

A Novel Island Model Based on Coral Reefs Optimization Algorithm for Solving the Unequal Area Facility Layout Problem

L. Garcia-Hernandez^a, L. Salas-Morera^a, C. Carmona-Muñoz^a, J.A. Garcia-Hernandez^a,
S. Salcedo-Sanz^b

e-mail: ir1gahel@uco.es, lsalas@uco.es, i32camuc@uco.es, jagh1977@gmail.com, sancho.salcedo@uah.es

^a*Area of Project Engineering, University of Cordoba, Spain.*

^b*Department of Signal Processing and Communications, Universidad de Alcalá, Alcalá de Henares, Madrid, Spain.*

Abstract

This paper proposes a novel approach to address the Unequal Area Facility Layout Problem (UA-FLP), based on the combination of both an Island Model and a Coral Reefs Optimization (CRO) algorithm. Two different versions of this Island Model based on Coral Reefs Optimization Algorithm (IMCRO) are proposed and applied to the UA-FLP. The structure of flexible bays has been selected as effective encoding to represent the facility layouts within the algorithm. The two versions of the proposed approach have been tested in 22 UA-FLP cases, considering small, medium and large size categories. The empirical results obtained are compared with previous state of the art algorithms, in order to show the performance of the IMCRO. From this comparison, it can be extracted that both versions of the proposed IMCRO algorithm show an excellent performance, accurately solving the UA-FLP instances in all the size categories.

Keywords: UA-FLP, Facility Layout, Island Model, Coral Reefs Optimization, Meta-heuristics

1. Introduction

Nowadays, intelligent industrial design aims at improving both efficiency and productivity of the industrial resources, trying this way to increase the final product quality (Abraham et al., 2020). The distribution of the elements that conform an industrial plant has been considered to be a very relevant issue, since it is known that a good design of a physical plant can substantially affect the production costs and efficiency in an industry (Singh and Sharma, 2006). This affirmation was also performed by Drira et al. (2007), who also identified and described a large number of different Facility Layout Problems (FLPs). Considering the FLP problem's taxonomy, the Unequal Area Facility Layout Problem (UA-FLP) proposed by Armour and Buffa (1963) has been identified as one of the most useful and studied FLPs (Palomo-Romero et al., 2017). In the UA-FLP, the elements to be distributed are *rectangular facilities* or departments, within a known area that is also rectangular. Additionally, overlapping of elements is not allowed. Normally, the main objective to take into

account in the UA-FLP concerns the minimization of the material flow that exists among the elements of the production system (Gonçalves and Resende, 2015).

To deal with the UA-FLP, different algorithms have been proposed in the literature, though meta-heuristics techniques have been the most successful approaches to this kind of problems by far (Garcia-Hernandez et al., 2013b; Gonçalves and Resende, 2015; Wong and Komarudin, 2010; Garcia-Hernandez et al., 2019). In order to encode UA-FLP instances within meta-heuristics, there are basically three main layout representations: Block Layout (BL), Slicing Trees Structure (STS) and Flexible Bays Structure (FBS). The last one was suggested by Tong (1991), and it has been the most used in the related literature because of its simple structure that does not add excessive computational complexity to the approach, and also because it allows the possibility of incorporating corridors in the design (Wong and Komarudin, 2010).

In spite of their good performance, meta-heuristics methods have shown some disadvantages, such as local optima stagnation (Kurdi, 2016) and the large amount of computational time needed to reach satisfactory solutions (Sitkoff, 1995). In this sense, parallel computing arises to try to solve this issue, because of its capacity to preserve population diversity thanks to the simultaneous evolution of different populations at the same time (Whitley et al., 1998). This produces in general better solutions than classical meta-heuristics (Cantu-Paz, 1998) within less algorithm iterations (Alba and Troya, 1999).

The good performance of parallel meta-heuristics in hard optimization tasks has motivated the creation of a new proposal for addressing UA-FLPs. Specifically, in this paper, we propose and evaluate two novel Island Model based on Coral Reefs Optimization (IMCRO) algorithms for solving the UA-FLP. These algorithms combine the idea of several independent populations with migrating movements (Island Model) with the exploration and exploitation strategies of Coral Reefs Optimization (CRO) approaches, which are inspired by the way in which corals reproduce and survive in real-world reefs (Salcedo-Sanz et al., 2014a; Salcedo-Sanz, 2017). Both approaches have been applied separately for solving UA-FLP with positive results (Palomo-Romero et al., 2017; Garcia-Hernandez et al., 2019). To the best of our knowledge, there are not previous works that have attempted to combine both algorithms to create a new technique for the UA-FLP, and therefore the originality of this research proposal is ensured. Additionally, two different versions of the novel IMCRO approach have been proposed and implemented in this article.

The remainder of this paper has been structured in the following way: next Section presents the UA-FLP definition, and describes its associated state-of-art approaches. Section 3 details the proposed IMCRO approach and the two IMCRO versions designed in this work. Section 4 describes the experimental set of this paper, including the testing parameters, experimental methodology, the results obtained and a discussion on their importance. Section 5 gives some final concluding remarks to close the paper.

2. The Unequal Area Facility Layout Problem: definition and background

2.1. UA-FLP Problem definition and literature review

The UA-FLP was first suggested by Armour and Buffa (1963), where it was specified by a rectangular plant of known width and height dimensions ($W \times H$). Within it, a set of rectangular installations (**facilities**) are arranged in the most possible efficient distribution, **while satisfying certain restrictions/constraints**: first the overlapping of facilities is not allowed. Second, these facilities must be distributed within the limits of the industrial plant, in such a way that the sum of their n associated areas A_i is equal or less than the total area plant (Equation (1)):

$$\sum_i^n A_i \leq W \times H \quad (1)$$

The UA-FLP has been tackled before in a large number of research works. Among them, we can highlight some Deterministic Methods (Montreuil, 1991; Meller et al., 1998; Sherali et al., 2003; Castillo et al., 2005; Norman and Smith, 2006; Saraswat et al., 2015; Purnomo and Wiwoho, 2016) that have **optimally** solved the problem with up to 12 facilities (Chae and Regan, 2016). However, meta-heuristic proposals (Ramadas and Abraham, 2019) have been much more used to solve the UA-FLP, especially in problems with a large number of facilities, where deterministic approaches cannot be applied due to their excessive computational complexity. Examples of these meta-heuristics are Tabu Search (Scholz et al., 2009; Kulturel-Konak, 2012), Simulated Annealing (Tam, 1992), Genetic Algorithms (Tate and Smith, 1995; Azadivar and Wang, 2000; Wu and Appleton, 2002; Gomez et al., 2003; Enea et al., 2005; Aiello et al., 2006; Liu et al., 2005; Garcia-Hernandez et al., 2013b; García-Hernández et al., 2015; Palomo-Romero et al., 2017), Harmony Search (Kang and Chae, 2017), Ant Colonies (Komarudin and Wong, 2010; Wong and Komarudin, 2010; Kulturel-Konak and Konak, 2011; Liu and Liu, 2019), Coral Reefs Optimization (Garcia-Hernandez et al., 2019) and also other alternative meta-heuristics (Ulutas and Kulturel-Konak, 2012; Gonçalves and Resende, 2015; Sikaroudi and Shahanaghi, 2016; Paes et al., 2017).

Traditionally, there are three possible structures in order to represent an industrial plant within UA-FLP algorithms. The first one is the BL, which has been employed by some early approaches to the problem, such as Meller and Gau (1996); Castillo et al. (2005); Gonçalves and Resende (2015). The second structure that is referring to STS has been more frequently applied to UA-FLP problems than the previous one: (Shayan and Chittilappilly, 2004; Scholz et al., 2009; Azadivar and Wang, 2000; Komarudin and Wong, 2010; Kang and Chae, 2017). This layout representation has as a main disadvantage: it is the most complex structure among the three discussed here, and therefore it usually needs a large amount of computation time to obtain a layout result. Finally, the FBS is the last layout representation usually considered for UA-FLP problems. It has been the most used for addressing UA-FLPs ((Shayan and Chittilappilly, 2004; Scholz et al., 2009; Azadivar and Wang, 2000; Komarudin and Wong, 2010; Kang and Chae, 2017), among others) because it

is simple and easy to implement, and it requires less resources for solving the UA-FLP than its counterpart alternatives. An example of FBS layout representation is given in Figure 1. In this example, the plant area has been divided into six rectangles (bays) and facilities have been inserted in each one of them in consecutive order: for the first bay the facilities I and C, for the second one facilities J, G and L, etc.

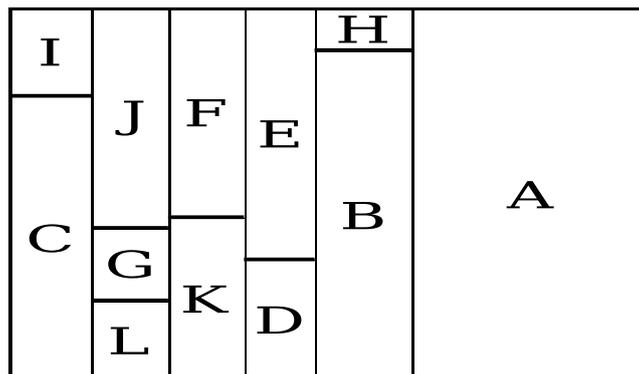


Figure 1: An example of a Flexible Bays Structure solution for the UA-FLP.

Regarding the objective function for UA-FLPs, note that there are many different criteria that can be considered to optimize the industrial plant layout. These criteria can be both qualitative and/or quantitative type. In the first group, some related works (Garcia-Hernandez et al., 2013b,a; Garcia-Hernandez et al., 2015) have taken into account qualitative preferences of the Decision Maker as the main consideration by means of interactive approaches. According to these works, some examples of qualitative optimization goals are specific placement of certain facilities, location and shape of empty spaces, or other personal preferences that the designer may have. Later, they also incorporated the expert knowledge into the UA-FLP using an evolutionary neural system.

