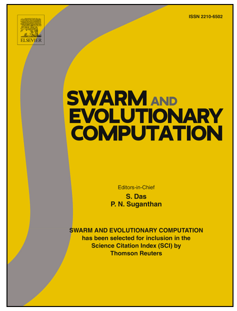
Journal Pre-proof



A novel multi-objective Interactive Coral Reefs Optimization algorithm for the Unequal Area Facility Layout Problem

L. Garcia-Hernandez, L. Salas-Morera, C. Carmona-Muñoz, A. Abraham, S. Salcedo-Sanz

PII: S2210-6502(19)30549-8

DOI: https://doi.org/10.1016/j.swevo.2020.100688

Reference: SWEVO 100688

To appear in: *Swarm and Evolutionary Computation BASE DATA*

Received Date: 19 July 2019

Revised Date: 27 February 2020

Accepted Date: 19 March 2020

A Novel Multi-objective Interactive Coral Reefs Optimization Algorithm for the Unequal Area Facility Layout Problem

L. Garcia-Hernandez[a], L. Salas-Morera[a], C. Carmona-Muñoz[a], A. Abraham[b], S. Salcedo-Sanz[c]

e-mail: ir1gahel@uco.es, lsalas@uco.es, i32camuc@uco.es, ajith.abraham@ieee.org, sancho.salcedo@uah.es

[a]*Area of Project Engineering, University of Cordoba, Spain.*
[b]*Machine Intelligence Research Labs (MIR Labs) Scientific Network for Innovation and Research Excellence Auburn, Washington 98071, USA*
[c]*Department of Signal Processing and Communications, Universidad de Alcalá, Alcalá de Henares, Madrid, Spain.*

Abstract

The Unequal Area Facility Layout Problem (UA-FLP) has been widely analyzed in the literature using several heuristics and meta-heuristics to optimize some qualitative criteria, taking into account different restrictions and constraints. Nevertheless, the subjective opinion of the designer (Decision Maker, DM) has never been considered along with the quantitative criteria and restrictions. This work proposes a novel approach for the UA-FLP based on an Interactive Coral Reefs Optimization (ICRO) algorithm, which combines the simultaneous consideration of both quantitative and qualitative (DM opinion) features. The algorithm implementation is explained in detail, including the way of jointly considering quantitative and qualitative aspects in the fitness function of the problem. The experimental part of the paper illustrates the effect of including qualitative aspects in UA-FLP problems, considering three different hard UA-FLP instances. Empirical results show that the proposed approach is able to incorporate the DM preferences in the obtained layouts, without affecting much to the quantitative part of the solutions.

Keywords:
UA-FLP, Coral Reefs Optimization, Interactive Algorithms, Bio-inspired algorithms

1. Introduction

Facility Layout Design (FLD) arranges the disposition of a number of facilities (or *departments*) in a manufacturing system, in order to accomplish a given design objective (or objectives), while satisfying certain constraints. A satisfactory facility disposition directly affects the efficiency of the manufacturing system and it has been associated with reductions between 20% and 50% of the total cost in an industrial company [1]. Therefore, obtaining a high quality FLD is considered as very important to reduce waiting times in the manufacturing of products and production costs [2]. Different Facility Layout Problems (FLPs) have been identified related to FLD, depending on several design factors. Thus, different FLP classifications have been described in the works by Drira et al. [3], Anjos and Vieira [4] and Hosseini-Nasab et al. [5], among others. In this sense, the Unequal Area Facility Layout Problem (UA-FLP) is one of the most important FLPs, because of its direct application to real world problems.

The UA-FLP was first formulated by Armour and Buffa [6], and it consideres a rectangular region and a set of departments (also with rectangular shape). These departments have to be distributed in the most adequate arrangement within the boundary of the space plant layout, considering as restriction that the overlap between facilities is not permitted. Usually, quantitative performance (particularly, minimizing the total handling cost between the departments that exist in the industrial plant [7]) is considered as the main optimization criterion in the UA-FLP. However, there is additional qualitative knowledge that is sometimes very relevant in the facility layout design [8]. According to Tuzkaya and Ertay [9], qualitative aspects must be taken into account in order to obtain robust layouts. Some of the qualitative aspects considered could be: specific department position or orientation, remaining space allocation and /or distribution, or any other qualitative feature that can be considered as relevant for the Decision Maker (DM). Note that qualitative aspects of UA-FLPs are difficult to be considered by means of a classical heuristic or meta-heuristic optimization approach [10]. According to García-Hernández et al. [11], this fact is due to these non quantitative aspects could be subjective, unknown beforehand or changing during the design procedure. This makes the inclusion of the DM into the optimization approach as primordial, in order to deal with these qualitative aspects of the design.

The Coral Reefs Optimization (CRO) algorithm is a recently-proposed evolutionary-type technique, in which the evolution mechanisms mimic the processes occurring in natural coral reefs. These operations are reproduction processes, the fight for space and the depredation of corals in the reef. The CRO finally results in a kind of combination of Simulated Annealing and Evolutionary Algorithms [12]. This strategy has been able to outperform other meta-heuristics algorithms in many different areas such as, for example, Bio-medical applications [13, 14], Telecommunications [15, 16], Structural Engineering [17, 18] or Energy [19, 20]. Furthermore, the CRO has been successfully applied to other hard optimization problems such as resource allocation problems [21], neural network training [22], clustering [23] and time series analysis [24]. Recently, this algorithm has been applied to solve a version of the UA-FLP problem [25].

In this paper, we propose a novel approach to address the UA-FLP that combines the Coral Reefs Optimization (CRO) with an interactive technique, in order to take into account simultaneously both quantitative and qualitative aspects in UA-FLPs. To the best of our knowledge, this is the first time that an interactive mechanism is merged with a CRO algorithm. Hence, the aim of the paper is twofold: first, to show the performance of the Interactive CRO (ICRO) in the UA-FLP instances, and second to evaluate the robustness of the CRO when an interactive process is included in the algorithm.

The rest of the paper is structured as follows: Section 2 details a literature review on the UA-FLP. Section 3 describes the novel proposed interactive CRO approach. Section 4 describes the experimentation performed in order to validate this research, offering the achieved results and their analysis. Finally, Section 5 concludes this work by means of a brief of the principal conclusions and some future research lines that this work open.

2. Unequal area facility layout problem

In order to address the UA-FLP, different techniques and algorithms have been considered. Following the classification by Komarudin and Wong [26], it is possible to divide them into deterministic methods and heuristics (or meta-heuristics) approaches. Regarding the first category, a first proposal based on a branch and bound method was suggested by Meller et al. [27]. Later, Montreuil [28] and Konak et al. [29] considered mixed integer programming for solving the UA-FLP. Afterwards, Meller et al. [30] adapted Montreuil's approach and applied to large-size UA-FLPs. In this context, Sherali et al. [31] proposed a improved technique in order to better resolve UA-FLPs in terms of efficiency. This approach was further modified by Castillo et al. [32] to reach optimal designs for an UA-FLP with a size of nine departments. More recently, optimal solutions for UA-FLPs up to 12 departments was achieved by Chae and Regan [33].

