	Best	T (seconds)	Best-10	T (seconds)	Best-10	Avg-10	T (seconds)	
L020	1157.00	0.3	**1130.00**	0.48	1142.00	1164.00	0.31	1130.00
L028	6447.25	1.5	**6014.07**	0.95	6131.75	6284.01	0.46	6014.07
L050	78244.68	7	69404.64	6.26	**69153.36**	69419.92	9.48	68903.33*
L062	3939362	4996	3685136.02	9.07	**3676132.00**	3707506.90	10.51	3650789.50*
L075	34396.38	13	31482.84	11.60	**30804.10**	31242.44	11.18	30476.20*
L100	538193.10	14	478910.09	57.03	**473097.90**	474404.32	16.88	470722.32*
L125A	288774.60	110	256860.78	83.58	**250849.42**	252135.03	44.76	247017.37*
L125B	1084451.00	70	**943140.06**	118.65	945076.14	947571.20	31.44	937516.83*
Gap (percent)	11.35	–	1.41	–	**1.00**	2.03	–	
Gap-LG (percent)	13.54	–	1.88	–	**0.83**	1.46	–	
Avg T (seconds)	–	651.48	–	35.95	–	–	15.63	

operator in line 9, as described in Section 3.3. As a result, *nPop* offsprings are generated and inserted in the population.

After the loop of Lines 7–11, the population has $2 \times nPop$ individuals. The selection of *nPop* individuals to compose the next generation $P(t)$ is made at Line 12. This process continues until one of the following three conditions hold:

- the cost of the best individual of $P(t)$ differs from the average cost by less than 0.05 percent,
- all individuals generated in the current iteration are equal to some individual in $P(t-1)$,
- or the population has not changed for 50 iterations: $P(t) = P(t-1) = \cdots = P(t-50)$.

3.5. Phase II – deconstruction and reconstruction

To improve the quality of the solutions obtained by the genetic algorithm, we finally apply a deconstruction and reconstruction phase, which re-optimizes separately each quadrant of the best found solution. For this second optimization, we simply apply the same GA as Algorithm 1. We refer to the complete algorithm as the Deconstruction and Reconstruction Quadrant-based Genetic Algorithm (DRQGA). A pseudo-code for the DRQGA is given by Algorithm 2. The DRQGA receives as parameters the size of population *nPop*, the randomness parameter α, the greedy criterion *GC*, and N_{iter}, which controls the number of times that each quadrant of the current solution is deconstructed and reconstructed. The overall pseudo-code is displayed in Algorithm 2.

Algorithm 2 QGA with Deconstruction and Reconstruction.

```
procedure DRQGA(nPop, α, GC, N_iter)
    S* ← QGA(α, GC, nPop);            { Phase 1 of the algorithm }
    for iter = 1, ..., N_iter do  { Phase 2 of algorithm: strategy of decomposition }
        for q = 1, ..., 4 do
            Generate P(t) with nPop individuals, perturbing the chromosome of S* based on α;
            Apply the steps 4–13 of Algorithm 1, restricting the construction algorithm only for the considered quadrant q and fixing the other variables;
            If a solution better than S* was found, update S*;
        end for
    end for
    return S*;
end procedure
```

4. Computational experiments

Extensive computational experiments have been performed to evaluate the performance of the proposed algorithms. The algorithms were coded in C++ and the experiments were conducted on a computer with Intel Core $i7-3770$ 3.4 gigahertz CPU and 11.7 gigabyte of memory running the Linux operating system. A single thread was used for the tests. To compare the performance of DRQGA against the previous algorithms for the unconstrained case, we used the classic 8 large instances from the literature (L020–L125B, group 1), the 100 recent benchmark instances from Gonçalves and Resende (2015) (RND010–RND100, group 2), which were constructed in such a way that the optimal solution is known, and 25 new randomly-generated instances (F050–F150, group 3). These new instances are used to test the performance of the quadrant-based decomposition approach for larger application cases, also with a different number of soft facilities. In this comparison, we set $nPop = 150$, $\alpha = 5$ and $GC = z$. Let us refer to this configuration as the *reference configuration*. The DRQGA is compared with the best previous approach from the literature: the BRKGA by Gonçalves and Resende (2015). We also establish a comparison with VIP-PLANOPT (VIP-PLANOPT, 2006). In Section 4.2, we will also report extensive experiments where we measure the effect of each parameter on the performance of the method as well as the comparison of the DRQGA against its variants BGA and QGA.

4.1. Performance comparison with previous algorithms

Table 2 compares the results of the DRQGA with the results of BRKGA for the instances of group 1. The BRKGA runs were conducted on a Intel Xeon E5-2630 2.3 gigahertz CPU with 16 gigabyte of physical memory, and the VIP-PLANOPT runs on a Pentium 4 2.4 gigahertz CPU. For the DRQGA, we report the best cost (Best-10), the average cost (Avg-10), and the average time (T (seconds)) over 10 runs. The last column of the table shows the costs of the best known solutions (BKS), where the values of the two first instances were obtained by the BRKGA, and the others ones (marked with $*$) are new solutions found in the experiments reported in this and the next subsections. In all presented tables, boldface numbers refer to the best solutions found in the corresponding experiment. Note that the DRQGA has improved the best solutions of the literature with smaller or similar processing times for the instances L050, L062, L075, L100 e L125A. A longer run (Section 4.2.4) also led to a new best solution for L125B.

The last three lines of Table 2 report the average gaps (Gap (percent)) obtained by each algorithm with respect to the best known solutions (BKS), the same average gaps excluding the instances with $n < 50$ (gap for large instances – Gap-LG (percent)), and the average times in seconds (Avg T (seconds)) for all instances. Only the best costs of 10 runs are available for the other approaches. Note that both the Gap (1.00 percent) and the Gap-LG (0.83 percent) of the DRQGA, as well as the average times $T = 15.63$, are lower than that for previous algorithms.

Table 3
Performance analysis on the instances of the second group.

n	GUROBI		BRKGA		BGA		
	T (seconds)	Best Gap (percent)	T (seconds)	Best Gap (percent)	T (seconds)	Best Gap (percent)	Avg Gap (percent)
10	3600	0.21	0.18	0.00	0.00	**0.00**	0.00
20	3600	0.01	0.61	0.00	0.03	**0.00**	0.00
30	3600	0.32	1.50	0.00	0.24	**0.00**	0.00
40	3600	2.37	2.87	0.00	0.83	**0.00**	0.00
50	3600	3.99	4.83	0.11	2.33	**0.00**	0.00
60	3600	16.65	7.29	0.02	4.19	**0.00**	0.01
70	3600	12.21	10.29	1.44	7.69	**0.00**	0.02
80	3600	22.31	14.34	3.31	19.55	**0.00**	0.21
90	3600	36.11	18.69	6.00	16.91	**0.01**	0.47
100	3600	101.78	23.58	7.36	18.65	**0.03**	0.53

Table 4
Impact of parameter choices on Gap (percent) and computational time T (seconds)