Considering quantitative aspects, the vast majority of published works focus on minimizing the flow of materials as the main objective to be optimized ((Kulturel-Konak and Konak, 2011; Kulturel-Konak, 2012; Gonçalves and Resende, 2015; Kang and Chae, 2017) to name some of them). This value is equal to the sum of costs associated with the movement of material between each pair of facilities i and j in the plant (f_{ij}) multiplied by the distance between those facilities' centers (d_{ij}). There are also other works that consider a multi-objective optimization approach (Abraham and Jain, 2005) to solve this problem (Ripon et al., 2013; Saraswat et al., 2015; Purnomo and Wiwoho, 2016; Sherali et al., 2003), by taking into account aspects such as adjacency requirements, distance requests, etc. (Aiello et al., 2006; Gomez et al., 2003). These objectives are often referred to the closeness or remoteness between facilities, respectively. For example, it might be necessary to have the administration offices close to the plant reception.

Another aspect must be taken into account when designing the plants: a shape constraint for each facility, which may be its aspect ratio (quotient between the largest and minimum side measure) or just its minimum side length. This refers to the proportion between the sides of a facility, which must be reasonable, otherwise, too stretched facilities would appear in the design, leading to non-feasible plant layouts. For handling this issue, a penalty function is normally employed. The most frequently used was proposed by Tate and Smith (1995) and depends on the existence of facilities that do not satisfy the aspect ratio or minimum side constraint, and also, on the quantity of them that are left in the industrial plant. So, in the end, the objective function for optimizing the material flow in the UA-FLP, that also considers aspect ratio constraints, is:

$$OF_z = \sum_i^n \sum_j^n f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all}) \quad (2)$$

where z is associated with a certain facility design, n is the number of departments to be allocated in the industrial plant layout, f_{ij} is the material flow between departments i and j , d_{ij} is the distance measure between them (it can be Manhattan or Rectilinear), D_{inf} is the quantity of unfeasible departments, k is a value that allows adjusting the penalization grade (established to 3 according to Tate and Smith (1995)), V_{feas} is the minimum Objective Function (OF) value from all feasible designs (without unfeasible departments on it) which have not been penalized and, and V_{all} is best OF value achieved overall.

2.2. Parallel Evolutionary Algorithms

Given the increasing complexity of some UA-FLPs, a new way to solve these problems is needed, since the amount of time and computational capabilities necessary for reaching an acceptable solution becomes prohibitive even with the application of meta-heuristic search. In this scenario parallelization seems a good way to address large UA-FLP instances. In fact, there have been previous parallel approaches to the problem, specifically Palomo-Romero et al. (2017) using an Island Model Genetic algorithm for solving UA-FLPs, which obtained good results in comparison with other non-parallel meta-heuristic approaches.

According to Cantu-Paz (1998) parallel evolutionary algorithms can be classified into three groups, depending on the treatment given to the total population and the relationships established between the computation units:

- **Global single-population master-slave:** For a single population there are several processes to compute the individuals' fitness.
- **Single-population fine-grained:** The entire population is divided into small regions, that only relate to other neighbor regions for crossover.
- **Multiple-population coarse-grained:** In this type of algorithm, there are several populations that evolve independently from each other, and the only interaction between them consists in an exchange of individuals (migration) performed with a certain frequency.

In this paper, we propose an Island Model with CRO characteristics, which belongs to the last family of parallel algorithms: those that can improve the population diversity thanks to their imposed independence.

3. An island model meta-heuristic based on the Coral Reefs Optimization (IM-CRO) algorithm

This section presents the proposed IMCRO, detailing its characteristics, different components and algorithm versions. We first briefly introduce the CRO algorithm, and show how an island model can be constructed over it for the UA-FLP. Then, an extended version of the proposed approach, which includes different search procedures in each island, is described.

3.1. Basic CRO

The CRO algorithm Salcedo-Sanz et al. (2014b) is an evolutionary-type algorithm in which it is defined a surface Λ of size $M \times N$ where different corals X_k (potential solutions to the problem at hand) can settle down. Initially, the reef is not completely occupied, but there are holes in the reef. During the evolutionary process, the corals in the reef will reproduce in two different ways: sexually and asexually. There can be two types of sexual reproduction: external (broadcast spawning), where two corals exchange genetic information to create a new individual (larva), or internal (brooding), where the resulting larva is created with a small alteration of a single parent coral. In turn, asexual reproduction (or budding) is performed by the reef's best individuals, which produce larvae by means of self-duplication. The solutions produced by the reproduction process will try to find a place in the reef to settle in and survive for the following generations. A larva (new solution to the problem) can settle down in the reef if the spot it has selected for settling is empty, or if the current coral in the reef has worse health (fitness function) than the trying larva. If after three attempts a larva has not found a place to prosper it is discarded. At the end of each generation some of the worst corals can be also depredated, forming new holes in the reef. Figure 2 illustrates the evolution process for a generation of the reef.

3.2. Corals encoding for the UA-FLP

According to Gomez et al. (2003), a valid chromosome (coral in this case) to represent a particular UA-FLP solution with FBS consists of two different parts, or fragments: the first one refers to the order in which the n facilities appear in the layout and the second one marks the end of each bay. The order in which the departments appear in the first chromosome fragment corresponds to a reading of the departments in the layout, from top to bottom and left to right. The second fragment is a binary-encoded vector where a value '1' in a position i represents that the facility placed in the position i in the first fragment is the last one in the bay. This way, the chromosome below is associated with Figure 1.

Chromosome structure:

1. **Fragment 1:** {I, C, J, G, L, F, K, E, D, H, B, A}
2. **Fragment 2:** {0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1}

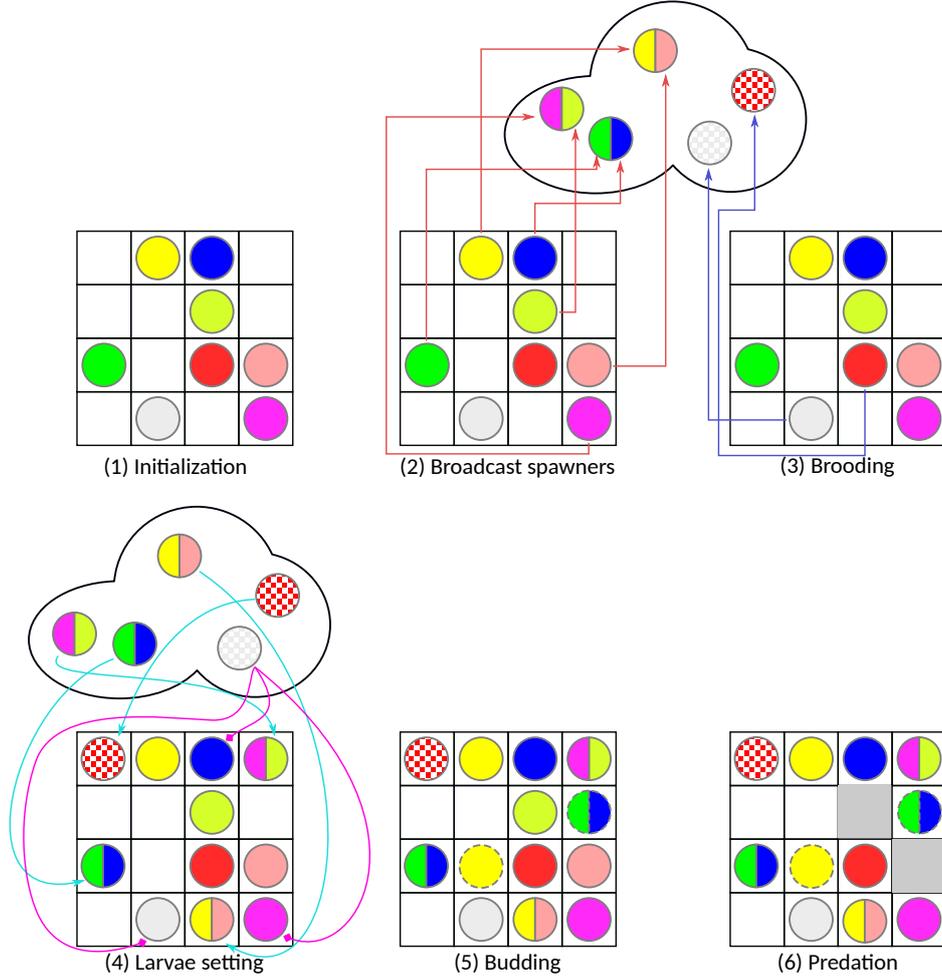


Figure 2: Evolution process in the CRO algorithm.

3.3. The Island Model CRO

The **IMCRO** algorithm can be described as a multiple population version of the basic CRO. Let $K = \{\Lambda_0, \Lambda_1, \dots, \Lambda_{r-1}\}$ be an array of r independent reefs of dimensions $\{M_0 \times N_0, M_1 \times N_1, \dots, M_{r-1} \times N_{r-1}\}$. Each reef Λ evolves as a basic CRO, and during every *migr_freq* generation the best *num_migr* corals from each reef are redistributed between the rest of the reefs, guaranteeing that no coral ends up in its original reef.

The migration scheme followed to corals rearrangement is somewhat like networks' mesh topology: each reef is "connected" to every other reef in the sense that a coral X_k^r can move to any other reef $s \neq r; s \in K$, but corals' movements are independent from each other. Figure 3 shows an example of migration for a 5 reefs system with 4 migrating corals. Note that each coral from a certain reef may (or may not) have different destinations.

In order to better explain the IMCRO proposal, the algorithm's pseudocode is shown below, and it is fully explained step by step in the next enumeration.