On the other hand, meta-heuristic approaches have gained importance during the last few years, since these approaches obtain in general better results than deterministic methods, specially in medium and large UA-FLPs. One of the first works dealing with meta-heuristics to solve UA-FPLs was carried out by Tam [34], who developed a Simulated Annealing proposal (LOGIC). Other meta-heuristics such as Tabu Search have also been tested in this problem, in works such as Scholz et al. [35] and Kulturel-Konak [36]. Genetic Algorithms (GAs) have been applied as well to UA-FLPs, as in the following related works: Tate and Smith [37], Wu and Appleton [38], Gomez et al. [39], Enea et al. [40], Aiello et al. [41], Liu and Meller [42], García-Hernández et al. [11], García-Hernández et al. [43] , García-Hernández et al. [44], Palomo-Romero et al. [45]. Alternative meta-heuristic proposals have been employed in order to resolve the UA-FLP. For example, the works of Komarudin and Wong [26], Wong and Komarudin [46], Kulturel-Konak and Konak [47] and Liu and Liu [48] used ant colony optimization. Ulutas and Kulturel-Konak [49] used an artificial immune system. Gonçalves and Resende [7] developed a biased random-key GA. Sikaroudi and Shahanaghi [50] applied collision detection and response method. Paes et al. [51] suggested an hybrid GA and a decomposition technique. Finally, Kang and Chae [52] proposed a variation of the Harmony Search algorithm suggested in the research by Shayan and Chittilappilly [53].

Generally, the majority of the methodologies applied to the UA-FLP considered a single objective. Usually this objective is the material handling cost as for example in the works by Kulturel-Konak and Konak [47], Kulturel-Konak [36], Gonçalves and Resende [7], Kang and Chae [52], Palomo-Romero et al. [45], among others. However, it is well-known that taking into account some different objectives results in better UA-FLP designs [54]. In this sense, Aiello et al. [54] stated that both quantitative and qualitative criteria should be considered at the same time in order to obtain more robust designs in UA-FLPs. Thus, Gomez et al. [39] proposed a multi-objective genetic algorithm for solving the UA-FLP. Their approach allowed the consideration of aisles in the plant layout. In the approaches suggested by Aiello et al. [41] and

Aiello et al. [55], a multi-objective genetic algorithm and the Electre method was applied for optimizing three objectives (material handling cost, closeness request and distance request). Also, Ripon et al. [56] solved UA-FLPs by means of a multi-objective approach that included a variable neighborhood search (VNS) and an adaptive strategy. In addition, in order to take into account more than one objective, both Saraswat et al. [57] and Purnomo and Wiwoho [58] employed the methodology suggested by Sherali et al. [31]. Moreover, Liu et al. [59] suggested a particle swarm optimization approach in order to solve the multi-objective UA-FLP. Their technique was based on an heuristic configuration mutation operation and a subsequent local search that considers a gradient method. Liu and Liu [48] proposed a multi-objective ant colony optimization algorithm to address the UA-FLP considering both material handling cost and distance requirements. They change the constrained problem into an unconstrained one by means of a heuristic technique, then, they applied a local search, Pareto Optimization, facility deformation and a niching method in order to reach effective facility designs. However, according to García-Hernández et al. [11] and García-Hernández et al. [43], considering exclusively these objectives (such as, for example: material handling cost, closeness request and distance request) may not represent all the relevant information that can appear in a facility layout design as the Decision Maker knowledge and experience. For that matter, they suggested a Genetic approach that includes the DM opinion in the search process. Lately, for considering both material handling cost and the DM preferences, García-Hernández et al. [60] proposed an interactive genetic approach that was applied to recycling plants. Finally, García-Hernández et al. [44] suggested a proposal that used a multi-objective interactive genetic algorithm to support the DM. In this proposal, the algorithm evolves exclusively using qualitative criteria and the DM's evaluation is required only every n generations. These evaluations are used to penalize all the members in the population and guide the search until the DM intervenes again n generations later. The process stops when the DM is completely satisfied with a solution.

Regarding the problem's encoding, mainly, three different representations have been used for solving the UA-FLP. These are the Block Layout Design Problem (BLDP) representation, the Slicing Tree Structure (STS) and the Flexible Bay Structure (FBS). The first one allows that each department of the plant layout can be allocated in any position of the plant considering that the overlapping is not permitted. As an example, the proposals by Meller and Gau [61], Castillo et al. [32], Gonçalves and Resende [7] used the BLDP representation as facility layout representation.

When evolutionary algorithms are considered, authors have mainly employed STS and FBS as layout representation. In STS, the plant is split in a recursive way into vertical and horizontal portions [53, 35, 62, 26, 52]. In the FBS representation however, the plant is exclusively split into vertical or horizontal sections [63, 47]. Consequently, STS can reach facility layout designs that cannot be achieved by FBS. So that, the results obtained by each one of them are not comparable in the majority of cases. Considering the FBS as layout representation presents as advantage that it makes the UA-FLP solving process easier and more simple [46]. This is due to the problem complexity is in general, smaller when this encoding is considered, because it selects the department position order and the number of departments that made up each bay of the plant layout. For this reason, the FBS is chosen in this work for representing a facility plant layout as a solution in an evolutionary-type algorithm. The FBS encoding was first described by Tong [64] and it has been widely applied in the works that deal with the UA-FLP [46, 45, 48]. Briefly, the FBS splits a rectangular region into vertical or horizontal sub-regions (called bays). Next, the facilities of the plant layout are inserted in each sub-region. In accordance with Tate and Smith [37], the originated sub-regions have the characteristic of having flexible amplitude for offering the necessary are in order to allocate varying number of departments. The FBS also offers the advantage of allowing an easy incorporation of passageways in the plant layout [55].

3. Proposed approach

In order to solve the UA-FLP, in this paper we propose a novel interactive multi-objective CRO algorithm. Our approach takes into account two optimization objectives. The first one refers to the cost of moving material between the facilities that compose the plant layout. The second one is related to the Decision Maker satisfaction about his/her preferences for each particular UA-FLP design. In this section, the structure and development of the complete system is fully detailed.

3.1. The Coral Reef Optimization Algorithm

In this section, the basic Coral Reef Optimization Algorithm (Salcedo-Sanz et al. [12]) is described, as well as its different stages.

Let Λ be a model of a rectangular-shaped reef of size $M \times N$. Each space $\Lambda(i, j)$ can be empty or contain a coral $X_k(i, j)$ (a solution of the UA-FLP), where i and j are the coordinates determining the position of the coral X_k in the reef Λ. Below the algorithm process is described per stages:

1. **Initialization**: A fraction ρ_0 of the total spaces of the reef is occupied with randomly generated corals. The position of these corals is also selected randomly between the $N \times M$ possible spaces.