		Population size									
		100		150		200		300		500	
		Avg Gap (percent)	T (seconds)	Avg Gap (percent)	T (seconds)	Avg Gap (percent)	T (seconds)	Avg Gap (percent)	T (seconds)	Avg Gap (percent)	T (seconds)
$\alpha = \infty$	$GC = z$	0.86	47.61	0.80	68.92	0.86	95.97	0.51	174.61	0.20	443.28
$\alpha = 5$	$GC = z$	1.10	19.23	**0.83**	**20.71**	0.95	34.03	0.74	71.78	0.40	154.39
$\alpha = 10$	$GC = z$	1.22	25.94	0.89	52.57	0.90	62.75	0.54	119.55	0.40	248.78
$\alpha = \infty$	$GC = z^D$	1.33	28.92	0.88	59.09	0.72	101.45	0.59	188.17	0.16	434.22
$\alpha = 5$	$GC = z^D$	1.06	18.87	1.01	24.62	0.73	40.18	0.63	72.02	0.23	154.52
$\alpha = 10$	$GC = z^D$	1.10	25.27	0.80	44.41	0.82	83.28	0.62	111.00	0.40	257.33

For the instances of the group 2, we used the same parameters as for the group 1. Yet, we selected the BGA as the method of choice (instead of the DRQGA) as it obtained the best solutions for these instances. Table 3 compares the results of BGA with the best results of the literature, obtained by Gonçalves and Resende (2015), as well as with the GUROBI solver, run on a Mixed Integer Programming formulation proposed in Gonçalves and Resende (2015) for 3600 seconds. The table reports the gaps of the average (Avg Gap (percent)) and the best (Best Gap (percent)) solutions obtained by each method over 10 runs, as well as the average CPU time per run. Each percentage gap is calculated as the difference between solution cost and best known solution cost, divided by the latter. These values have been averaged over the 10 instances of each class. Detailed results of the proposed approaches, on all 100 instances, are included in Appendix.

The results in Table 3 show that the BGA has both the smallest gaps and CPU times when compared to others approaches. It is also remarkable that the best solution of BGA out of 10 runs is optimal in 98 out of 100 cases (see Table 11 in Appendix).

4.2. Sensitivity analyses

4.2.1. Parameter calibration and impact of the decomposition strategies

In this section, we report experiments conducted to find a good parameter calibration and to measure the impact of the new methodological elements. Our algorithms use three main parameters: the amount α of randomness in the initial population, the greedy criterion *GC*, and the population size *nPop*. For each parameter, we tested the following values:

- **Population size:** $nPop = 100$, 150, 200, 300, or 500 individuals;
- **Randomization parameter for initial population:** $\alpha = 5, 10$ or ∞;
- **Greedy criterion:** $GC = z$ (Eq. 14) or z^D (Eq. 15).

Thus, combining the five possible values of *nPop* with the three possible values of α and the two possible values of *GC*, 30 configurations were tested for each instance. Finally, the value $N_{iter} = 2$ has been chosen to balance the CPU time between both phases.

Table 4 contains the average percentage gaps (Avg Gap (percent)) between the best costs and the BKSs (supplied in Table 2) over all instances of the group 1 with $n \geq 50$, as well as the average times (T) in seconds, for the 30 tested configurations. The reference configuration, with parameters $nPop = 150; \alpha = 5; CG = z$, presents a good balance between solution quality and the computational time: 0.83 percent of gap for 20.71 seconds. This table also shows that the versions with larger populations have lower gaps, but at the cost of significantly higher CPU times.

Next, we measure the impact of the quadrant assumption and the deconstruction and reconstruction phase in the DRQGA approach. This impact is measured comparing the gaps of solutions found by BGA, DRQGA–P1 (QGA) and DRQGA–P2, with the reference configuration. We test it separately for the two groups of instances because they have very disparate characteristics.

The gaps of solution cost for all methods, on the first group of instances, are reported in Table 5. DRQGA-P2 achieves the best results on all instances, and the contribution of the decomposition phases can be clearly observed by comparing it with DRQGA-P1 and BGA. The decomposition phase helps to reduce the gap by a factor two with only a moderate increase of CPU time. DRQGA–P1 also obtained better solutions than BGA for six out of eight instances, with significantly less CPU time. This speed-up is a consequence of the decomposition strategy, as it enables to store the EMS into four smaller-size data structures, hence facilitating their update. Finally, BGA obtained two solutions of relatively low quality for the instances L075 and L125A, with soft facilities. Note that this performance improves significantly when a larger population is used (see Table 9). For all instances of group 1, the structural constraints have a significant positive impact.

For the instances of the group 2, the gaps are reported in Table 6. In this table, one can note that BGA finds the optimal

Table 5
Impact of the quadrants and decompositions for L020-125B instances.

Instance	Configuration: $nPop = 150$; $\alpha = 5$; $GC = z$								
	BGA			DRQGA–P1			DRQGA–P2		
	Best Gap (percent)	Avg Gap (percent)	T (seconds)	Best Gap (percent)	Avg Gap (percent)	T (seconds)	Best Gap (percent)	Avg Gap (percent)	T (seconds)
L020	1.59	3.12	0.40	3.097	3.81	0.25	**1.06**	**3.01**	0.31
L028	3.64	5.77	0.80	2.313	6.71	0.40	**1.96**	**4.49**	0.46
L050	0.41	1.09	13.43	0.603	1.09	7.96	**0.36**	**0.75**	9.48
L075	15.12	17.54	15.45	3.962	5.27	8.49	**1.08**	**2.51**	11.18
L100	0.92	1.54	24.95	0.739	1.14	12.68	**0.51**	**0.78**	16.88
L125A	15.12	17.00	71.88	2.823	3.65	33.38	**1.55**	**2.07**	44.76
L125B	1.00	1.82	44.95	0.969	1.64	23.95	**0.81**	**1.07**	31.44
L062	1.77	2.19	8.63	1.617	2.43	8.62	**0.69**	**1.55**	10.51

Table 6
Impact of the quadrants and decompositions for RND instances.

Instance	Configuration: $nPop = 150$; $\alpha = 5$; $GC = z$								
	BGA			DRQGA–P1			DRQGA–P2		
	Best Gap (percent)	Avg Gap (percent)	T (seconds)	Best Gap (percent)	Avg Gap (percent)	T (seconds)	Best Gap (percent)	Avg Gap (percent)	T (seconds)
RND010	**0.00**	**0.00**	0.00	0.92	0.98	0.02	0.88	0.96	0.04
RND020	**0.00**	**0.00**	0.03	0.90	1.27	0.19	0.88	1.19	0.31
RND030	**0.00**	**0.00**	0.24	1.61	2.31	0.50	1.25	2.06	0.84
RND040	**0.00**	**0.00**	0.83	1.31	1.97	1.37	1.06	1.81	1.78
RND050	**0.00**	**0.00**	2.33	1.28	1.81	2.14	1.05	1.58	3.08
RND060	**0.00**	**0.01**	4.19	1.98	2.86	4.09	1.68	2.46	5.40
RND070	**0.00**	**0.02**	7.69	1.69	2.51	6.95	1.31	2.16	8.10
RND080	**0.00**	**0.21**	19.55	2.99	3.98	9.23	2.47	3.24	11.60
RND090	**0.01**	**0.47**	16.91	2.43	3.91	12.53	1.71	2.71	16.39
RND100	**0.03**	**0.53**	18.65	3.06	4.75	14.55	2.33	3.48	20.96

solution for almost all instances, with best gaps equal or very close to 0.00 percent. On the other hand, the performance of the DRQGA is significantly worse. We believe that this is caused by the special structure of the optimal solutions for these instances, where the centroids of the all facilities are aligned along either the horizontal or the vertical line with some other facility, not favoring the construction of the solution in quadrants. This hypothesis is investigated in the next section.