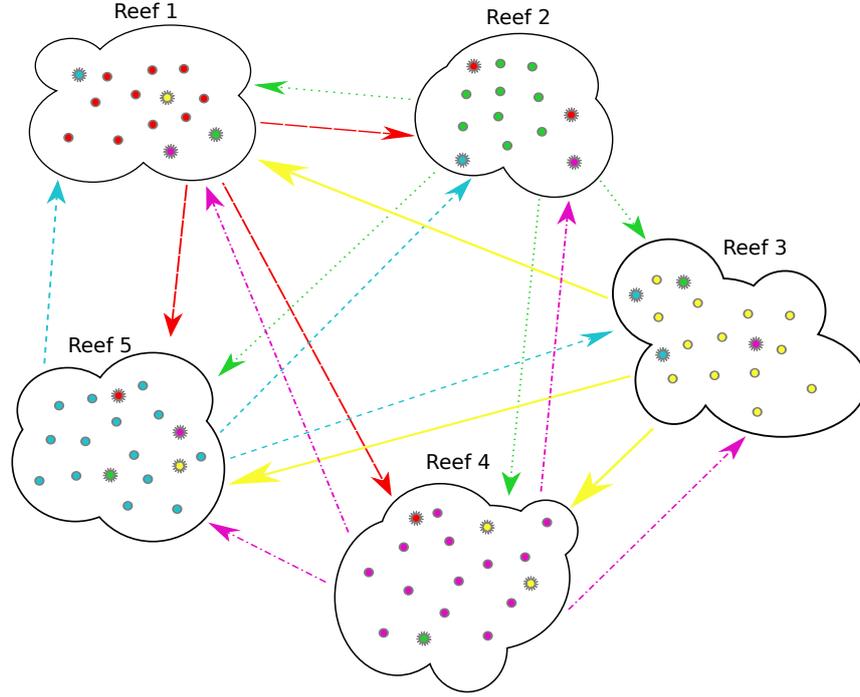


Figure 3: Graphical diagram of the migration process of larvae among reefs in the IMCRO algorithm.

Algorithm 1 IMCRO algorithm

Input Algorithm's control parameters

Output Feasible solution with best *fitness*

```

1: procedure IMCRO(num_reefs, m, n,  $\rho_0$ ,  $f_b$ ,  $f_a$ ,  $f_d$ ,  $p_d$ , migr_freq, num_migr)
2:    $r \leftarrow 0$ 
3:   for num_reefs times do
4:     initialize reef r with size  $m_r \times n_r$  and occupation rate  $\rho_0^r$ 
5:      $r \leftarrow r + 1$ 
6:   end for
7:   num_it  $\leftarrow 0$ 
8:   repeat
9:     for each reef r do
10:      reproduce corals fraction  $f_b^r$  by broadcast spawning
11:      reproduce corals fraction  $1 - f_b^r$  by brooding
12:      larvae evaluation
13:      larvae setting
14:      reproduce best corals fraction  $f_a^r$  by asexual reproduction
15:      depredation of  $f_d^r$  worst reef corals with  $p_d^r$  probability
16:    end for
17:    num_it  $\leftarrow num\_it + 1$ 
18:    if num_it = migr_freq then
19:      migrate top num_migr corals from each reef
20:      num_it  $\leftarrow 0$ 
21:    end if
22:  until stop condition
23:  return best feasible solution
24: end procedure

```

1. **Initialization:** Each reef r is randomly filled with randomly generated corals until an occupation fraction ρ_0^r of the total surface is reached.
2. **For each reef:**
 - (a) **Broadcast spawning:** A fraction f_b^r of the reef corals are coupled randomly, avoiding multiple parenting (a coral can reproduce once per generation). Each couple produces a larva by means of genetic operators.
 - (b) **Brooding:** from the remaining corals $(1 - f_b^r)$ several larvae are generated by a small alteration applied to each one of them.
 - (c) **Larvae setting:** All the larvae previously created try to find a place in the reef to settle following the next procedure:
 - i. Select a random spot.
 - ii. **If** the spot is **empty**: settle down OR
 - iii. **If** the larva has **better health** than the coral occupying the spot: substitute coral.
 - iv. **Otherwise** go to step 2.(c).i.

These steps can be repeated up to three times. After that the larva is discarded.
 - (d) **Asexual reproduction:** A fraction f_a^r of the healthier reef's corals duplicate themselves, mutate and try to settle down as in the previous step (**Larvae Setting**). This phase is also called *budding*.
 - (e) **Predation:** The worst f_d^r fraction of corals can be predated with a low probability p_d^r .
3. **Migrate** best corals: Every $migr_freq$ generations the migration process is performed, where the best num_migr corals are displaced to another reef.
4. Go back to step 2 until a stop condition is met. In the present implementation the algorithm stops after a prefixed number of generations.

3.4. The Extended Island Model CRO

The approach previously described, however, may produce one unintended effect: if every reef explores the search space in the same way (that is, using the same crossover and mutation operators), there is a high probability that all the reefs converge to the same search space, instead of exploiting different zones of that space, as intended. In order to prevent this, a possible solution would be the implementation of different evolution strategies per reef (combination of crossover and mutation operators). This way, broadcast spawning and brooding operators implementation depends on the reef from which the parents are picked up. This idea has been previously implemented in an advanced version of the CRO algorithm, the so-called CRO-SL approach (Coral Reefs Optimization with Substrate Layer), which is a multi-method approach in which different search operators are jointly applied within a single population (Salcedo-Sanz et al., 2016; Camacho-Gómez et al., 2018; Bermejo et al.,

2018; Salcedo-Sanz, 2017). In this case, the proposed extended island model brings this idea and extends it to parallelize the algorithm, by including each search operator into a different island.

The operators implemented for our approach are enumerated and briefly described in the following subsections.

3.4.1. Crossover operators

This section describes the crossover operators considered in the Extended IMCRO algorithm, both for facility order (first chromosome fragment) and bay structure (second fragment).

- Crossover operators for facility order:
 1. Partially Mapped Crossover (**PMX**) (Goldberg and Lingle, 1985). The crossover operates following the next steps:
 - (a) Select two cut points randomly.
 - (b) Copy the segment defined by the cut points from each parent and create an equivalence map for recombination (the i -th value from a parent corresponds to the other parent's i -th value). This map is used to ensure that both children represent valid permutations.
 - (c) Copy the remaining elements (not present in the segment) from the other parent using the equivalence map for those elements which are already present because they have been copied in the previous step.
 2. Order Crossover (**OX**) (Eiben and Smith, 2003). The recombination performed by this operator is similar to PMX, but no equivalence map is used. Instead, the elements outside the cut points appear in the same order as in the other parent starting from the second cut point.
 3. **Cycle** (Eiben and Smith, 2003). The way to combine parents is the following:
 - (a) Find cycles aligning parent chromosomes and pairing values in a given position.
 - (b) Select one of the found cycles (if any).
 - (c) Create children maintaining the elements from the found cycle from one parent and filling the remaining gaps with the elements from the other parent.
 4. **Edge** (Eiben and Smith, 2003). This operator is based on the idea that the child obtained must be very different from both parents. In order to accomplish this, an edge table is constructed which stores what linkages each element has in both parents. After that a child is built following this procedure:
 - (a) Pick element *elem* from the table randomly and add to the offspring.
 - (b) Remove all *elem* references from the table.
 - (c) Examine table content for the entry corresponding to *elem* and select, in order of preference:

- i. Common edge.
 - ii. Table entry with less elements. In case of tie, select randomly from that set.
 - iii. In case of empty list of edges, the next entry to be examined will be the one corresponding to the current last element of the offspring, otherwise random selection is applied.
 - (d) Update *elem* value according to the value selected in the previous step.
 - (e) Go back to (b) until offspring is complete.
- Crossover operators for bay structure:
 1. **One-point** (Holland, 1992). A cut point is randomly selected, and both segments created after that are combined alternatively to create two children.
 2. **N-point** (Eiben and Smith, 2003). This operator is a generalization from the previous one. The number of cut points is chosen randomly and children are created alternating segment from both parents.
 3. **Uniform** (Eiben and Smith, 2003). Every allele is randomly selected from one or another parent with the same probability.

3.4.2. Mutation operators

Mutation operators considered in the Extended IMCRO are presented below.

- Mutation operators for facility order:
 1. Towers (**TWORS**) (Eiben and Smith, 2003). Two positions are picked at random and its content is exchanged.
 2. **Inversion** (Eiben and Smith, 2003). As in the previous operator, two positions are randomly selected. After that, the section delimited by both positions is inverted.
 3. **Scramble** (Eiben and Smith, 2003). The elements contained in a section defined by two random positions are scrambled.
 4. **Insert** (Eiben and Smith, 2003). Two elements are selected at random. The second one is positioned after the first, and all elements after the first one are moved to make room.
- Mutation operators for bay structure:
 1. **Bit-swap** (Eiben and Smith, 2003). For each allele, with a certain probability, changes its value (from 0 to 1 or vice versa).

The algorithm proceeds just like the basic version previously explained (Algorithm 1), but using different operators in the Broadcast spawning and Brooding phases, depending on the reef (island) in which [these](#) operations are performed.

4. Computational experiments and results

In this section we evaluate the performance of the proposed IMCRO approach in the two suggested versions: Basic IMCRO and Extended IMCRO. For that matter, a common set of well-known UA-FLPs has been selected in order to analyse the results reached by our proposal. Then, a comparison with alternative approaches and a full discussion on the results obtained has been carried out.

4.1. Description of UA-FLPs test instances

A large set of well-known UA-FLPs has been chosen considering different problem sizes (small, medium and large). The description related to each UA-FLP is expressed in Table 1. The following information is detailed: problem name, number of facilities, layout dimensions (in squared meters), shape constraint (let it be aspect ratio α or minimum side measure *side*), distance measure and problem reference.

Table 1: Characteristics of the tested UA-FLPs.