2. **Evolution**: Once the reef has been populated the evolution process begins. This process is divided in five phases:

   (a) **Sexual reproduction**: In this phase a number of new solutions (larvae set) is created from the ones belonging to the reef in order to compete for a place in the reef in the next stages. There are two ways to perform the sexual reproduction: external and internal (similar to crossover and mutation in classic Evolutionary Algorithms). Therefore a percentage F_b of the corals present in the reef is selected to pair and reproduce via external reproduction (also referred as Broadcast spawning) and the rest of them $(1 - F_b)$ reproduce themselves by means of internal sexual reproduction (brooding). The reproduction processes are described below:

      i. **Broadcast spawning**: from the set of corals selected for external sexual reproduction, couples are made randomly so a coral can become a parent only once per generation. Each couple generates a child which is released to the water to be placed later in the process. The crossover operator used in this work is PMX Goldberg and robert [65] for facility order and 2-point crossover for bay divisions.

      ii. **Brooding**: each one of the rest of the corals that were not selected in the previous phase $(1-F_b)$ produce a larva by means of a random mutation. The produced larvae are released to the water, just like in the previous step. The operators used for mutation are TWORS [66] for facility order and 1-bit-swap [66] in the case of bay divisions.

   (b) **Larvae setting**: In this step, all the larvae (new solutions) produced during the sexual reproduction phase try to find a spot in the reef to settle. A reef position (i, j) is chosen randomly and the larva will settle in that spot if one of these two conditions is fulfilled:

i. The spot is empty.

ii. The larva has a better health function (fitness) that the coral that currently occupies the spot.

Each larva will try to settle in the reef up to three times. If after that number of attempts the larva has not been able to settle down, it is discarded and not considered anymore in the current generation.

(c) **Asexual reproduction**: In this phase (also named budding) a fraction F_a of the corals with better fitness presented in the reef duplicate themselves including a small random mutation, and try to settle in the reef following the same procedure described in the previous step.

(d) **Depredation**: Last, a fraction F_d of the worst corals in the reef can be predated (erased) from the reef with a low probability P_d.

The Algorithm 1 summarizes the whole evolution process, which is also illustrated in Figure 1.

3.2. Individual encoding

Individuals belonging to the reef are represented following the bay structure proposed by Gomez et al. [39], which consists in two parts:

1. Facility sequence: order in which the facilities are placed along the plant (from top to bottom, left to right). This segment of the chromosome is a permutation from 1 to the total number of facilities n of the plant.

2. Bay divisions: describes the structure of the plant by marking the facilities that delimitate the end of each bay. To represent this information a binary vector can be used. The positions with value 1 are the ones that mark the end of a bay.

The Figure 2 shows a chromosome example, as well as its corresponding layout representation.

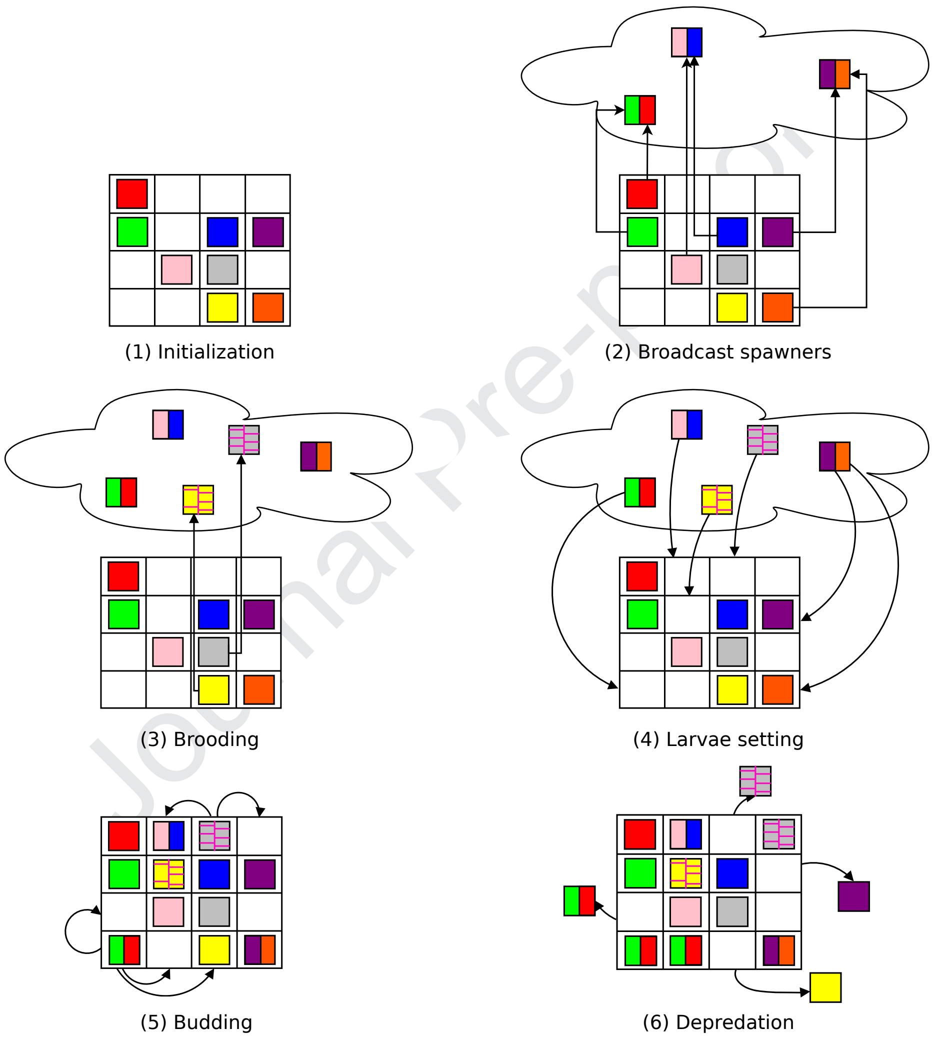


Figure 1: Phases of the basic CRO algorithm.

Algorithm 1 CRO algorithm

Input Algorithm's control parameters
Output Feasible solution with best *fitness*

procedure CRO($n, m, \rho_0, f_b, f_a, f_d, p_d$) ▷ Coral Reef Optimization algorithm
    initialize reef with size $n \times m$ and occupation rate ρ_0
    repeat
        reproduce corals fraction f_b by **broadcast spawning**
        reproduce corals fraction $1 - f_b$ by **brooding**
        larvae evaluation
        larvae setting
        reproduce best corals fraction f_a by **asexual reproduction**
        depredation of f_d worst reef corals with p_d probability
    until *stop condition*
    return best *feasible* solution
end procedure

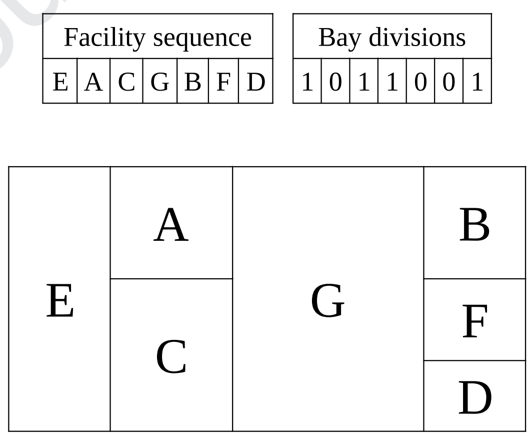


Figure 2: Facility layout chromosome example.