4.2.2. Modified instances

As previously mentioned, the datasets of the group 2 were generated in such a way that the optimal solutions are known. In these instances, the flow matrices are sparse, and the optimal costs are given by the following equation:

$$OPT = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{ij} \min \left\{ \frac{w_i + w_j}{2}, \frac{h_i + h_j}{2} \right\}. \tag{23}$$

This means that each flow follows either an horizontal or a vertical path from its origin to its destination, corresponding to the smallest possible distance between the centroids of the corresponding facilities. To make these instances more realistic and challenging, while maintaining the sparsity of the flow matrices, we propose the following modification. Let f_{ij} and f'_{ij} be the flows from facility i to facility j in the original and the modified instance, respectively. We thus set:

$$f'_{ij} = \begin{cases} f_{ij} & \text{if } j \neq i+1 \\ f_{ij} + 1 & \text{if } j = i+1. \end{cases} \tag{24}$$

This modification breaks the structure of the optimal solution with only a small increase in the sum of all flows. In this way, we can investigate again the effect of the quadrant assumption and the deconstruction and reconstruction of the solutions. The resulting gaps and CPU times are reported in Table 7. This table follows the same format as Table 6 where the gaps are calculated with respect to the best solution found by the two algorithms tested. Comparing the phase 2 with the phase 1 of DRQGA, we notice that the decomposition phase is now able to reduce the best gap in almost all instances. For the larger instances, with $n > 60$, this reduction compensates the loss in the solution quality caused by the prohibition to cross the X and Y axis. This shows that even small changes of instance characteristics can lead to a variety of algorithmic behaviors.

4.2.3. New instances with different percentages of soft facilities

We also investigated the effect of the quadrant assumption and the deconstruction and reconstruction of solutions on a new benchmark composed of a larger variety of instances with different size and percentage of soft facilities. This benchmark is made of 25 randomly generated instances, described in Table 8. The five groups of five instances each have their number of facilities equal to 50, 75, 100, 125, and 150. Each group contains five instances named Fn_1, ..., Fn_5, one for each percentage of soft facilities in {0 percent, 10 percent, 25 percent, 50 percent, 100 percent}. For each instance, the area A_i of each facility i, for $i = 1, \ldots, n$, was randomly generated from an uniform distribution between 5 and 15. Now, assume that P_i is the perimeter of facility i, and let $P_i^* = 4\sqrt{A_i}$ be the minimum perimeter that facility i could have keeping its area unchanged. Following Tompkins et al. (2010), reasonable facility shapes are obtained only if the ratio $\frac{P_i}{P_i^*}$ does not exceed 1.4, which is equivalent to $\frac{w_i + h_i}{\sqrt{w_i \cdot h_i}} \leq 2.8$. From this inequality, we obtain the following range for the aspect ratio:

$$0.176 \leq \frac{w_i}{h_i} \leq 5.664 \tag{25}$$

Table 7
Impact of the quadrants and decompositions for MODIFIED-RND instances.

Instance	Configuration: $nPop = 150$; $\alpha = 5$; $GC = z$								
	BGA			DRQGA–P1			DRQGA–P2		
	Best Gap (percent)	Avg Gap (percent)	T (seconds)	Best Gap (percent)	Avg Gap (percent)	T (seconds)	Best Gap (percent)	Avg Gap (percent)	T (seconds)
RND010	**0.00**	**0.01**	0.03	0.90	1.08	0.02	0.90	1.07	0.05
RND020	**0.00**	**0.12**	0.17	1.62	2.62	0.19	1.59	2.50	0.27
RND030	**0.00**	**0.64**	0.46	1.98	3.28	0.46	1.65	2.89	0.61
RND040	**0.00**	**4.55**	1.30	3.62	7.89	1.05	2.58	5.91	1.33
RND050	**0.13**	**3.41**	2.93	2.75	6.70	1.66	1.76	4.77	2.17
RND060	**0.30**	**3.92**	4.34	2.93	7.50	2.74	1.53	4.86	3.67
RND070	1.69	5.70	5.79	4.61	8.65	4.09	**0.45**	**4.07**	5.66
RND080	1.74	7.59	10.13	4.91	10.72	6.82	**0.45**	**4.80**	8.24
RND090	4.24	9.88	13.40	5.91	12.15	10.06	**0.00**	**4.30**	12.30
RND100	6.45	12.23	20.36	7.69	14.69	12.92	**0.06**	**5.22**	16.17

Table 8
Impact of the quadrants and decompositions for new instances with $nPop = 150$, $\alpha = 5$ and $GC = z$

Instance	Configuration: $nPop = 150$; $\alpha = 5$; $GC = z$								
	BGA			DRQGA–P1			DRQGA–P2		
	Best Gap (percent)	Avg. Gap (percent)	T (seconds)	Best Gap (percent)	Avg. Gap (percent)	T (seconds)	Best Gap (percent)	Avg. Gap (percent)	T (seconds)
F050_1	8.47	10.19	5.25	3.71	4.64	2.55	**1.63**	**2.65**	3.31
F050_2	9.08	11.80	5.12	1.93	4.71	3.12	**0.73**	**2.38**	4.20
F050_3	8.74	10.28	5.40	3.68	4.58	3.48	**0.54**	**1.65**	4.48
F050_4	9.62	11.90	6.43	4.38	5.28	4.15	**0.37**	**2.43**	5.50
F050_5	9.87	12.08	9.46	3.71	5.07	4.99	**1.02**	**2.14**	7.26
F075_1	10.26	12.53	11.54	3.19	4.51	8.55	**1.18**	**2.02**	11.78
F075_2	9.27	11.29	13.33	4.60	5.53	10.19	**1.79**	**2.91**	14.45
F075_3	9.07	10.14	18.03	4.16	5.06	8.86	**0.86**	**2.42**	12.26
F075_4	8.75	10.85	18.51	3.89	5.41	11.81	**1.65**	**2.60**	16.28
F075_5	9.27	10.63	28.16	2.69	4.21	13.26	**0.56**	**1.43**	25.33
F100_1	10.48	11.69	28.32	2.89	4.09	15.90	**0.95**	**1.81**	22.36
F100_2	9.68	11.13	27.19	3.46	3.98	22.80	**0.98**	**1.57**	30.42
F100_3	10.08	10.65	33.07	3.87	4.65	24.86	**1.31**	**2.04**	34.02
F100_4	9.24	10.61	43.47	3.72	4.91	20.77	**1.48**	**1.98**	32.22
F100_5	10.04	11.83	56.93	3.13	4.28	28.63	**0.94**	**1.81**	40.23
F125_1	9.94	12.01	42.02	3.74	4.56	25.72	**1.49**	**2.24**	38.83
F125_2	12.08	12.78	45.71	3.07	4.16	36.42	**0.94**	**1.85**	51.58
F125_3	10.82	12.33	67.15	3.16	4.49	47.29	**1.22**	**2.32**	66.02
F125_4	11.44	12.79	78.12	3.67	4.39	43.87	**1.39**	**2.14**	62.63
F125_5	10.35	12.48	118.34	2.81	3.86	64.78	**1.03**	**1.85**	90.32
F150_1	12.42	13.64	71.45	3.59	4.25	83.03	**1.10**	**1.80**	106.82
F150_2	10.37	11.98	79.39	2.71	3.73	71.63	**1.14**	**1.80**	100.59
F150_3	10.52	11.45	118.59	2.65	3.71	81.37	**1.27**	**1.81**	109.52
F150_4	9.92	11.34	128.78	2.73	3.54	106.23	**1.06**	**1.61**	137.02
F150_5	11.23	12.84	154.27	3.12	3.67	103.47	**1.45**	**2.11**	135.48
	10.04	11.65	48.56	3.37	4.45	33.91	**1.12**	**2.05**	46.52

For the rigid facilities, we randomly generated a number AR_i in the interval given by (25), and set $AR_i^{min} = AR_i^{max} = AR_i$. For all soft facilities, we simply set $AR_i^{min} = 0.176$ and $AR_i^{max} = 5.664$.