<i>Problem name</i>	<i>Fac.</i>	$W \times H$ (m^2)	<i>Shape constr.</i>	<i>Dist.</i>	<i>Reference</i>
Slaughterhouse	12	51.14×30.00	$\alpha=4$	Eucl.	Salas-Morera et al. (1996)
CartonPacks	11	20.00×14.50	$\alpha=4$	Eucl.	Garcia-Hernandez et al. (2013a)
ChoppedPlastic	10	10.00×30.00	$\alpha=4$	Eucl.	Garcia-Hernandez et al. (2013a)
O7	7	8.54×13.00	$\alpha=4$	Rect.	Meller et al. (1998)
O8	8	11.3×13.00	$\alpha=4$	Rect.	Meller et al. (1998)
O9	9	12.00×13.00	$\alpha=5$	Rect.	Wong and Komarudin (2010)
vC10Ra	10	25.00×51.00	$\alpha=5$	Rect.	van Camp et al. (1992)
Vc10Rs	10	25.00×51.00	<i>side</i> =5	Rect.	van Camp et al. (1992)
vC10Ea	10	25.00×51.00	$\alpha=5$	Eucl.	van Camp et al. (1992)
Vc10Es	10	25.00×51.00	<i>side</i> =5	Eucl.	van Camp et al. (1992)
Ba12	12	6.00×10.00	<i>side</i> =1	Rect.	Bazaraa (1975)
MB12	12	6.00×8.00	$\alpha=4$	Rect.	Bozer and Meller (1997)
Ba14	14	7.00×9.00	<i>side</i> ={1,0}	Rect.	Komarudin and Wong (2010)
AB20_ar3	20	2.00×3.00	$\alpha=3$	Rect.	Armour and Buffa (1963)
AB20_ar5	20	2.00×3.00	$\alpha=5$	Rect.	Armour and Buffa (1963)
AB20_ar7	20	2.00×3.00	$\alpha=7$	Rect.	Armour and Buffa (1963)
AB20_ar10	20	2.00×3.00	$\alpha=10$	Rect.	Armour and Buffa (1963)
AB20_ar15	20	2.00×3.00	$\alpha=15$	Rect.	Armour and Buffa (1963)
AB20_ar50	20	2.00×3.00	$\alpha=50$	Rect.	Armour and Buffa (1963)
SC30	30	12.00×15.00	$\alpha=5$	Rect.	Liu and Meller (2007)
SC35	35	16.00×15.00	$\alpha=4$	Rect.	Liu and Meller (2007)
Du62	62	Arbitrary	$\alpha=4$	Rect.	Dunker et al. (2003)

4.2. Experimental methodology

In order to test the proposed IMCRO approach considering the two proposed versions (Basic and Extended), a full empirical test has been performed in order to find those tuning parameters that best fit when solving the UA-FLP, i.e., what combination of parameters yield the best solutions for a certain problem. Table 2 shows the possible values that have considered for each algorithm parameter. Not every possible value has been tested, of course, due to the unattainable number of possible combinations but a set of the most recommended ones. For example, predation fraction F_d and predation probability P_d should not take high values, as that would lead to an excessively elitist behavior, so values above 0.1 have not been considered. Parameter tuning has been performed by selecting one of the three possible values considered for each parameter and using that configuration for a representative problem of a certain size (O9 for the smaller instances, BA14 for the medium-sized ones and Du62 for large instances). For small problems, the number of generations carried out by the algorithm has been 1000, while for medium and large instances this parameter has been set at 5000. The algorithm can also stop before reaching that number of generations if after 500 iterations no better solution has been found. The number of generations has been set that high because during the search we do not want to leave potential good areas of the search space unexplored, but the second stop condition will ensure that if the algorithm converges no computation time is wasted. Each one of these parameter combinations has been cloned 5 times as in the Komarudin and Wong (2010) proposal. The parameters that best adapt to each UA-FLP category are shown in Table 3, where S stands for small problems (up to 12 facilities), M for medium (13 to 25) and L for large (26 or more). In the basic IMCRO algorithm, we have used PMX, One-point crossover, TWORS mutation and bit-swap as search operators. In turn, for the Extended IMCRO we have defined five combinations of crossover and mutation operators:

- a) **Crossover:** PMX, 1-point | **Mutation:** TWORS, Bit-swap.
- b) **Crossover:** Cycle, N-point | **Mutation:** Inversion, Bit-swap.
- c) **Crossover:** OX, Uniform | **Mutation:** Scramble, Bit-swap.
- d) **Crossover:** PMX, Uniform | **Mutation:** Insert, Bit-swap.
- e) **Crossover:** Edge, N-point | **Mutation:** Scramble, Bit-swap.

Our IMCRO proposal has been developed using Python 3. Since a time comparison with previous basic CRO approaches have been carried out, it is necessary to specify that the test experimentation has been executed using an Intel Core i5 6200U (2.30 GHz \times 4), 8GB RAM and a Linux operating system.

4.3. Results

This subsection presents the results obtained by both versions (Basic IMCRO and Extended IMCRO) of the proposed IMCRO approach. Then a comparison and discussion

Table 2: Tested values for parameter tuning.

Parameter	$N \times M$	r	ρ_0	F_b	F_a	F_d	P_d	<i>migr_freq</i>	<i>num_migr</i>
	10×10	5	0.6	0.7	0.1	0.01	0.01	5	5
Tested values	15×15	10	0.7	0.8	0.15	0.05	0.05	10	10
	25×25	20	0.8	0.9	0.2	0.1	0.1	20	20

Table 3: Selected parameter values according to problem size.

Parameter	$N \times M$	r	ρ_0	F_b	F_a	F_d	P_d	<i>migr_freq</i>	<i>num_migr</i>
Selected values (S)	10×10	5	0.7	0.8	0.1	0.1	0.1	5	5
Selected values (M)	15×15	10	0.7	0.8	0.2	0.1	0.1	5	10
Selected values (L)	25×25	10	0.7	0.8	0.2	0.1	0.1	5	10

is carried out using alternative previous approaches reported in the literature. Initially, a summary of the best results (OF Best) and their associated mean (OF mean) reached by the two versions of IMCRO proposal for each of the tested UA-FLPs are listed in Table 4. Moreover, this table gives the difference (in percent) between the best solutions obtained by the two IMCRO versions ($100 \times (\text{the best known solution found in the literature} - \text{the novel IMCRO best solution}) / (\text{the novel IMCRO best solution})$). In addition, this table is completed offering the time employed by our IMCRO proposal for reaching the best result in each implemented version, and also, the CPU time needed in the Basic CRO proposal (Garcia-Hernandez et al., 2019). Looking at Table 4, it is possible to confirm that the two developed versions of the IMCRO approaches are robust, since the difference between the best solution’s fitness and the average fitness is very small. This difference becomes larger as the number of departments in the problem increases. Regarding the time units needed to achieve the best solutions for each problem, the IMCRO approach is more efficient in terms of CPU time, since it outperforms the basic CRO approach in all cases tested. However, CPU time is not considered as something crucial in the UA-FLP (See and Wong, 2008). Finally, Tables 7 and 8 show the performance of Basic and Extended versions of the IMCRO approach when solving UA-FLPs. From this analysis, we can state that the Extended version reaches better results than the Basic one, since it is able to [match the results obtained by its basic counterpart in 18 of all the tested instances and overcome them in 4 of them \(Slaughterhouse, Carton Packs, AB20_AR5 and AB20_AR7\)](#). The solutions achieved for the previous problems by the Extended IMCRO are illustrated in Figure 4.

Table 4: Results obtained by the Basic and the Extended IMCRO versions.

<i>UA-FLP</i>	<i>B_OF Best</i>	<i>B_OF Mean</i>	<i>E_OF Best</i>	<i>E_OF Mean</i>	<i>Diff(%)</i>	<i>B_CPU(s)</i>	<i>E_CPU(s)</i>	<i>CRO CPU(s)</i>
Slaughterhouse	3440.49	3490.49	3439.96	3487.11	0.02	72.00	74.00	75.00
Carton Packs	86.08	86.08	85.99	86.03	0.1	69.00	73.00	74.00
Chopped Plastic	257.94	257.94	257.94	257.94	0.00	64.00	63.00	65.00
O7	134.16	134.16	134.16	134.16	0.00	4.00	4.00	4.00
O8	245.48	245.48	245.48	245.48	0.00	22.00	22.00	24.00
O9	238.73	238.73	238.73	238.73	0.00	35.00	39.00	39.00
vC10Ra	20142.13	20142.13	20142.13	20142.13	0.00	60.00	59.00	61.00
Vc10Rs	22897.65	22897.65	22897.65	22897.65	0.00	61.00	60.00	63.00
vC10Ea	18554.59	18554.59	18554.59	18554.59	0.00	86.00	84.00	-
Vc10Es	18823.74	18823.74	18823.74	18823.74	0.00	82.00	81.00	-
Ba12	8021.0	8021.96	8021.0	8099.96	0.00	86.00	85.00	87.00
MB12	125.00	125.00	125.00	125.00	0.00	77.00	80.00	81.00
Ba14	4649.22	4740.23	4649.22	4739.15	0.00	89.00	90.00	92.00
AB20_ar3	5396.37	5455.11	5396.37	5475.77	0.00	131.00	137	-
AB20_ar5	5252.98	5371.04	5252.98	5360.55	0.00	182.00	198	202
AB20_ar7	4789.24	4881.19	4785.96	4802.24	0.07	160.00	145	-
AB20_ar10	4367.56	4425.32	4367.56	4403.21	0.00	147.00	160	-
AB20_ar15	4099.63	4202.67	4099.38	4152.33	0.01	143.00	137	-
AB20_ar50	2382.73	2419.90	2382.73	2423.84	0.00	136.00	135	-
SC30	3714.74	4002.54	3714.74	3999.13	0.00	617.00	611.00	622.00
SC35	4263.3	4899.35	4263.3	4714.14	0.00	545.00	550.00	552.00
Du62	3654022.27	3674086.53	3646062.45	3667124.28	0.22	861.00	860	871.00

In order to show the reached results are used Tables 5 and 7 for the Basic version of IMCRO, and 6 and 8 for the Extended IMCRO version. On them, it is only shown a comparison on them with the other proposals that have also used FBS as layout representation. Particularly, results obtained by the following works are displayed: Garcia-Hernandez et al. (2019), Palomo-Romero et al. (2017), Kulturel-Konak and Konak (2011), Kulturel-Konak (2012), Wong and Komarudin (2010) and Enea et al. (2005). For each UA-FLP, the best solution value has been highlighted in bold font. Taking a look at the Tables 5 and 6, it can be seen that our proposal is capable of reaching the best solution in most of the cases tested. In particular, the both versions of our novel proposal are able to find the best result in 19 UA-FLPs out of the 22 tested. In the other three UA-FLPs, the both versions of the IMCRO system are able to obtain results very close to the best solutions obtained by alternative approaches. Both Basic and Extended versions of the IMCRO approach match the best known in 10 of the tested problems. These are the cases of the problems Slaughterhouse, Carton Packs, Chopped Plastic, O9, Ba14, AB20_AR3, AB20_AR5, AB20_AR7, AB20_AR15 and AB20_AR50. For better illustration of their associated facility designs, we have included the physical layout representation of all of these UA-FLPs where our IMCRO approach beats the best known solutions (Figures 4, 5 and 6). In all the solutions showed, the number of infeasible departments is zero, so the value of OF is the same as the flow term, which is the minimum found accomplishing the aspect ratio restrictions. Moreover, both Basic and Extended versions of the IMCRO are able to reach the best known solutions in the following UA-FLPs: O7, O8, Vc10Ra, Vc10Rs, Vc10Ea, Vc10Es, Ba12, MB12, and AB20_AR10.