3.3. Objective function

Recall that the UA-FLP objective is to place n facilities of area A_i in a rectangular plant of fixed dimensions $W \times H$ in such a way that a given criteria are optimized. Naturally, the sum of the facilities' areas cannot surpass the area of the plant (Equation (1)).

$$\sum_{i}^{n} A_i \leq W \times H \tag{1}$$

The usual criterion to be optimized in UA-FLPs is the material flow between facilities (Aiello et al. [55]). This value must be minimized in order to improve the overall efficiency of the work performed in the plant. Therefore, material flow is taken into consideration in the fitness function used in this approach.

However, if this was the only factor to consider to evaluate candidate solutions the search could converge easily in "stacked" solutions that minimize the material flow but stretches the facilities too much. Therefore, a new objective to minimize comes into the picture, leading to the creation of a Pareto front: the feasibility of the proposed plant design. In order to address this issue, Tate and Smith [37] proposed a penalization for the solutions which contain these unfeasible facilities. First, each facility defines aspect ratio constraint (aspect ratio or minimum side length). Second, a penalty value proportional to the number of unfeasible facilities (if any) is added to a solution's fitness function. Equation (2) shows the fitness function that takes into account these two objectives.

$$V_t = \sum_{i}^{n} \sum_{j}^{n} f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all}) \tag{2}$$

where t represents a certain solution, n is the number of facilities to position in the plant layout, f_{ij} is the material flow between facilities i and j, d_{ij} is the distance between them (let it be rectilinear or euclidean), D_{inf} is the number of unfeasible facilities, k is a penalty parameter that adjusts the grade of penalization (set to 3, following the recommendation in Tate and Smith

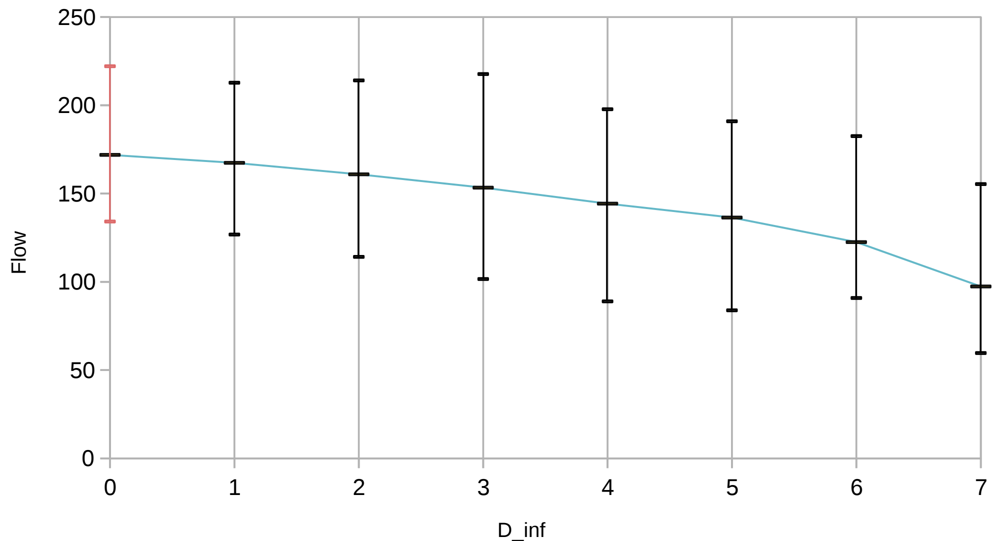


Figure 3: Relationship between material flow and D_{inf} parameters for the benchmark problem O7.

[37]), and V_{feas} and V_{all} are the best fitness value that has been found in the set of feasible solutions and the overall best fitness value found, respectively.

Material flow and number of unfeasible departments are usually opposite goals, as observed in Figure 3, which is obtained from the data of a large set of candidate solutions for the benchmark Problem O7 [30]. This shows how minimum, average and maximum (from bottom to top) flow values tend to be lower as the number of D_{inf} departments increases. Solutions belonging to the set that have no unfeasible departments are marked in red. Note that not all problems have completely feasible solutions. In those cases, the Pareto front is defined by the solutions that find a good balance between flow and department feasibility.

Additionally, there is another important factor in the evaluation of solutions. The objective of the proposed approach is to find good solutions that have a good (probably not the best) material flow value while keeping in mind the personal preferences of a domain expert. Those preferences also shape the problem's Pareto front, since their complete fulfillment is required. However, these constraints, if correctly formulated by the expert, can help the algorithm's search process, leading it to a better exploration of the search space. In any case, the preferences and its fulfillment is to be represented in the fitness function. In order to achieve that, an extra penalty factor U_t is defined (see Equation (3)). This factor has been designed to work in the same scale as the penalty factor in Tate and Smith's proposal [37] since it has provided good results in previous works.

$$U_t = (5 - x) \times \frac{n}{4} \tag{3}$$

where x is the user's evaluation given to the solution t (value from 1 to 5) and n is the number of facilities.

Having these three objectives in mind, the final fitness function is illustrated in Equation (4). Note that this function prioritize solutions fulfilling the Decision Maker's conditions (qualitative optimization) rather than the ones that have all their facilities disposed in a feasible way (quantitative optimization). This decision has been taken on the basis that this proposal, being interactive, needs to put the focus on the human side of things. The DM should not feel that his/her preferences are overlooked in order to perform a more analytic optimization.

$$V_t = (1 + U_t^k) \sum_i^n \sum_j^n f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all}) \tag{4}$$

The previous expression is a dynamically weighted fitness function. That means that a given solution will be penalized according to how bad it is considered when evaluated by a certain criterion (the worse a solution is according to that criterion, the more penalty it gets). In our case, for a solution with $D_{inf} = 2$ and $U_t = 4$ the penalty would be heavier on the user preferences' side. So that, department feasibility and user preferences act as corrective agents, where the last target has a heavier consideration when evaluating solutions. If no penalization has to be performed the fitness function is equal to the material flow. Figure 4 shows an example of how much weight each one of the three optimization objectives is given, on average per algorithm's generation, during a test run of the ICRO algorithm. As stated above, feasibility of solutions and user evaluation penalties are lighter in the fitness computation as both targets are fully optimized during execution.