We use these new instances to further analyze the impact of the quadrant assumption and the deconstruction and reconstruction phase in the DRQGA. This impact is measured by comparing the gaps of solutions found by BGA, DRQGA–P1 (QGA) and DRQGA–P2 (with the reference configuration) with respect to the best known solutions (BKS) listed in Table 10. The gaps of solution cost for all methods, on the third group of instances, are reported in Table 8. The last line of the table contains the average value of each column. DRQGA–P2 achieved the best results on all instances, and again the contribution of the decomposition phases can be clearly observed by comparing it with DRQGA-P1 and BGA. The decomposition phase helps to reduce the gap by about 7 percent, with a smaller average CPU time. Comparing DRQGA–P2 with BGA, an average improvement of 9.6 percent can be observed, with a small reduction on the CPU time. Yet, this improvement also indicates that the simple BGA performs poorly for the new set of instances when using these parameters, which are chosen to obtain a competitive CPU time. In the next section, we show that the performance of BGA improves substantially in longer runs, but the decomposition phase is still essential to reach higher quality solutions. We also note that the percentage of soft facility does not significantly impact the performance of the algorithms. As such, the technique introduced to optimize the dimensions of soft facilities appears to be effective.

4.2.4. Longer runs

Finally, in the cases where the CPU time is not critical, it is possible to select a parameter configuration designed for higher solution quality, such as $nPop = 500$, $\alpha = \infty$, and $GC = z^D$. Table 9 presents the results of the two algorithms using this configuration in the same format as Table 5 for the instances of group 1. We notice that all the best gaps are located below 2 percent, and all the average gaps for the instances with $n \geq 50$ are

Table 9
Impact of the quadrants and decompositions for L020-125B instances.

Instance	Configuration: $nPop = 500$; $\alpha = \infty$; $GC = z^D$								
	BGA			DRQGA–P1			DRQGA–P2		
	Best Gap (percent)	Avg Gap (percent)	T (seconds)	Best Gap (percent)	Avg Gap (percent)	T (seconds)	Best Gap (percent)	Avg Gap (percent)	T (seconds)
L020	1.02	1.66	1.89	1.28	2.41	0.88	0.49	1.76	1.77
L028	0.06	2.27	3.02	4.74	5.80	2.05	1.96	4.52	2.73
L050	0.37	0.77	105.10	0.54	0.83	96.25	0.00	0.46	110.54
L075	1.93	2.96	139.67	1.94	2.75	137.42	0.14	0.87	165.64
L100	0.07	1.05	306.78	0.26	1.02	294.95	0.00	0.43	334.01
L125A	1.44	2.99	3227.45	1.23	2.38	964.35	0.46	1.02	1071.18
L125B	0.71	1.72	565.31	0.51	1.02	781.40	0.34	0.64	861.09
L062	0.67	1.35	54.40	0.94	1.38	47.07	0.00	0.72	62.83

Table 10
Impact of the quadrants and decompositions for new instances with $nPop = 500$, $\alpha = 200$ and $GC = z^D$

Instance	Configuration: $nPop = 500$; $\alpha = 200$; $GC = z^D$									
	BGA			DRQGA–P1			DRQGA–P2			BKS
	Best Gap (percent)	Avg. Gap (percent)	T (seconds)	Best Gap (percent)	Avg. Gap (percent)	T (seconds)	Best Gap (percent)	Avg. Gap (percent)	T (seconds)	
F050_1	**0.00**	**1.35**	32.39	2.33	2.93	19.27	0.64	1.49	23.73	23432.36
F050_2	1.35	1.78	37.81	1.27	2.58	23.97	**0.00**	**1.01**	29.08	24390.73
F050_3	0.33	1.32	44.08	1.15	1.96	25.15	**0.00**	**1.02**	48.91	24417.46
F050_4	1.20	2.26	53.96	2.24	2.77	29.16	**0.00**	**1.26**	51.03	24134.87
F050_5	1.81	2.70	77.20	0.61	2.22	40.10	**0.00**	**0.99**	115.34	23620.18
F075_1	1.11	2.07	138.58	1.42	2.11	80.14	**0.00**	**0.67**	106.50	67525.82
F075_2	1.79	2.28	184.85	1.48	2.82	101.46	**0.00**	**0.92**	121.68	69309.44
F075_3	1.01	1.91	210.43	1.00	2.35	107.93	**0.00**	**0.74**	148.02	69419.58
F075_4	2.10	2.64	185.46	1.61	2.60	130.04	**0.00**	**0.98**	222.79	68941.44
F075_5	1.09	2.01	276.45	1.82	2.33	196.80	**0.00**	**0.80**	489.23	68165.40
F100_1	1.34	2.35	357.85	1.44	2.19	262.74	**0.00**	**0.66**	309.69	147987.06
F100_2	1.29	1.80	436.97	1.44	2.12	308.41	**0.00**	**0.59**	356.16	150628.57
F100_3	1.22	2.31	503.04	1.42	2.17	348.88	**0.00**	**0.32**	396.94	150165.52
F100_4	1.66	2.52	659.64	1.68	2.56	334.62	**0.00**	**0.58**	386.64	149519.43
F100_5	1.95	2.65	784.92	1.66	2.32	481.51	**0.00**	**0.33**	585.61	147432.17
F125_1	1.28	1.89	945.15	1.57	2.59	475.07	**0.00**	**0.66**	551.54	266801.12
F125_2	1.09	2.02	939.38	1.31	2.49	571.44	**0.00**	**0.43**	651.35	262805.18
F125_3	1.16	1.90	1543.48	1.23	2.20	707.59	**0.00**	**0.57**	796.95	261689.72
F125_4	1.66	2.64	1818.06	1.48	3.04	1007.36	**0.00**	**0.72**	1115.15	260059.24
F125_5	1.41	2.85	2035.39	1.56	1.89	1268.54	**0.00**	**0.35**	1420.85	257423.93
F150_1	0.99	2.43	1882.58	1.50	2.58	1363.20	**0.00**	**0.60**	1534.14	419162.73
F150_2	0.88	1.71	1398.84	1.10	2.24	1615.71	**0.00**	**0.50**	1812.07	416853.32
F150_3	0.83	1.85	1900.56	1.17	1.61	2303.54	**0.00**	**0.15**	2520.24	415074.35
F150_4	1.32	2.15	2210.28	1.20	1.69	2237.31	**0.00**	**0.32**	2471.54	412724.72
F150_5	1.82	2.82	2924.11	1.31	2.41	2444.23	**0.00**	**0.88**	2673.11	406670.90
	1.27	2.17	863.26	1.44	2.35	659.37	**0.03**	**0.70**	757.53	