Comparing both versions of the IMCRO approach with the basic CRO (Garcia-Hernandez et al., 2019), it is possible to state that the new algorithm based on an Island Model performs better in the UA-FLP, because of its ability to improve the basic CRO approach in 7 UA-FLP cases and also reaching the best UA-FLP solution in other 5 instances. Focusing on Palomo-Romero et al. (2017) approach, the IMCRO in its two versions (Basic and Extended) is able to improve their solutions in 9 UA-FLPs and reaching their best solutions in other 7 cases. Comparing our proposal against Kulturel-Konak and Konak (2011), the two developed versions of the novel IMCRO approach exceeded their best solutions in 8 problems and it is able to achieve the best known solution in other 2. The same occurs when we compare the IMCRO with the approach in Kulturel-Konak (2012). Additionally, comparing the IMCRO strategy in the two defined versions with the (Wong and Komarudin, 2010) research, our approach surpasses their results in all the UA-FLPs, reaching in one of them their best solution. Finally, our IMCRO approach beats Enea et al. (2005)'s approach in all the tested cases.

Table 7 and 8 add additional relevant information by comparing the solutions achieved by the Basic and Extended version of the IMCRO and the best-known solutions reached by previous approaches, respectively. Thus, for each UA-FLP tested, the best result achieved for the each version of the proposed IMCRO is displayed, and also their associated layout structure, and the percentage of difference between the best result value obtained by our approach and the best solution found in previous works in the literature.

Table 5: Summary of the solutions achieved by the suggested IMCRO in basic version and other state of the art works.

<i>UA-FLP</i>	<i>FBS approaches</i>						
	<i>B_IMCRO</i>	<i>Garcia (2019)</i>	<i>Palomo (2017)</i>	<i>Kulturel (2011)</i>	<i>Kulturel (2012)</i>	<i>Wong (2010)</i>	<i>Enea (2005)</i>
Slaughterhouse	3440.49	3487.12	-	-	-	-	3854.00
CartonPacks	86.08	89.02	-	-	-	-	94.10
ChoppedPlastic	257.94	265.77	-	-	-	-	377.18
O7	134.16	134.16	134.19	-	-	-	-
O8	245.48	245.48	245.51	-	-	-	-
O9	238.73	239.44	241.06	-	-	-	241.06
Vc10Ra	20142.13	20142.13	20142.13	20142.13	21463.07	21463.1	-
Vc10Rs	22899.65	22899.65	22899.65	22899.65	22899.65	22899.65	-
Vc10Ea	18554.59	-	18554.59	-	-	-	-
Vc10Es	18823.74	-	18823.74	-	20320.52	-	-
Ba12	8021.00	8021.00	8435.83	8129.00	8021.0	8786.00	-
MB12	125.00	125.00	125.00	-	-	-	-
Ba14	4649.22	4665.93	4665.93	4780.91	4739.74	5004.55	-
AB20_ar3	5396.37	-	5419.49	5594.30	5594.29	-	-
AB20_ar5	5252.98	5256.95	5256.10	5336.36	5297.6	5677.83	-
AB20_ar7	4789.24	-	4844.49	4793.5	4793.47	-	-
AB20_ar10	4367.56	-	4367.56	4440.70	4440.68	-	-
AB20_ar15	4099.63	-	4100.17	4119.8	4119.8	-	-
AB20_ar50	2382.73	-	2382.74	3526.5	2457.79	-	-
SC30	3714.74	3519.44	3613.11	3443.34	3563.95	-	-
SC35	4272.21	4263.30	3885.29	3700.75	-	-	-
Du62	3654022.27	3713876.55	-	3641497.00	-	-	-

Table 6: Summary of the solutions achieved by the suggested IMCRO in extended version and other state of the art works.

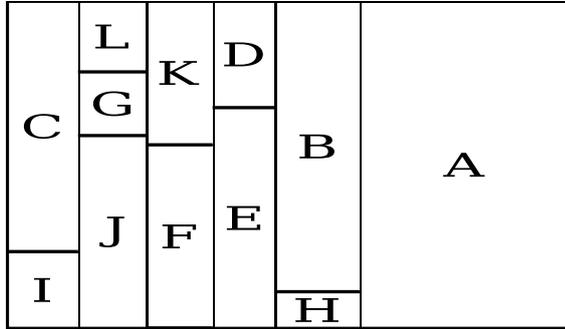
<i>UA-FLP</i>	<i>FBS approaches</i>						
	<i>E_IMCRO</i>	<i>Garcia (2019)</i>	<i>Palomo (2017)</i>	<i>Kulturel (2011)</i>	<i>Kulturel (2012)</i>	<i>Wong (2010)</i>	<i>Enea (2005)</i>
Slaughterhouse	3439.96	3487.12	-	-	-	-	3854.00
CartonPacks	85.99	89.02	-	-	-	-	94.10
ChoppedPlastic	257.94	265.77	-	-	-	-	377.18
O7	134.16	134.16	134.19	-	-	-	-
O8	245.48	245.48	245.51	-	-	-	-
O9	238.73	239.44	241.06	-	-	-	241.06
Vc10Ra	20142.13	20142.13	20142.13	20142.13	21463.07	21463.1	-
Vc10Rs	22899.65	22899.65	22899.65	22899.65	22899.65	22899.65	-
Vc10Ea	18554.59	-	18554.59	-	-	-	-
Vc10Es	18823.74	-	18823.74	-	20320.52	-	-
Ba12	8021.00	8021.00	8435.83	8129.00	8021.0	8786.00	-
MB12	125.00	125.00	125.00	-	-	-	-
Ba14	4649.22	4665.93	4665.93	4780.91	4739.74	5004.55	-
AB20_ar3	5396.37	-	5419.49	5594.30	5594.29	-	-
AB20_ar5	5252.98	5256.95	5256.10	5336.36	5297.6	5677.83	-
AB20_ar7	4785.96	-	4844.49	4793.5	4793.47	-	-
AB20_ar10	4367.56	-	4367.56	4440.70	4440.68	-	-
AB20_ar15	4099.38	-	4100.17	4119.8	4119.8	-	-
AB20_ar50	2382.73	-	2382.74	3526.5	2457.79	-	-
SC30	3714.74	3519.44	3613.11	3443.34	3563.95	-	-
SC35	4272.21	4263.3	3885.29	3700.75	-	-	-
Du62	3646062.45	3713876.55	-	3641497.00	-	-	-

Table 7: Results and chromosomes achieved by the IMCRO in the basic version.

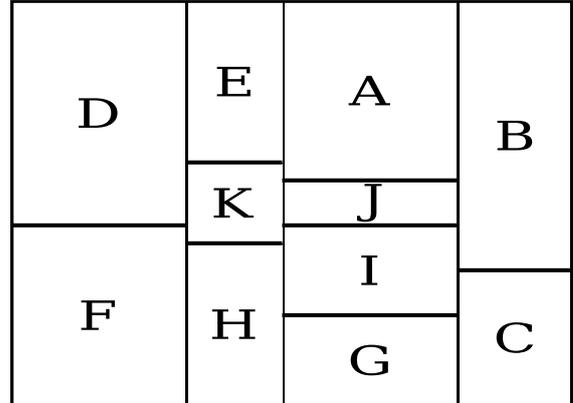
<i>UA-FLP</i>	<i>B_IMCRO</i>	<i>Diff. (%)</i>	<i>B_IMCRO chromosome</i>
Slaughterhouse	3440.49	1.35	9,3 10,7,12 6,11 5,4,13 8,2 1
CartonPacks	86.08	3.41	6,4 3,2 10,9,5,1 8,11,7
ChoppedPlastic	257.94	3.03	1,2,3,4,5,6,7 8,9,10
O7	134.16	0.00	3,5,7 1,4,6,2
O8	245.48	0.00	5,8,6,3 2,1,4,7
O9	238.73	0.29	3,1,6,9,5 4,2 7,8
Vc10Ra	20142.13	0.00	5,8,10,9,2,6,1 4,7,3
Vc10Rs	22899.65	0.00	3,5 9,10,8 2,4 6,7 1
Vc10Ea	18554.59	0.00	7,4,3 1 8,5,9,2,6,10
Vc10Es	18823.74	0.00	10,9 8,4,2 5,7,6 3,1
Ba12	8021.0	0.00	4,10 9,5,7 3 2,12 1 11,8,6
MB12	125.00	0.00	12 9,1,5,6,8,2,4,3,7,10 11
Ba14	4649.22	0.35	11,5,10 1 3 13 4 2,12 14,6,8,9,7
AB20_ar3	5396.37	0.42	18,20 6,4,2,7,8,5 1,19,3,10 12,9,14 17,13,15 16,11
AB20_ar5	5252.98	0.05	11,16 15,17 13,12 5,14,10,9,19,3 20,8,7,2,4,6 1,18
AB20_ar7	4789.24	0.08	1,20,8,7,6,5 13,15,14,10,3,19,2,4,18 9,12 17 16 11
AB20_ar10	4367.56	0.00	5,20,8,7,2,4,6,18 15,13,14,10,9,19,3,1 12 17 16 11
AB20_ar15	4099.38	0.02	11 11 16 17 12 15 13,6 5,3,14,10,9,19,4,2,7,8,20 18 1
AB20_ar50	2382.73	0.0004	11 16 13 17 12 15 9 14 10 3 19 4 2 6 7 8 20 5 18 1
SC30	3714.74	-7.3	20,30,19,26 29,27 25,3,14,24,17 18,28,23 15,11,9,8 12,7 16,14,13,10,5,6,2,21,22
SC35	4272.21	-13.37	27,21,7,8,9,11,31,16 24,2,6,5,12,32,14 23,1,25,30,10,13 19,29,17,15 28,33,18 20,34,22,4,26 3
Du62	3654022.27	-0.34	45,26,6 39,35,22,4,41 33,30,60,50,36,23,13 19,51,8,18,32,20,25,53 34,55,5,24,56,21,16,3,43,29,44,7 31,47,57,59,38,61,48 9,52,28,12,11,1,49,14 15,37,27,2,40,42,10 46, 58, 54 17, 62

Table 8: Results and chromosomes achieved by our IMCRO in the extended version.