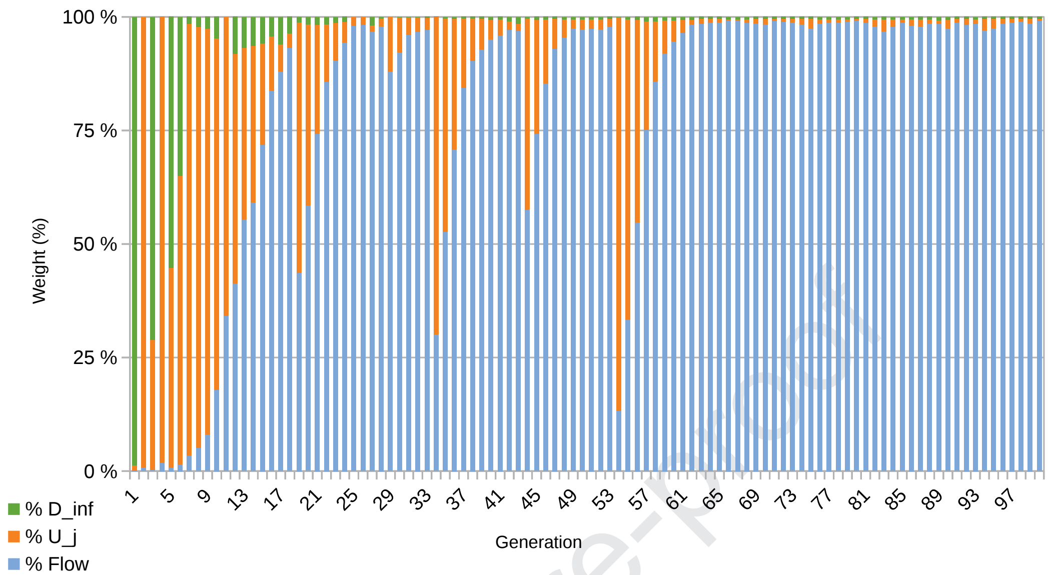


Figure 4: Average weight distribution of parameters in fitness function during an ICRO execution.

3.3.1. Graph example

In order to illustrate the calculation of layout fitness' an example is proposed in Figure 5. All facilities have an aspect ratio constraint $\alpha = 4$; rectilinear distance is used, the penalization factor is $k = 3$ and the solution has been evaluated by the user with a score $x = 3$. Material flow values are listed below:

$f_{AB} = 2$; $f_{AC} = 1$; $f_{AD} = 3$; $f_{AE} = 1$;
$f_{BC} = 2$; $f_{BD} = 1$; $f_{BE} = 3$;
$f_{CD} = 2$; $f_{CE} = 2$;
$f_{DE} = 1$;

Fitness computation proceeds as follows:

1. Find facility center coordinates:

   $C_A = (1, 1.5)$

   $C_B = (2.75, 2.5)$

   $C_C = (4, 1.5)$

   $C_D = (2.75, 0.25)$

   $C_E = (2.75, 1.25)$

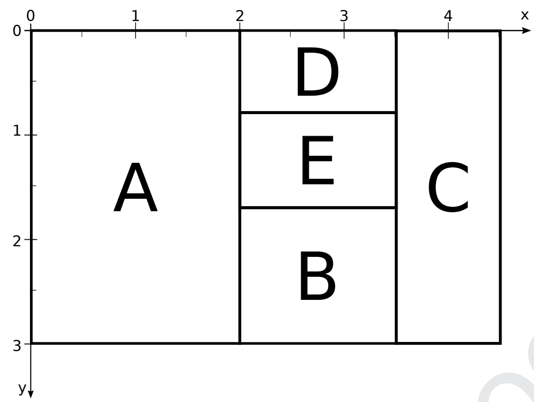


Figure 5: Facility layout example.

2. Compute distance between facilities:

   $d_{AB} = |x_{C_A} - x_{C_B}| + |y_{C_A} - y_{C_B}| = |1 - 2.75| + |1.5 - 2.5| = 2.75$

   $d_{AC} = |x_{C_A} - x_{C_C}| + |y_{C_A} - y_{C_C}| = 3$

   $d_{AD} = |x_{C_A} - x_{C_D}| + |y_{C_A} - y_{C_D}| = 3$

   $d_{AE} = |x_{C_A} - x_{C_E}| + |y_{C_A} - y_{C_E}| = 2$

   $d_{BC} = |x_{C_B} - x_{C_C}| + |y_{C_B} - y_{C_C}| = 2.25$

   $d_{BD} = |x_{C_B} - x_{C_D}| + |y_{C_B} - y_{C_D}| = 2.25$

   $d_{BE} = |x_{C_B} - x_{C_E}| + |y_{C_B} - y_{C_E}| = 1.25$

   $d_{CD} = |x_{C_C} - x_{C_D}| + |y_{C_C} - y_{C_D}| = 2.5$

   $d_{CE} = |x_{C_C} - x_{C_E}| + |y_{C_C} - y_{C_E}| = 1.5$

   $d_{DE} = |x_{C_D} - x_{C_E}| + |y_{C_D} - y_{C_E}| = 1$

3. Find unfeasible facilities (if any). In this case, a facility is unfeasible if its aspect ratio α is greater than 4:

   $\alpha_A = \frac{\max W_A, H_A}{\min W_A, H_A} = \frac{\max 2,3}{\min 2,3} = \frac{3}{2} = 1.5$ (feasible)

   $\alpha_B = \frac{\max W_B, H_B}{\min W_B, H_B} = 1.5$ (feasible)

   $\alpha_C = \frac{\max W_C, H_C}{\min W_C, H_C} = 3$ (feasible)

   $\alpha_D = \frac{\max W_D, H_D}{\min W_D, H_D} = 3$ (feasible)

   $\alpha_E = \frac{\max W_E, H_E}{\min W_E, H_E} = 1$ (feasible)

4. Compute U_t:
   $U_t = (5 - x) \times \frac{n}{4} = (5 - 3) \times \frac{5}{4} = 2.5$

5. Compute fitness:
   $V_t = (1 + U_t^k) \sum_i^n \sum_j^n f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all}) =$
   $= (1 + 1.25^3)(2 \cdot 2.75 + 1 \cdot 3 + 3 \cdot 3 + 1 \cdot 2 + 2 \cdot 2.25 + 1 \cdot 2.25 + 3 \cdot 1.25 +$
   $2 \cdot 2.25 + 2 \cdot 1.5 + 1 \cdot 1) + 0 =$
   $= (1 + 2.5^3) \cdot 39 = 648.375$

Note that if the user evaluation $x = 5$, $V_t = 39$ (material handling cost).

3.4. Evaluation

As stated in the previous section, a solution's fitness has two different parts. Note that the objective part (material flow and unfeasible penalty) is independent of the Decision Maker's (DM) intervention. On the other hand, if the DM had to evaluate each of the solutions created in each generation during the search process, it would be really difficult to keep the attention, and fatigue is very likely to appear. In order to avoid this, only a subset of the solutions are provided for DM evaluation. Particularly, nine representative individuals which are sufficiently different from each other. This way, the DM can evaluate directly nine solutions and the remaining are associated with an evaluation derived from the evaluated set.