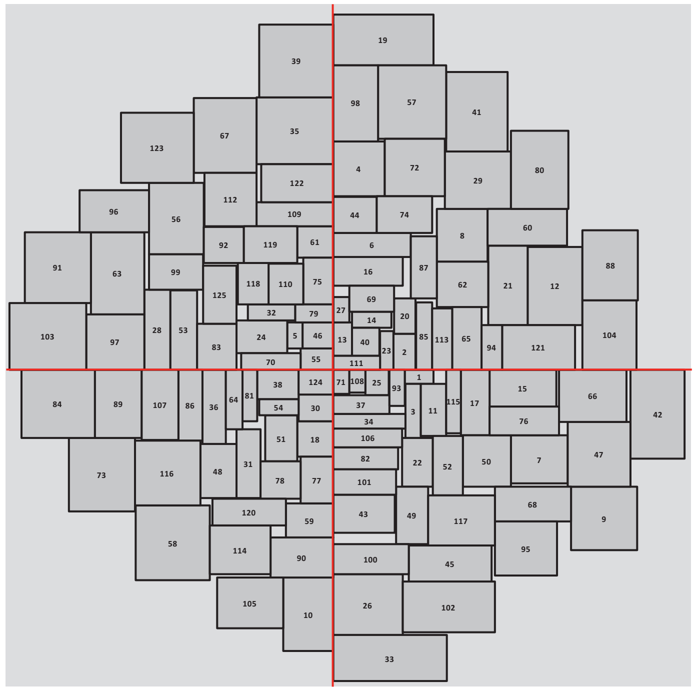
below 3 percent. This is naturally achieved at the expense of a larger CPU time. Moreover, the DRQGA–P2 obtains the best solutions for all instances with $n \geq 50$, and equals the BKS for the instances L050, L100, and L062. Two such new best solutions are displayed in Fig. 6, for the instances L075 and L125A.

Likewise, Table 10 presents the results of the two algorithms for longer runs on the new instances. The same format as Table 8 is used. DRQGA–P2 identified the current best known solutions for all instances except the first, as highlighted by a best gap of 0.00 percent. These best solution values are listed in the last column. In general, the difference of solution quality between BGA and DRQGA–P2 tends to be smaller than previously, although still significant. This means that BGA needs a more CPU intensive configuration to converge towards good solutions on the new instance. This larger CPU time is not needed when using the decomposition phases.

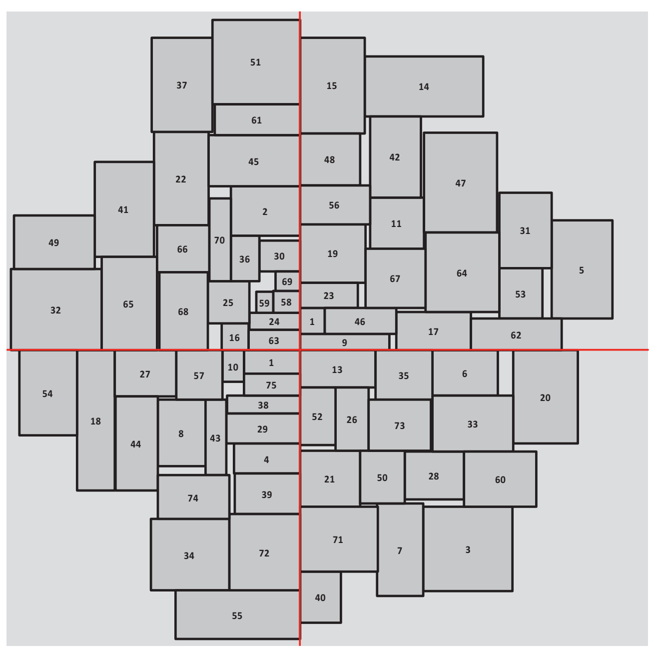
5. Concluding remarks

In this paper, we have introduced two alternative heuristics for the UA-FLP: a basic GA (BGA), which relies on a solution representation as a permutation of facilities and uses a greedy heuristic for their insertion on the floor space; and a GA with quadrant restrictions and decomposition phases (DRQGA). This more sophisticated method imposes an additional structure to the solutions, such that no facility should cross the X or Y axis, and uses this structure to perform efficient improvements during decomposition phases.

Both methods have been tested on classical benchmark instances from the literature, returning solutions of high quality. During the experiments on the first group of eight instances, intensively studied in previous literature, DRQGA improved the best known solutions for the largest six instances in small computational time. For the second group of instances of Gonçalves and Resende (2015), the BGA presented the best performance in the literature, obtaining 98 out of the 100 known optimal solutions. For some instances of group 1 and 2, we observed improvements of up to 4percent and 7 percent over previous literature. We finally report additional experiments on two sets of newly proposed instances. The first set results from a small modification introduced on the flow matrices of the



(a) L125A: Cost=250849.40; T*=49.66 seconds



(b) L075: Cost=30804.10; T*=10.75 seconds

Fig. 6. DRQGA solutions for datasets L125A and L075.

instances of group 2 to make them more realistic, and the second set is completely new, including five different sizes and five percentages of soft facilities, representing a larger subset of instance characteristics. On these new benchmark sets, the DRQGA performed significantly better than BGA for medium- and large-scale problems. Overall, the proposed structural constraints and decomposition approach appears to largely contribute to the search for hard instances with many (e.g., more than 50) facilities.

Acknowledgments

The authors would like to thank Imran Tasadduq and Abdul-Rahim Ahmad for making his datasets available and Maurício G. C. Resende and José Fernando Gonçalves for making their datasets and detailed results available. This research was supported by CAPES, CNPq (grant no. 301278/2013-0) and FAPERJ (grant no. E26-103.206/2011).

Appendix. Detailed computational results

Tables 11–13

Table 11
Solutions obtained by BGA for RND instances.