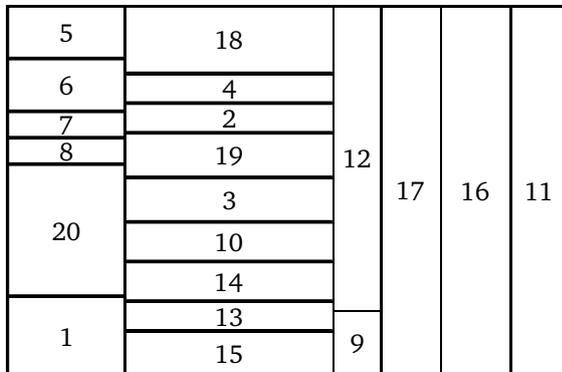
<i>UA-FLP</i>	<i>E-IMCRO</i>	<i>Diff. (%)</i>	<i>E-IMCRO chromosome</i>	
Slaughterhouse	3439.96	1.37	3,9 12,7,10 11,6,13 4,5 2,8 1	
CartonPacks	85.99	3.52	4,6 5,11,8 1,10,9,7 2,3	
ChoppedPlastic	257.94	3.03	1,2,3,4,5,6,7 8,9,10	
O7	134.16	0.00	3,5,7 1,4,6,2	
O8	245.48	0.00	5,8,6,3 2,1,4,7	
O9	238.73	0.29	3,1,6,9,5 4,2 7,8	
Vc10Ra	20142.13	0.00	5,8,10,9,2,6,1 4,7,3	
Vc10Rs	22899.65	0.00	3,5 9,10,8 2,4 6,7 1	
Vc10Ea	18554.59	0.00	7,4,3 1 8,5,9,2,6,10	
Vc10Es	18823.74	0.00	10,9 8,4,2 5,7,6 3,1	
Ba12	8021.0	0.00	4,10 9,5,7 3 2,12 1 11,8,6	
MB12	125.00	0.00	12 9,1,5,6,8,2,4,3,7,10 11	
Ba14	4649.22	0.35	11,5,10 1 3 13 4 2,12 14,6,8,9,7	
AB20_ar3	5396.37	0.42	18,20 6,4,2,7,8,5 1,19,3,10 12,9,14 17,13,15 16,11	
AB20_ar5	5252.98	0.05	11,16 15,17 13,12 5,14,10,9,19,3 20,8,7,2,4,6 1,18	
AB20_ar7	4785.96	0.15	5, 6,7,8,20,1 18,4,2,19,3,10,14,13,15 12,9 17 16 11	
AB20_ar10	4367.56	0.00	5,20,8,7,2,4,6,18 15,13,14,10,9,19,3,1 12 17 16 11	
AB20_ar15	4099.63	0.01	11 16 17 12 15 13,6 5,3,14,10,9,19,4,2,7,8,20 1 18	
AB20_ar50	2382.73	0.0004	11 16 13 17 12 15 9 14 10 3 19 4 2 6 7 8 20 5 18 1	
SC30	3714.74	-7.3	20,30,19,26 29,27 25,3,14,24,17 18,28,23 15,11,9,8 12,7 16,14,13,10,5,6,2,21,22	
SC35	4272.21	-13.37	27,21,7,8,9,11,31,16 24,2,6,5,12,32,14 23,1,25,30,10,13 19,29,17,15 28,33,18 20,34,22,4,26 3	
Du62	3646062.45	-0.12	45,39,6 26,35,22,4,41 33,30,60,36,50,23,13 19,51,8,18,32,20,25,53	
			34,55,5,24,56,21,16,3,43,29,49,7 31,47,12,57,38,61,10	
			9,52,28,59,11,1,44,14 15,37,27,2,40,42,48 46,58,54 17,62	



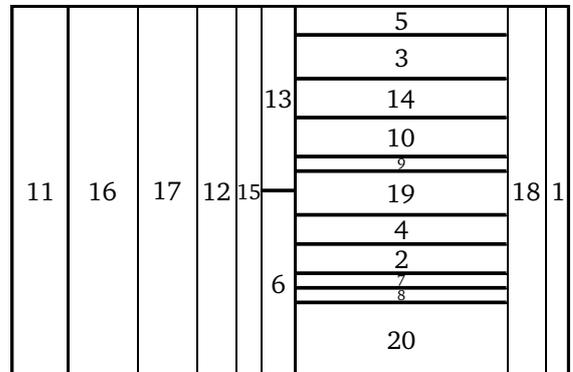
(a) Slaughterhouse



(b) CartonPacks



(c) AB20_Ar7



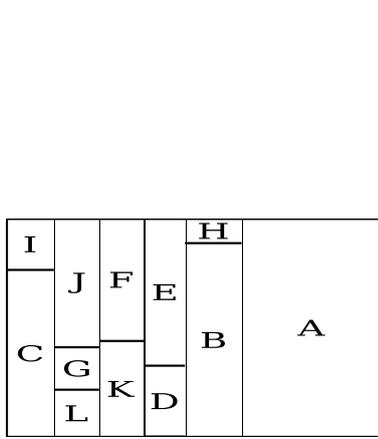
(d) AB20_AR15

Figure 4: Best solutions obtained by Extended IMCRO for some of the tested UA-FLP instances.

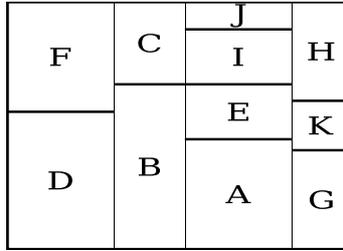
After analyzing the information shown above, we can say that the proposed IMCRO algorithm is effective when addressing UA-FLPs of different categories according to their size. So, the IMCRO algorithm is able to win or reach the best known solutions in all small UA-FLPs. These are the cases of Slaughterhouse, Carton Packs, Chopped Plastic, O7, O8, O9, Vc10Ra, Vc10Rs, Vc10Ea, Vc10Es, Ba12, MB12, Ba14. By solving medium problems as AB20_AR3, AB20_AR5, AB20_AR7, AB20_AR10, AB20_AR15, and AB20_AR50, the proposed IMCRO also demonstrates its effectiveness in achieving the best solutions. In this size category, referring to SC30, the proposed approach reaches to solutions very close to the best known results. This happens too in large UA-FLPs as SC35 and DU62, where, again, our IMCRO is able to approximate to the best known solutions for these instances.

5. Concluding remarks

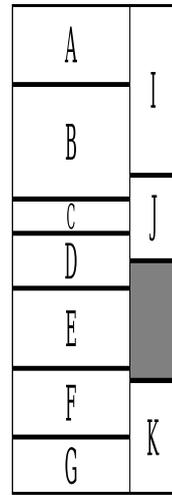
In this research, a new approach that combines an Island Model and the Coral Reefs Optimization algorithm has been proposed. This novel IMCRO system has been applied to the UA-FLP considering FBS as facility layout representation. Two different versions of the



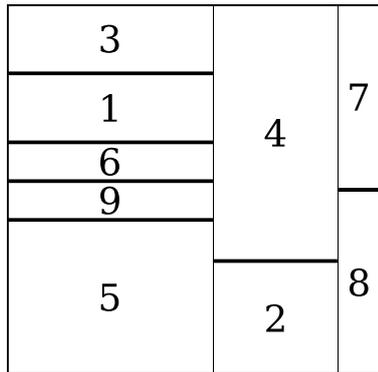
(a) Slaughterhouse



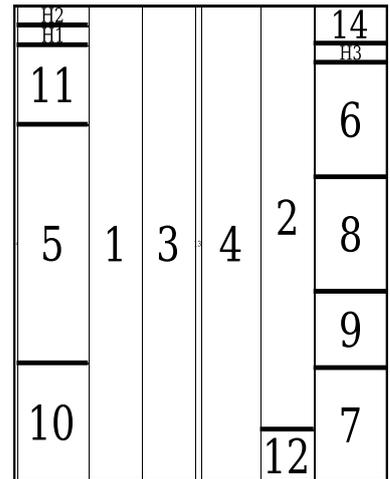
(b) CartonPacks



(c) ChoppedPlastic



(d) O9



(e) Ba14

Figure 5: Best solutions obtained by Basic IMCRO for some of the tested UA-FLP instances.