The algorithm used to group the sets of solutions is the Fuzzy c-Means proposed by Dunn [67], which is fully explained in García-Hernández et al. [11]. This algorithm is similar to k-Means with the fundamental difference that an individual belongs, to a greater or lesser degree, to all the clusters defined. Similarity between solutions is computed as the sum of distances between the centers of each facility if both solutions are superimposed, so similar layouts are close to each other, composing a cluster. This way, the expected DM evaluation is assumed to be alike for a determined cluster's members.

The DM evaluates the most representative (higher membership value) solutions found in a solution set and the rest of the set is assigned an evaluation value following the expression in Equation (5).

$$\text{eval}_i = \sum_{j=1}^{c} m_{ij} e_j; \\ \sum_{j=1}^{c} m_{ij} = 1 \quad \forall i \tag{5}$$

where c is the number of clusters, m_{ij} is the membership value of the individual i to the cluster j and e_i is the score given to the representative individual of cluster j by the DM. This way, if a representative element gets a positive evaluation from the DM, any element that belongs strongly to that cluster will also obtain a positive evaluation.

There is still another aspect to consider in the proposed optimization process. In earlier works ([11], [44]), the search performed by the algorithm ended up when a solution evaluated by the DM with a value of 5 was found. This is no longer suitable for the objective proposed in this paper, since the objective solution minimizes the material flow while meets the DM's preferences and locates the facilities in a feasible way. Thus, in this case the algorithm's stop condition changes to be performed a number of iterations, and the DM is given a certain degree of control on how often his intervention is required. This also contributes to minimize user fatigue because when a solution rated with a 5 is found, a number of iterations (previously established) are performed without the DM's intervention. In any case, this user intervention frequency can be modified anytime during execution.

However, this possibility leads to the necessity to store all the evaluations (without repetition) the DM performs during the execution. Otherwise, the information about user preferences could be lost during the algorithm's evolutionary process and finish with a classic quantitative approach, which is not the goal of this work. Additionally, the symmetrical solutions to the ones evaluated must be considered too in order to do a proper estimation of the DM's score for the solutions which are not directly evaluated by the DM. Section 3.5 details how to identify and create symmetrical layouts to a given one.

3.5. Symmetrical layouts

In this section, a way to identify whether two facilities have a symmetrical layout when using FBS codification is described.

There are three types of symmetry in geometry: reflection, point and rotational. For the purposes of this work, only the first two are described:

1. **Reflection** symmetry: as its name suggests, this type relates to the perfect reflection of a reference object along an axis called line of symmetry or *mirror line*. This line can have any direction, but typically is vertical or horizontal. Figures 6 (b) and 6 (c) show examples of both cases of this symmetry type.

2. **Point** symmetry: also called origin symmetry, since all points have a corresponding one that are equally distant to an origin point but have opposite directions, as is showed in Figure 6 (d). In flat geometry, the point symmetrical of a figure can be obtained by means of a vertical and then horizontal mirroring, or vice versa.

It can be easily noticeable that material flow between facilities does not depend on the orientation of the plant, but on the distance separating facilities. Likewise, DM's preferences still fulfilled in layouts that only change their orientation. Hence a way to create symmetrical layouts to the ones the user have directly evaluated is really useful to correctly infer the evaluation the DM would concede to solutions that are not shown to him/her.

So, in this research, we distinguish three types of symmetrical layouts:

1. **Horizontal** symmetry: Horizontal mirroring of a plant. At the chromosome level, the difference is that the bays in the mirrored facility are exchanged (first with the last, second with second-to-last, etc.) while keeping the original structure, both in facility order and bay cuts. Figure 6 (b) illustrates this case.

2. **Vertical** symmetry: Vertical mirroring of a plant. At the codification level, the only change is the order of the facilities in each bay, which is inverted. Cuts remain the same as in the original layout. Figure 6 (c) shows the vertical reflection of the layout present in Figure 6 (a).

3. **Radial** symmetry: Both horizontal and vertical mirroring of a plant. This case joins the alterations of the two previous cases. So, by facility order, bays are exchanged as in horizontal symmetry and facility order in each bay is inverted (as in vertical symmetry), leading to a complete flip of the original order. Cuts, on their behalf, are redistributed by

Table 1: Summary of chromosome differences between symmetrical layouts.

	Original layout	Horizontal symmetry	Vertical symmetry	Radial symmetry
Facility sequence	original	exchange bays	flip per bay	complete flip
Bay divisions	original	exchange bays	original	exchange bays

bays exchange, as in horizontal symmetry. In Figure 6 (d) the point symmetrical to Figure 6 (a) is represented.

The differences at the solution encoding level among the three types of symmetry are listed in Table 1.

3.6. The Interactive Coral Reef Optimization Algorithm

In the previous sections, several aspects of the Interactive Coral Reef Optimization Algorithm proposed (ICRO) have been presented and discussed. In summary, the algorithm works combining automatic Coral Reef Optimization with user intervention to guide the search. User's desired characteristics for the final design are expressed by means of his/her evaluation of a subset of representative solutions in several steps of the algorithm. Generally, the need of user's intervention will decrease gradually since the system will learn what are the desired characteristics. The following enumeration is a brief step by step description of the whole algorithm.

1. **Initialize reef**. The same initialization procedure as in basic CRO is performed.

2. **Group reef members** in 9 clusters using c-Means.

3. **Show cluster centers** for evaluation.

4. **Process evaluations**. The evaluated layouts and their evaluations (and their three symmetrical) must be stored to direct the search process. An update of fitness values of the reef is also performed.

5. **Compute number of automatic iterations** to perform before asking for user feedback. At first, the DM intervenes in the evaluation process after one iteration. But this behaviour can be altered in two ways:

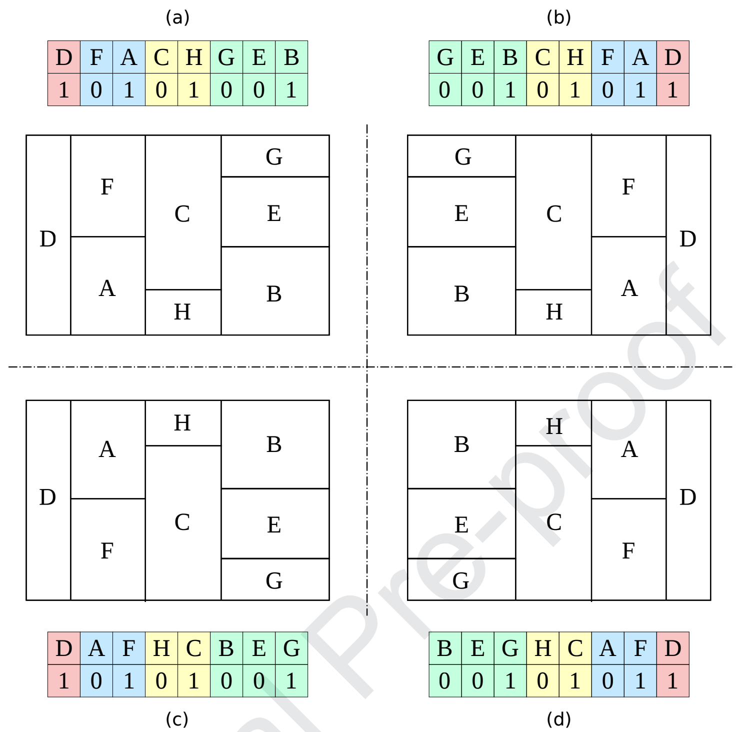


Figure 6: Example of a facility and its corresponding symmetrical layouts.