Instance	Best-10	Avg-10	T^*	T	Opt	Instance	Best-10	Avg-10	T^*	T	Opt
010-001	3728.50	3728.50	0.00	0.01	3728.50	060–001	16561.50	16561.50	3.62	4.63	16561.50
010-002	2865.50	2865.50	0.00	0.01	2865.50	060–002	23959.00	23959.00	1.86	2.48	23959.00
010-003	3679.50	3679.50	0.00	0.00	3679.50	060–003	18691.00	18691.00	1.18	5.75	18691.00
010-004	3915.00	3915.00	0.00	0.00	3915.00	060–004	18196.50	18218.45	2.41	3.03	18196.50
010-005	3235.00	3235.00	0.00	0.00	3235.00	060–005	21292.50	21292.50	0.00	3.29	21292.50
010-006	3650.50	3650.50	0.00	0.00	3650.50	060–006	18879.50	18879.50	0.00	4.18	18879.50
010-007	3771.00	3771.00	0.00	0.00	3771.00	060–007	17023.00	17023.00	0.00	3.81	17023.00
010-008	2919.00	2919.00	0.00	0.00	2919.00	060–008	22292.50	22292.50	0.00	3.39	22292.50
010-009	4404.00	4404.00	0.00	0.00	4404.00	060–009	18580.00	18580.00	0.00	6.10	18580.00
010-010	2764.00	2764.00	0.00	0.00	2764.00	060–010	17981.00	17983.10	0.02	5.28	17981.00
020-001	5865.50	5865.50	0.00	0.04	5865.50	070–001	22698.00	22703.60	0.04	10.47	22698.00
020-002	6202.50	6202.50	0.00	0.01	6202.50	070–002	22033.50	22039.10	0.00	5.66	22033.50
020-003	6158.00	6158.00	0.00	0.01	6158.00	070–003	24113.50	24113.50	0.00	5.85	24113.50
020-004	7572.00	7572.00	0.00	0.03	7572.00	070–004	23191.00	23210.00	5.80	8.10	23191.00
020-005	6317.00	6317.00	0.00	0.01	6317.00	070–005	24423.50	24426.50	5.56	8.27	24423.50
020-006	6784.50	6784.50	0.00	0.04	6784.50	070–006	18089.50	18089.50	0.00	11.81	18089.50
020-007	5814.50	5814.50	0.00	0.03	5814.50	070–007	24606.00	24619.20	7.75	9.35	24606.00
020-008	5854.00	5854.00	0.00	0.06	5854.00	070–008	20609.00	20609.00	3.76	4.58	20609.00
020-009	6047.00	6047.00	0.00	0.02	6047.00	070–009	19134.50	19134.50	0.01	8.78	19134.50
020-010	7867.50	7867.50	0.00	0.02	7867.50	070–010	23293.00	23294.90	0.05	4.11	23293.00
030-001	9422.50	9422.50	0.00	0.38	9422.50	080–001	24440.50	24489.85	10.89	12.13	24440.50
030-002	9143.00	9143.00	0.00	0.28	9143.00	080–002	24146.00	24267.05	11.33	12.67	24146.00
030-003	7335.50	7335.50	0.00	0.03	7335.50	080–003	30018.00	30085.00	21.13	28.00	30018.00
030-004	8569.00	8569.00	0.00	0.25	8569.00	080–004	25012.00	25037.90	0.07	38.54	25012.00
030-005	10558.00	10558.00	0.00	0.34	10558.00	080–005	27869.00	27869.00	0.02	27.91	27869.00
030-006	10974.00	10974.00	0.00	0.22	10974.00	080–006	25329.00	25404.40	5.45	5.92	25329.00
030-007	9807.00	9807.00	0.00	0.14	9807.00	080–007	27030.00	27037.80	0.05	11.48	27030.00
030-008	9169.50	9169.50	0.00	0.33	9169.50	080–008	25426.50	25525.55	3.71	17.48	25426.50
030-009	10057.50	10057.50	0.00	0.19	10057.50	080–009	24547.50	24559.30	0.02	23.37	24547.50
030-010	9976.00	9976.00	0.00	0.29	9976.00	080–010	23955.50	24019.55	17.16	17.98	23955.50
040-001	11243.50	11243.50	0.00	0.71	11243.50	090–001	30235.50	30399.25	9.91	10.81	30235.50
040-002	12838.50	12838.50	0.00	0.87	12838.50	090–002	32660.00	32717.15	0.01	12.53	32660.00
040-003	15173.00	15173.00	0.00	1.05	15173.00	090–003	26151.00	26175.20	0.07	21.13	26131.00
040-004	12275.50	12275.50	0.00	0.14	12275.50	090–004	28635.00	28760.25	11.69	13.23	28635.00
040-005	13428.50	13428.50	0.00	1.02	13428.50	090–005	22671.50	22688.20	0.04	22.58	22671.50
040-006	13461.50	13461.50	0.00	0.83	13461.50	090–006	30003.50	30489.85	0.05	21.01	30003.50
040-007	13473.00	13473.00	0.00	0.73	13473.00	090–007	27487.50	27502.60	9.59	13.70	27487.50
040-008	14522.00	14522.00	0.00	0.89	14522.00	090–008	29370.50	29488.50	0.01	10.50	29370.50
040-009	14881.00	14881.00	0.00	0.85	14881.00	090–009	32194.00	32474.05	0.05	14.23	32194.00
040-010	12300.50	12300.50	0.00	1.18	12300.50	090–010	30860.00	30958.65	0.09	29.39	30860.00
050-001	17320.50	17320.50	0.00	3.05	17320.50	100–001	33332.50	33646.00	11.050	12.645	33332.50
050-002	16829.50	16829.50	0.00	1.73	16829.50	100–002	30841.00	30971.30	15.759	17.358	30841.00
050-003	16589.50	16589.50	0.02	2.15	16589.50	100–003	31855.50	31975.10	0.094	24.476	31755.00
050-004	15654.00	15654.00	0.00	1.57	15654.00	100–004	34566.00	34663.40	21.398	22.931	34566.00
050-005	15581.50	15581.50	0.00	2.88	15581.50	100–005	34163.50	34457.30	13.403	16.516	34163.50
050-006	15984.50	15984.50	0.00	1.82	15984.50	100–006	32345.50	32436.60	20.798	20.904	32345.50
050-007	17018.50	17019.50	1.91	2.19	17018.50	100–007	34029.00	34210.40	0.095	32.835	34029.00
050-008	15077.50	15077.50	0.00	2.46	15077.50	100–008	33594.50	33765.63	8.587	8.924	33594.50
050-009	14132.50	14132.50	0.01	3.77	14132.50	100–009	32824.00	33002.15	16.590	17.733	32824.00
050-010	15764.00	15764.00	0.00	1.63	15764.00	100–010	34660.50	34761.15	11.098	12.167	34660.50

Table 12
Solutions obtained by BGA for the modified RND instances.

Instance	Best-10	Avg-10	T^*	T	BKS	Instance	Best-10	Avg-10	T^*	T	BKS
010-001	4097.00	4097.00	0.00	0.03	4097.00	060–001	20805.00	21627.80	4.51	4.82	20805.00
010-002	3138.00	3138.00	0.00	0.02	3138.00	060–002	29906.50	31294.32	5.37	5.49	29906.50
010-003	3943.00	3943.00	0.00	0.02	3943.00	060–003	24389.50	25686.07	5.82	6.19	24389.50
010-004	4210.50	4210.50	0.00	0.02	4210.00	060–004	21893.00	22307.23	2.17	2.55	21893.00
010-005	3419.50	3422.70	0.00	0.06	3419.50	060–005	25992.00	26545.55	3.10	3.89	25992.00
010-006	3782.00	3782.00	0.00	0.02	3782.00	060–006	23397.50	24516.57	4.08	4.92	23397.50
010-007	4038.50	4038.50	0.00	0.02	4038.00	060–007	20927.50	21474.09	1.87	2.55	20927.50
010-008	3204.50	3204.50	0.00	0.01	3204.50	060–008	26116.50	26300.94	2.27	2.59	26116.50
010-009	4635.50	4635.50	0.00	0.02	4635.50	060–009	26529.20	26824.12	5.61	5.98	25750.50
010-010	2976.00	2976.00	0.00	0.02	2976.00	060–010	24193.83	26374.04	4.11	4.37	24193.83
020-001	6562.50	6562.50	0.06	0.13	6562.50	070–001	30748.00	31550.82	5.79	6.28	29458.50
020-002	6948.50	6948.50	0.06	0.16	6948.50	070–002	31396.09	32992.69	5.59	5.73	30670.50
020-003	6864.50	6915.30	0.17	0.38	6864.50	070–003	34706.93	36648.81	5.20	5.48	33448.00
020-004	8585.00	8585.00	0.07	0.13	8585.00	070–004	28662.83	29818.25	6.13	6.80	28662.83
020-005	6992.00	7001.82	0.08	0.19	6992.00	070–005	32731.50	35263.38	3.79	4.43	32731.50

(*continued on next page*)