IMCRO algorithm have been proposed and tested: the Basic and Extended IMCRO versions. To illustrate the effectiveness of both versions, 22 well-known UA-FLPs instances extracted from the literature have been tested, and a full comparison with the previous related works has been carried out. From the empirical study and analysis performed, it is evident that the proposed IMCRO is a robust algorithm, able to reach the best design solutions in most of the tested problems. Specifically, the IMCRO approach reaches the best solution in 19 instances out of the 22 tested UA-FLPs. In addition, in 10 UA-FLPs, the IMCRO approach outperforms the results achieved by previous proposals. Analysing the performance of Basic and Extended versions of the IMCRO approach when solving UA-FLPs, it seems that the Extended version reaches better results than the Basic one, since it is able to overcome

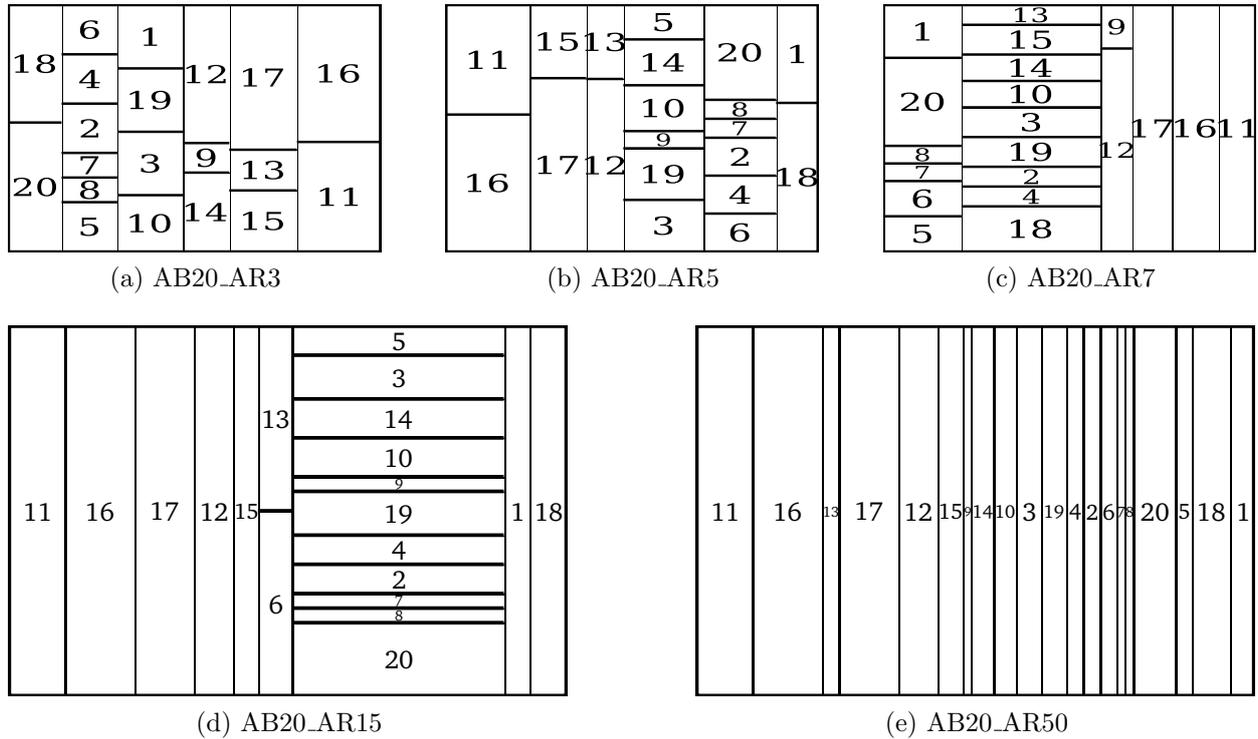


Figure 6: Best solutions obtained by the Basic IMCRO for some of the tested UA-FLP instances.

the results achieved by the Basic version of the IMCRO algorithm in 4 UA-FLP instances. Moreover, in the remaining 18 UA-FLPs, the Extended IMCRO version equals the best results achieved by the Basic IMCRO version. The IMCRO system has also demonstrated to be effective for addressing UA-FLPs of different categories according to their size, i.e., small, medium and large UA-FLPs. This effectiveness is due to the independence existing between each reef (or island), that encourages to explore different zones of the search space in the same amount of time, leading to an efficient use of the computing resources, which improves in a significant factor the time needed to find an optimal solution. On the other hand, migration helps increase diversity too, because of the previously mentioned islands independence.

Given the importance of a correct distribution of spaces in any installation, especially in industrial ones, the impact of the new proposal is high. The plant layout design affects substantially the efficiency of production and it is a relevant issue in every new plant construction or in the remodeling of existing ones. Here, the minimization of materials handling cost is considered to be crucial to the success of the design in terms of the production economy. Nevertheless, it is also important to take into account the shape of every individual department, in order to improve the plant performance. These two requirements are opposite, in the way that the worse the aspect ratio the better the material handling cost. So finding the layout which gives the best possible material handling cost while respecting aspect ratio restrictions is a critical issue that has not been satisfactorily solved yet, espe-

cially in problems of large size and complex material flow relationships among departments. In the present approach, a clear improvement has been obtained in most of the problems tested, so it is a promising line to be further explored.

Regarding future work, this research can be improved by incorporating new extended CRO versions to the suggested Island Model. Also, other migration strategies could be created and studied in order to further improve solution diversity between reefs, taken into account both individual distribution and migrating elements selection. Additionally, an interesting research line could be to apply these approaches over other facility layout problems with as, for example, Dynamic Facility Layout Problem that has the characteristic of varying material flow requirements. Another line to explore could be to use a more problem-driven initialization of the initial reefs' population by means of another approach such a specific heuristic method.

Acknowledgements

This research has been co-funded by the project of the Spanish Ministry of Economy, Industry and Competitiveness with reference: TIN2017-85887-C2-2-P, and also the Research Project "XXII PP. Mod. 4.1." of the University of Córdoba (Spain)

References

- Abraham, A., Au, E., Binotto, A., Garcia-Hernandez, L., Marik, V., Marmol, F. G., Snasel, V., Strasser, T. I., and Wahlster, W. (2020). Industry 4.0: Quo vadis? *Engineering Applications of Artificial Intelligence*, 87:103324.
- Abraham, A. and Jain, L. (2005). *Evolutionary Multiobjective Optimization*, pages 1–6. Springer London, London.
- Aiello, G., Enea, M., and Galante, G. (2006). A multi-objective approach to facility layout problem by genetic search algorithm and electre method. *Robotics and Computer-Integrated Manufacturing*, 22:447–455.
- Alba, E. and Troya, J. M. (1999). A survey of parallel distributed genetic algorithms. *Complex.*, 4(4):31–52.
- Armour, G. C. and Buffa, E. S. (1963). A heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, 9:294–309.
- Azadivar, F. and Wang, J. (2000). Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, 38(17):4369–4383.
- Bazaraa, M. S. (1975). Computerized layout design: A branch and bound approach. *A I I E Transactions*, 7(4):432–438.
- Bermejo, E., Chica, M., Damas, S., Salcedo-Sanz, S., and Cerdón, O. (2018). Coral reef optimization with substrate layers for medical image registration. *Swarm and Evolutionary Computation*, 42:138 – 159.
- Bozer, Y. A. and Meller, R. D. (1997). A reexamination of the distance-based facility layout problem. *IIE Transactions*, 29(7):549–560.
- Camacho-Gómez, C., Wang, X., Pereira, E., Díaz, I., and Salcedo-Sanz, S. (2018). Active vibration control design using the coral reefs optimization with substrate layer algorithm. *Engineering Structures*, 157:14 – 26.
- Cantu-Paz, E. (1998). A survey of parallel genetic algorithms. *CALCULATEURS PARALLELES*, 10.
- Castillo, I., Westerlund, J., Emet, S., and Westerlund, T. (2005). Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods. *Computers and Chemical Engineering*, 30(1):54–69.

- Chae, J. and Regan, A. C. (2016). Layout design problems with heterogeneous area constraints. *Computers & Industrial Engineering*, 102:198 – 207.
- Drira, A., Pierreval, H., and Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31(2):255–267.
- Dunker, T., Radons, G., and Westkämper, E. (2003). A coevolutionary algorithm for a facility layout problem. *International Journal of Production Research*, 41(15):3479–3500.
- Eiben, A. and Smith, J. (2003). *Introduction to Evolutionary Computing*. SpringerVerlag.
- Enea, M., Galante, G., and Panascia, E. (2005). The facility layout problem approached using a fuzzy model and a genetic search. *Journal of Intelligent Manufacturing*, 16:303–316. 10.1007/s10845-005-7025-4.
- García-Hernández, L., Arauzo-Azofra, A., Salas-Morera, L., Pierreval, H., and Corchado, E. (2013a). Recycling plants layout design by means of an interactive genetic algorithm. *Intelligent Automation & Soft Computing*, 19(3):457–468.
- García-Hernández, L., Arauzo-Azofra, A., Salas-Morera, L., Pierreval, H., and Corchado, E. (2015). Facility layout design using a multi-objective interactive genetic algorithm to support the DM. *Expert Systems*, 32(1):94–107.
- García-Hernández, L., Palomo-Romero, J. M., Salas-Morera, L., Arauzo-Azofra, A., and Pierreval, H. (2015). A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem. *Expert Systems with Applications*, 42(10):4697 – 4708.
- García-Hernández, L., Pierreval, H., Salas-Morera, L., and Arauzo-Azofra, A. (2013b). Handling qualitative aspects in unequal area facility layout problem: An interactive genetic algorithm. *Appl. Soft Comput.*, 13(4):1718–1727.
- García-Hernández, L., Salas-Morera, L., García-Hernández, J., Salcedo-Sanz, S., and de Oliveira, J. V. (2019). Applying the coral reefs optimization algorithm for solving unequal area facility layout problems. *Expert Systems with Applications*, 138:112819.
- Goldberg, D. E. and Lingle, R. (1985). Alleles, loci, and the traveling salesman problem. In Grefenstette, J. J., editor, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 154–159. Lawrence Erlbaum Associates, Publishers.
- Gomez, A., Fernandez, Q., la Fuente Garcia, D. D., and Garcia, P. (2003). Using genetic algorithms to resolve layout problems in facilities where there are aisles. *International Journal of Production Economics*, 84(3):271 – 282.
- Gonçalves, J. F. and Resende, M. G. (2015). A biased random-key genetic algorithm for the unequal area facility layout problem. *European Journal of Operational Research*, 246(1):86–107.
- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA.
- Kang, S. and Chae, J. (2017). Harmony search for the layout design of an unequal area facility. *Expert Systems with Applications*, 79:269 – 281.
- Komarudin and Wong, K. Y. (2010). Applying ant system for solving unequal area facility layout problems. *European Journal of Operational Research*, 202(3):730 – 746.
- 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.
- Kulturel-Konak, S. and 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.
- Kurdi, M. (2016). An effective new island model genetic algorithm for job shop scheduling problem. *Computers and Operations Research*, 67:132 – 142.
- Liu, F., Geng, H., and Zhang, Y.-Q. (2005). Interactive fuzzy interval reasoning for smart web shopping. *Applied Soft Computing*, 5(4):433 – 439.
- Liu, J. and Liu, J. (2019). Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems. *Applied Soft Computing*, 74:167 – 189.
- Liu, Q. and Meller, R. D. (2007). A sequence-pair representation and mip-model-based heuristic for the facility layout problem with rectangular departments. *IIE Transactions*, 39(4):377–394.