- If a layout evaluated with maximum score has been found, the next user intervention will occur after *user_freq* iterations, parameter fixed prior to the algorithm execution, or until the stop condition is fulfilled.
- If the user explicitly changes the value of *user_freq* parameter. In this case, the previous condition is ignored.

6. **Perform sexual reproduction**. Create larvae following the same procedure described in Section 3.1.

7. **Create random larvae**. In order to increase diversity in the solutions present in the reef and also to give the DM more options to rate, randomly created solutions are considered in the next step. The number of random larvae is computed as the l_r fraction of the larvae generated in the present step.

8. **Place larvae**. All larvae generated in the previous three phases are

placed in the reef following the same process described in Section 3.1.

9. **Perform asexual reproduction** as in basic CRO (Section 3.1).

10. **Predate** worst solutions as described in Section 3.1.

11. **Repeat**. At this point, three scenarios can take place:

    (a) Evolution **has not been performed** *num_iterations* times. Repeat from step 6.

    (b) Evolution **has been performed** *num_iterations* times. Repeat from step 2.

    (c) The algorithm's **stop condition** is **satisfied**. In that case, finish execution.

The described process is represented in Algorithm 2 as a pseudocode and in Figure 7 with a flow diagram.

4. Experiments and results

In this section, the performance of the suggested interactive CRO system is evaluated and compared using all the previous approaches that handle qualitative features in UA-FLP. For that matter, we have taken the following UA-FLP instances: Slaughterhouse, proposed in Salas-Morera et al. [68] and detailed in García-Hernández et al. [44]; CartonPacks, illustrated in García-Hernández et al. [44]; and ChoppedPlastic, from García-Hernández et al. [60]. Their characteristics are detailed in next sub-sections. Experiments with each data set have been repeated 5 times.

4.1. Slaughterhouse instance

This case was first described by Salas-Morera et al. [68]. The facility plant dimensions are $30m \times 51.14m$. The facility characteristics of this problem are summarized in Table 2. The material flow that exist between the departments that made up the plant layout is illustrated in Figure 8. In this UA-FLP, the Decision Maker is interested in the following aspects:

1. The plant layout must be split into 4 or 5 bays.

2. Facility B must be in the perimeter of the plant.

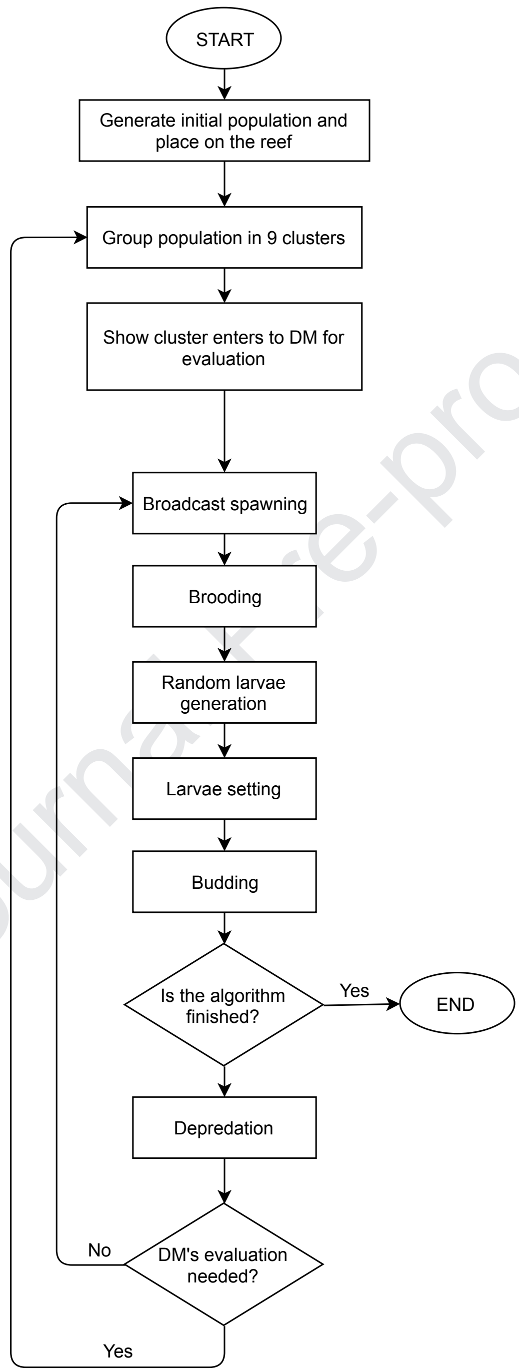


Figure 7: Flow diagram of the Interactive Coral Reef Optimization Algorithm.

Algorithm 2 ICRO algorithm

Input Algorithm's control parameters
Output Feasible solution with best *fitness* meeting user's requirements

```
procedure ICRO(n, m, ρ0, f_b, l_r, f_a, f_d, p_d, eval_freq)   ▷ Interactive Coral Reef Optimization algorithm
    initialize reef with size n × m and occupation rate ρ0
    repeat
        group reef in 9 clusters
        show cluster centers to user and collect user evaluations
        update reef members' fitness
        compute num_iterations
        repeat
            reproduce corals fraction f_b by broadcast spawning
            reproduce corals fraction 1 − f_b by brooding
            create n_l × l_r random larvae
            larvae evaluation
            larvae setting
            reproduce best corals fraction f_a by asexual reproduction
            depredation of f_d worst reef corals with p_d probability
        until done num_iterations times
    until stop condition
    return best feasible solution
end procedure
```

3. Facility A must be in the right bottom corner.

4. Facility F must be adjacent to facility A.

4.2. CartonPacks instance

This problem is referred to a carton recycling plant of $20m \times 14.5m$. It was proposed by García-Hernández et al. [44]. Table 3 described details (facility name, area and aspect ratio restriction) about the facilities that compose the plant. The requirements about material handling flow between facilities is represented by means of Figure 9. Related to the aspects that are advisable for the Decision Maker in order to reach a satisfactory solution, the following considerations must be taken into account:

Table 2: Facility features for the Slaughterhouse problem

Id	Facility	Area (m^2)	Aspect ratio
A	Stables	570	4
B	Slaughter	206	4
C	Entrails	150	4
D	Leather & skin	55	4
E	Aeration chamber	114	4
F	Refrigeration chamber	102	4
G	Entrails chamber	36	4
H	Boiler room	26	4
I	Compressor room	46	4
J	Shipping	109	4
K	Offices	80	4
L	Byproduct shipping	40	4

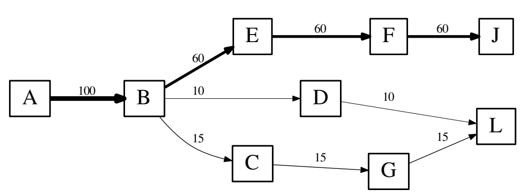


Figure 8: Material flow requirements for the Slaughterhouse problem.