Table 12 (*continued*)

Instance	Best-10	Avg-10	T^*	T	BKS	Instance	Best-10	Avg-10	T^*	T	BKS
020-006	7346.00	7347.10	0.10	0.16	7346.00	070–006	25414.50	26005.77	7.14	7.37	25122.50
020-007	6437.00	6458.60	0.14	0.21	6437.00	070–007	33688.00	35098.08	5.80	6.95	32722.00
020-008	6652.50	6652.50	0.00	0.08	6652.50	070–008	28173.50	29657.65	4.54	5.13	27931.00
020-009	6797.00	6797.00	0.05	0.12	6797.00	070–009	25634.00	25921.93	5.02	5.50	25272.50
020-010	8639.00	8639.00	0.00	0.09	8639.00	070–010	28247.00	28670.66	3.83	4.24	28247.00
030-001	10505.50	10514.12	0.25	0.42	10505.50	080–001	35586.44	37764.80	14.85	15.16	35219.50
030-002	10566.00	10566.00	0.06	0.25	10566.00	080–002	31612.50	32678.23	7.49	8.72	31612.50
030-003	8890.00	8966.75	0.37	0.58	8890.00	080–003	40657.33	43232.44	9.10	9.76	38349.00
030-004	9600.00	9600.00	0.23	0.35	9600.00	080–004	34092.06	35928.66	9.46	9.97	34092.06
030-005	12087.50	12087.50	0.69	0.85	12087.50	080–005	40603.25	42187.42	10.16	10.52	39126.50
030-006	12374.50	12896.63	0.11	0.29	12374.50	080–006	35329.00	37849.80	10.03	10.67	35329.00
030-007	11224.50	11363.75	0.34	0.47	11224.50	080–007	39437.00	41402.13	8.56	8.93	37502.00
030-008	10541.50	10544.00	0.35	0.55	10541.50	080–008	35517.00	37173.71	6.23	6.85	35014.00
030-009	11486.50	11486.50	0.26	0.49	11486.50	080–009	31952.50	35309.65	10.66	12.36	31952.00
030-010	11339.00	11339.00	0.23	0.37	11339.00	080–010	31686.06	33304.89	7.76	8.36	31686.06
040-001	14151.00	15584.89	1.50	1.71	14151.00	090–001	46518.10	48069.67	13.33	14.50	44486.00
040-002	14794.00	15633.43	0.92	1.14	14794.00	090–002	44864.50	46847.07	12.22	13.69	44410.50
040-003	17393.50	17453.65	0.91	1.24	17393.50	090–003	43354.53	44351.22	10.91	10.97	39407.00
040-004	14311.00	15767.18	1.33	1.52	14311.00	090–004	42659.79	44052.66	16.72	17.06	40792.50
040-005	15508.50	15678.78	0.71	0.94	15508.50	090–005	36570.38	38034.12	10.29	10.67	36025.50
040-006	15721.00	16122.65	0.81	1.00	15721.00	090–006	40963.50	43591.30	11.97	12.75	40768.50
040-007	15564.81	15899.76	1.11	1.44	15564.81	090–007	40844.60	43923.15	13.01	13.24	37712.50
040-008	16489.00	16489.00	0.44	0.73	16489.00	090–008	44109.00	46185.40	12.95	13.40	41443.00
040-009	17408.50	18205.07	1.04	1.26	17408.50	090–009	48614.21	51952.67	14.32	15.85	46169.50
040-010	14613.00	15903.95	1.66	2.03	14613.00	090–010	42971.00	47923.72	10.92	11.91	42883.00
050-001	20331.00	20825.59	1.99	2.27	20331.00	100–001	50242.19	52760.73	19.31	19.79	47139.00
050-002	21286.50	22069.70	2.32	2.56	21286.50	100–002	47919.80	50396.23	21.38	22.26	45312.00
050-003	20139.77	20794.97	3.16	3.16	20139.77	100–003	51357.38	54330.07	14.95	16.64	47828.50
050-004	18429.50	19169.14	1.70	2.12	18429.50	100–004	51627.51	54776.74	17.56	18.72	48129.00
050-005	19016.50	19456.15	3.60	3.87	19016.50	100–005	50463.29	53821.37	25.51	26.04	47486.00
050-006	20154.50	21354.30	4.04	4.29	20154.50	100–006	52251.50	55672.99	18.37	19.01	47059.00
050-007	20942.00	22134.53	2.72	3.48	20942.00	100–007	50421.50	52863.51	18.08	18.85	47337.00
050-008	17652.00	18132.40	1.92	2.13	17652.00	100–008	49355.50	51335.40	17.25	18.82	46519.00
050-009	18020.50	18370.78	2.86	3.25	18020.50	100–009	46098.00	48292.01	16.65	18.86	46098.00
050-010	19396.50	19544.74	1.53	2.14	19148.50	100–010	52295.72	55061.04	23.12	24.58	48586.00

Table 13
Solutions obtained by DRQGA for the modified RND instances.

Instance	Best-10	Avg-10	T^*	T	BKS	Instance	Best-10	Avg-10	T^*	T	BKS
010-001	4159.00	4163.10	0.01	0.04	4097.00	060–001	21081.97	22050.10	3.80	4.06	20805.00
010-002	3150.50	3166.25	0.01	0.09	3138.00	060–002	30209.36	31216.94	4.37	4.49	29906.50
010-003	3958.00	3958.00	0.01	0.04	3943.00	060–003	25241.00	26085.76	3.23	3.69	24389.50
010-004	4336.00	4340.20	0.00	0.02	4210.00	060–004	22478.00	22894.15	2.21	2.71	21893.00
010-005	3419.50	3419.50	0.03	0.08	3419.50	060–005	26124.00	26659.40	2.51	3.02	25992.00
010-006	3809.00	3824.60	0.01	0.03	3782.00	060–006	23846.00	24480.25	3.88	4.25	23397.50
010-007	4038.50	4038.50	0.00	0.02	4038.00	060–007	21272.50	22266.30	2.21	3.04	20927.50
010-008	3236.50	3254.20	0.00	0.07	3204.50	060–008	26643.00	27135.10	2.57	3.03	26116.50
010-009	4681.50	4681.50	0.01	0.04	4635.50	060–009	25750.50	26515.44	3.95	4.29	25750.50
010-010	3006.00	3006.00	0.01	0.04	2976.00	060–010	24366.50	25712.01	4.44	5.18	24193.83
020-001	6602.00	6640.75	0.02	0.30	6562.50	070–001	29458.50	30616.05	6.94	7.51	29458.50
020-002	7074.50	7084.40	0.17	0.24	6948.50	070–002	30670.50	31752.43	6.06	6.25	30670.50
020-003	6905.50	7053.15	0.35	0.43	6864.50	070–003	33448.00	34411.17	5.78	5.84	33448.00
020-004	8688.50	8737.70	0.14	0.17	8585.00	070–004	28880.50	30230.68	5.94	7.27	28662.83
020-005	7147.00	7244.85	0.18	0.30	6992.00	070–005	33066.50	34078.95	8.96	9.11	32731.50
020-006	7357.00	7563.90	0.12	0.35	7346.00	070–006	25122.50	26240.90	6.32	6.71	25122.50
020-007	6601.00	6604.65	0.08	0.18	6437.00	070–007	32722.00	34273.45	5.12	5.88	32722.00
020-008	6792.50	6823.50	0.08	0.21	6652.50	070–008	27931.00	29082.75	5.99	6.42	27931.00
020-009	6874.00	6930.20	0.09	0.28	6797.00	070–009	25272.50	25943.43	5.94	6.88	25272.50
020-010	8946.00	8946.00	0.06	0.28	8639.00	070–010	29026.50	29609.40	6.06	7.01	28247.00
030-001	10711.50	10732.85	0.39	0.48	10505.50	080–001	35219.50	36722.25	10.64	10.76	35219.50
030-002	10909.00	10946.75	0.24	0.46	10566.00	080–002	31835.50	32620.35	6.38	6.45	31612.50
030-003	8976.00	9121.82	0.52	0.70	8890.00	080–003	38349.00	40529.40	12.27	12.42	38349.00
030-004	9725.00	10010.83	0.39	0.69	9600.00	080–004	34354.50	36042.75	10.86	11.95	34092.06
030-005	12344.00	12345.00	0.33	0.45	12087.50	080–005	39126.50	40799.40	13.46	13.95	39126.50
030-006	12616.00	12939.95	0.65	0.71	12374.50	080–006	35440.00	36189.80	7.11	7.19	35329.00
030-007	11530.50	11651.08	0.66	0.75	11224.50	080–007	37502.00	38981.69	8.85	9.02	37502.00
030-008	10658.00	10878.80	0.79	0.94	10541.50	080–008	35014.00	36614.59	7.82	8.20	35014.00
030-009	11519.00	11618.15	0.57	0.67	11486.50	080–009	32069.00	34213.14	15.97	16.25	31952.00
030-010	11434.50	11491.90	0.36	0.41	11339.00	080–010	32412.00	33859.02	7.86	8.52	31686.06
040-001	15055.50	15636.50	1.22	1.22	14151.00	090–001	44486.00	46083.02	21.70	22.70	44486.00
040-002	15064.00	15671.46	1.43	1.69	14794.00	090–002	44410.50	46023.15	11.04	11.45	44410.50
040-003	17764.50	17949.65	0.87	0.91	17393.50	090–003	39407.00	41346.53	13.57	13.65	39407.00
040-004	14493.00	15775.89	1.29	1.45	14311.00	090–004	40792.50	41722.57	14.78	15.40	40792.50