- Meller, R., Narayanan, V., and Vance, P. (1998). Optimal facility layout design. *Operations Research Letters*, 23(3-5):117–127.
- Meller, R. D. and Gau, K.-Y. (1996). The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 15(5):351–366.
- Montreuil, B. (1991). A modelling framework for integrating layout design and flow network design. In *Material Handling '90*, pages 95–115, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Norman, B. A. and Smith, A. E. (2006). A continuous approach to considering uncertainty in facility design. *Comput. Oper. Res.*, 33(6):1760–1775.
- Paes, F. G., Pessoa, A. A., and Vidal, T. (2017). A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem. *European Journal of Operational Research*, 256(3):742 – 756.
- Palomo-Romero, J. M., Salas-Morera, L., and Garcia-Hernandez, L. (2017). An island model genetic algorithm for unequal area facility layout problems. *Expert Systems with Applications*, 68:151 – 162.
- Purnomo, M. R. A. and Wiwoho, Y. S. (2016). Multi-objective mixed integer programming approach for facility layout design by considering closeness ratings, material handling, and re-layout cost. *IOP Conference Series: Materials Science and Engineering*, 105:012045.
- Ramadas, M. and Abraham, A. (2019). *Metaheuristics and Data Clustering*, pages 7–55. Springer International Publishing, Cham.
- Ripon, K. S. N., Glette, K., Khan, K. N., Hovin, M., and Torresen, J. (2013). Adaptive variable neighborhood search for solving multi-objective facility layout problems with unequal area facilities. *Swarm and Evolutionary Computation*, 8:1 – 12.
- Salas-Morera, L., Cubero-Atienza, A., and Ayuso-Munoz, R. (1996). Computer-aided plant layout — Distribucion en planta asistida por ordenador. *Informacion Tecnologica*, 7(4):39–46.
- Salcedo-Sanz, S. (2017). A review on the coral reefs optimization algorithm: new development lines and current applications. *Progress in Artificial Intelligence*, 6(1):1–15.
- Salcedo-Sanz, S., Camacho-Gómez, C., Mallol-Poyato, R., Jiménez-Fernández, S., and DelSer, J. (2016). A novel coral reefs optimization algorithm with substrate layers for optimal battery scheduling optimization in micro-grids. *Soft Computing*, 20(11):4287–4300.
- Salcedo-Sanz, S., Del Ser, J., Landa-Torres, I., Gil-López, S., and Portilla-Figueras, J. A. (2014a). The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *Scientific World Journal*, 2014.
- Salcedo-Sanz, S., Pastor-Sánchez, A., Prieto, L., Blanco-Aguilera, A., and García-Herrera, R. (2014b). Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization – extreme learning machine approach. *Energy Conversion and Management*, 87:10 – 18.
- Saraswat, A., Venkatadri, U., and Castillo, I. (2015). A framework for multi-objective facility layout design. *Computers & Industrial Engineering*, 90:167 – 176.
- Scholz, D., Petrick, A., and 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.
- See, P. C. and Wong, K. Y. (2008). Application of ant colony optimisation algorithms in solving facility layout problems formulated as quadratic assignment problems: a review. *International Journal of Industrial and Systems Engineering*, 3(6):644–672.
- Shayan, E. and Chittilappilly, A. (2004). Genetic algorithm for facilities layout problems based on slicing tree structure. *International Journal of Production Research*, 42(19):4055–4067.
- Sherali, H., Fraticelli, B., and Meller, R. (2003). Enhanced model formulations for optimal facility layout. *Operations Research*, 51:629 – 644.
- Sikaroudi, A. M. E. and Shahanaghi, K. (2016). Facility layout by collision detection and force exertion heuristics. *Journal of Manufacturing Systems*, 41:21 – 30.
- Singh, S. and Sharma, R. (2006). A review of different approaches to the facility layout problems. *Int J Adv Manuf Technol*, 138:30–425.
- Sitkoff, N. (1995). Implementing a genetic algorithm on a parallel custom computing machine. In *Proceedings of the IEEE Symposium on FPGA's for Custom Computing Machines*, FCCM '95, pages 180–, Washington, DC, USA. IEEE Computer Society.

- Tam, K. Y. (1992). A simulated annealing algorithm for allocating space to manufacturing cells. *International Journal of Production Research*, 30(1):63–87.
- Tate, D. M. and Smith, A. E. (1995). Unequal area facility layout using genetic search. *IIE Transactions*, 27:465–472.
- Tong, X. (1991). *SECOT: A Sequential Construction Technique For Facility Design*. Doctoral Dissertation, University of Pittsburg.
- Ulutas, B. H. and Kulturel-Konak, S. (2012). An artificial immune system based algorithm to solve unequal area facility layout problem. *Expert Systems with Applications*, 39(5):5384 – 5395.
- van Camp, D., Carter, M., and Vannelli, A. (1992). A nonlinear optimization approach for solving facility layout problems. *European Journal of Operational Research*, 57(2):174–189.
- Whitley, D., Rana, S., and Heckendorn, R. B. (1998). The island model genetic algorithm: On separability, population size and convergence. *Journal of Computing and Information Technology*, 7:33–47.
- Wong, K. Y. and Komarudin (2010). Solving facility layout problems using flexible bay structure representation and ant system algorithm. *Expert Systems with Applications*, 37(7):5523 – 5527.
- Wu, Y. and Appleton, E. (2002). The optimisation of block layout and aisle structure by a genetic algorithm. *Comput. Ind. Eng.*, 41(4):371–387.

AUTHOR DECLARATION TEMPLATE

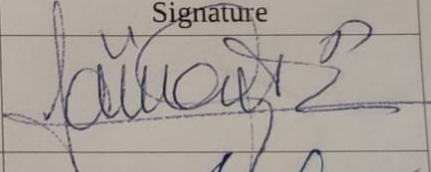
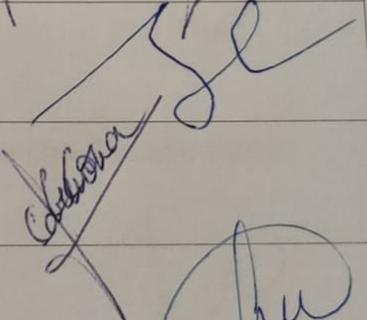
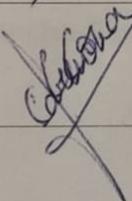
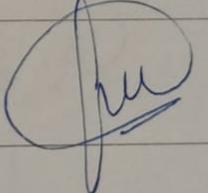
We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from (ir1gahel@uco.es)

Signed by all authors as follows:

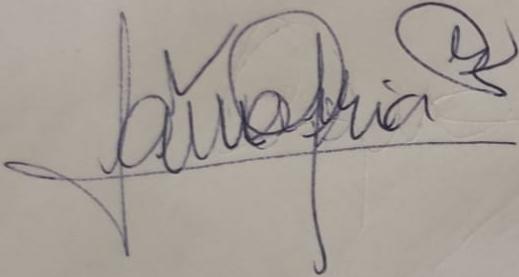
Author	Date	Signature
Laura Garcia-Hernandez	November 16, 2019	
Lorenzo Salas-Morera	November 16, 2019	
Carlos Carmona Muñoz	November 16, 2019	
Sancho Salcedo-Sanz	November 16, 2019	

Declaration of interests

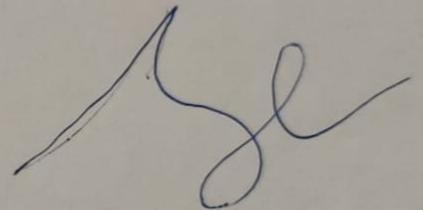
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Laura Garcia Hernandez:



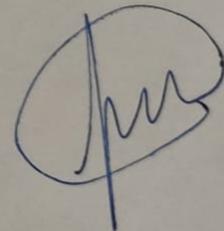
Lorenzo Salas Morera:



Carlos Carmona Muñoz:



Sancho Salcedo Sanz:



Authors' individual contributions

Manuscript title: A Novel Island Model Based on Coral Reefs Optimization Algorithm for Solving the Unequal Area Facility Layout Problem

Author 1: Laura García-Hernández

- x Conceptualization
- x Data curation
- x Formal analysis
- x Funding acquisition
- x Investigation
- x Methodology
- x Project Administration
- x Resources
- x Software
- x Supervision
- x Validation
- x Visualization
- x Roles/Writing - original draft
- x Writing - review & editing

Author 2: Lorenzo Salas-Morera

- x Conceptualization
- x Data curation

- x Formal analysis
- x Funding acquisition
- x Investigation
- x Methodology
- x Project Administration
- x Resources
- x Software
- x Supervision
- x Validation
- x Visualization
- x Roles/Writing - original draft
- x Writing - review & editing

Author 3: Carlos Carmona-Muñoz

- x Conceptualization
- x Data curation
- x Formal analysis
- Funding acquisition
- x Investigation
- x Methodology
- Project Administration
- x Resources
- x Software
- Supervision
- Validation

- x Visualization
- Roles/Writing - original draft
- x Writing - review & editing

Author 4: José Antonio García-Hernández

- Conceptualization
- Data curation
- Formal analysis
- Funding acquisition
- x Investigation
- Methodology
- Project Administration
- x Resources
- x Software
- Supervision
- Validation
- x Visualization
- Roles/Writing - original draft
- x Writing - review & editing

Author 5: Sancho Salcedo-Sanz

- x Conceptualization
- Data curation
- x Formal analysis
- x Funding acquisition

- x Investigation
- x Methodology
- x Project Administration
- x Resources
- Software
- x Supervision
- x Validation
- x Visualization
- x Roles/Writing - original draft
- x Writing - review & editing