(*continued on next page*)

Table 13 (*continued*)

Instance	Best-10	Avg-10	T^*	T	BKS	Instance	Best-10	Avg-10	T^*	T	BKS
040-005	15630.50	15886.90	1.06	1.08	15508.50	090–005	36025.50	36497.83	9.60	10.01	36025.50
040-006	16221.00	16648.50	1.57	1.85	15721.00	090–006	40768.50	42492.83	11.95	12.39	40768.50
040-007	15722.00	16309.00	2.47	2.77	15564.81	090–007	37712.50	40285.60	18.25	18.63	37712.50
040-008	16692.50	16796.80	0.42	1.00	16489.00	090–008	41443.00	43570.25	16.07	16.23	41443.00
040-009	17879.00	18082.65	0.69	0.97	17408.50	090–009	46169.50	48194.04	13.47	14.09	46169.50
040-010	15388.00	16121.94	2.83	3.03	14613.00	090–010	42883.00	45758.35	11.62	11.89	42883.00
050-001	20752.50	21246.30	2.01	2.18	20331.00	100–001	47139.00	49462.95	26.91	27.17	47139.00
050-002	21323.50	22089.70	1.80	2.03	21286.50	100–002	45312.00	47621.86	21.98	22.37	45312.00
050-003	20362.50	20862.28	1.78	1.92	20139.77	100–003	47828.50	50175.53	25.66	25.81	47828.50
050-004	18916.50	19628.20	2.23	2.65	18429.50	100–004	48129.00	50962.12	25.68	26.52	48129.00
050-005	19575.00	19862.75	2.46	2.55	19016.50	100–005	47486.00	50645.85	24.77	25.30	47486.00
050-006	21032.00	21855.36	2.53	2.99	20154.50	100–006	47059.00	49246.20	13.30	13.53	47059.00
050-007	21497.50	22652.74	1.84	2.13	20942.00	100–007	47337.00	49505.59	12.94	13.29	47337.00
050-008	17937.00	18201.05	1.99	2.26	17652.00	100–008	46519.00	49225.92	12.58	12.64	46519.00
050-009	18036.00	18528.05	1.68	1.76	18020.50	100–009	46389.00	47875.02	14.36	14.71	46098.00
050-010	19148.50	19600.95	1.39	1.67	19148.50	100–010	48586.00	51412.32	24.66	24.72	48586.00

References

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

Dunker, T., Radons, G., & Westkämper, E. (2003). A coevolutionary algorithm for a facility layout problem. *International Journal of Production Research, 41*(15), 3479–3500.

Garey, M. R., Johnson, D. S., & Stockmeyer, L. (1976). Some simplified np-complete graph problems. *Theoretical computer science, 1*(3), 237–267.

Gonçalves, J. F., & Resende, M. G. C. (2015). A biased random-key genetic algorithm for the unequal area facility layout problem. *European Journal of Operational Research, 246*(1), 86–107.

Imam, M. H., & Mir, M. (1993). Automated layout of facilities of unequal areas. *Computers & Industrial Engineering, 24*, 355–366.

Imam, M. H., & Mir, M. (1998). Cluster boundary search algorithm for building-block layout optimization. *Advances in Engineering Software, 29*(2), 165–173.

Kado, K. (1996). *An investigation of genetic algorithms for facility layout problems*. Ph.D. thesis University of Edinburgh.

Komarudin, K. Y. W. (2010). Applying ant system for solving unequal area facility layout problems. *European Journal of Operational Research, 202*(3), 730–746.

Kulturel-Konak, S., & Konak, A. (2011). A new relaxed flexible bay structure representation and particle swarm optimization for the unequal area facility layout problem. *Engineering Optimization, 43*(12), 1263–1287.

Lai, K. K., & Chan, J. W. M. (1997). Developing a simulated annealing algorithm for the cutting stock problem. *Computers & Industrial Engineering, 32*(1), 115–127.

McKendall, A. R., & Hakobyan, A. (2010). Heuristics for the dynamic facility layout problem with unequal-area departments. *European Journal of Operational Research, 201*(1), 171–182.

Mir, M., & Imam, M. H. (1996). Analytic annealing for macrocell placement optimization. *Computers & Electrical Engineering, 22*, 169–177.

Mir, M., & Imam, M. H. (2001). A hybrid optimization approach for layout design of unequal-area facilities. *Computers & Industrial Engineering, 39*(1), 49–63.

Scholz, D., Petrick, A., & Domschke, W. (2009). Stats: a slicing tree and tabu search based heuristic for the unequal area facility layout problem. *European Journal of Operational Research, 197*(1), 166–178.

Singh, S. P., & Sharma, R. R. K. (2006). A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology, 30*(5–6), 425–433.

Syswerda, G., & Palmucci, J. (1991). The application of genetic algorithms to resource scheduling. In *Proceedings of International Computer Games association, ICGA* (pp. 502–508).

Taillard, E. D., & Voss, S. (2002). Popmusic–partial optimization metaheuristic under special intensification conditions. In *Essays and surveys in metaheuristics* (pp. 613–629). Springer.

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

VIP-PLANOPT (2006). VIP-PLANOPT 2006 user's manual. Engineering Optimization Software. http://www.planopt.com.