1. Facility A must be in the perimeter of the plant.
2. Facility D must be in the perimeter of the plant.
3. Facility A must be near to facility E, F and A.
4. Facility C must be near to facility J, I, G and H.

Table 3: Facility features for the CartonPacks problem

Id	**Facility**	**Area (m^2)**	**Aspect ratio**
A	Raw Material	40	4
B	Finished products	40	4
C	Mechanic	20	4
D	Offices	50	4
E	Staff WC	20	4
F	Expedition	40	4
G	Hydraulic 1	20	4
H	Hydraulic 2	20	4
I	Crushing	20	4
J	Circ. saw	10	4
K	Heat exchange	10	4

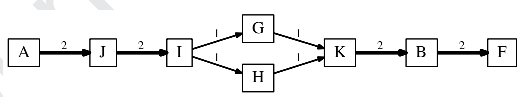


Figure 9: Material flow requirements for the CartonPacks problem.

4.3. ChoppedPlastic instance

The third tested problem was formulated by García-Hernández et al. [60]. This UA-FLP expresses a chopped plastic recycling plant layout which dimensions are $30m \times 10m$. The description about facilities and their associated areas and aspect ratio constraints are expressed in Table 4. Figure 10 offers information related to the existing material flow between facilities, which is sequential in this particular case. Below, the subjective aspects that are desired by the Decision Maker in this problem are listed:

1. The remaining space must be located in a corner of the plant.

2. Facility A must be in the perimeter of the plant.

3. Facility F must be in the perimeter of the plant.

4. Facility E must be adjacent to facility D.

Table 4: Facility features for the ChoppedPlastic problem

Id	Facility	Area (m^2)	Aspect ratio
A	Reception	35	4
B	Raw material	50	4
C	Washing	15	4
D	Drying & skin	24	4
E	Chopped	35	4
F	Finished product	30	4
G	Expedition	25	4
I	Office	30	4
J	Toilets	15	4
K	Repair shop	20	4

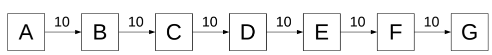


Figure 10: Material flow requirements for the ChoppedPlastic problem.

4.4. Parameters

The proposed algorithm's performance depends much on the values of its parameters, described in earlier sections. Specifically, exploration must be maximized as far as possible, in order to learn the DM's preferences in few generations and give time to improve accepted solutions. The parameters chosen for the ICRO algorithm are detailed in Table 5.

The ICRO algorithm has been coded in Python (version 2.7.3) and the experiments have been performed on an Intel Core i3 4010U (2 x 1.70GHz), 4GB RAM using a Linux-based OS.

Table 5: Chosen ICRO algorithm parameters.

Parameter	***Value***
Number of generations	100
Reef size	20×20
ρ_0	0.6
f_b	0.7
l_r	0.2
f_a	0.1
p_d	0.15
f_d	0.2
User interaction frequency	$1 \rightarrow 5$

4.5. Results

In this section, the results obtained by the proposed algorithm in the aforementioned problems are presented and discussed. Tables 6, 7 and 8 depict the following information per algorithm's run: the first section is referred to the first solution found that was given an evaluation of 5 by the DM; the first column in the section corresponds to the generation when that solution was found, followed by that solution's layout, Material Handling Cost (MHC), number of unfeasible facilities and fitness value. The second section is referred to the best solution found during the whole execution. This solution is considered to be the one that has the lowest fitness value while maximizing the user's score, whether is directly obtained from the DM (integer values) or derived by the algorithm (real values). For example, in the fifth row of Table 6, column *Eval*, the value is 4.999, since this particular solution has not been directly evaluated by the DM, but given a score according to its similarity with other solutions from the user-evaluated set. The information displayed in this section is, per column, generation when that solution was found, layout, MHC, number of unfeasible facilities, evaluation value, and fitness value. The best solution found for each problem instance set has been highlighted in boldface. An interesting fact that can be observed in the tables is that the derived user evaluations are correct: the solutions with high evaluation values (more than 4.9) fulfill the DM's conditions, although they cannot be given a 5 because they have not been showed to the user, since they are not cluster centers.

Table 6: Result table for Slaughterhouse instance launches

	First solution with user score 5					***Best solution found***					
Launch	*Gen.*	*Layout*	*MHC*	D_{inf}	*Fitness*	*Gen.*	*Layout*	*MHC*	D_{inf}	*Eval*	*Fitness*
Slaughterhouse-L1	5	3 | 2, 5 | 12, 11 | 6, 8, 10, 9, 7 | 4, 1	7683.518	4	451715.013	95	1 | 9, 6, 7, 12, 4, 8 | 10, 5, 2 | 11, 3	5767.967	0	5	5767.967
Slaughterhouse-L2	0	3 | 8, 2 | 11, 10, 4 | 7, 9, 12, 5, 6, 1	7051.331	6	2.45e10	85	1 | 10, 6, 5 | 2, 8 | 9, 3, 7, 12 | 11, 4	5097.368	0	5	5097.368
Slaughterhouse-L3	0	3, 8, 2, 4 | 7, 9, 5 | 11, 6, 10, 12 | 1	6510.934	1	1.24e8	**34**	**3, 7, 12, 4 | 9, 8, 11 | 10, 6, 5, 2 | 1**	**4136.462**	**0**	**5**	**4136.462**
Slaughterhouse-L4	4	11, 1 | 3, 8, 10, 4, 6, 5, 12 | 7 | 2, 9	6697.062	5	1.52e7	74	1 | 6, 5, 2 | 10, 4, 12, 7 | 9, 11, 8, 3	4366.876	0	5	4366.876
Slaughterhouse-L5	0	7, 9 | 10 | 3, 11, 2 | 5, 6, 8, 4 | 12, 1	7503.720	4	7.2e9	43	11, 12, 7, 8 | 9, 4, 10 | 2, 5, 6 | 1, 3	5026.563	0	4.999	5026.563

In all cases, from Tables 6, 7 and 8, it may be observed that even though a good user evaluation has been already given in the first generations, during the execution of the algorithm solutions with good values of MHC without losing user satisfaction have been found, which satisfies both criteria. It is also remarkable the fact that the MHC and fitness values of best solutions found are always equal, which confirms that the maximum score from the user and the minimum of unfeasible facilities have been simultaneously found.

In Table 9 it is shown a comparison between different approaches for the UA-FLP instances addressed in this work, both qualitative and quantitative. For each instance, the first two columns correspond to the results delivered by the presented ICRO algorithm (best fitness found and mean fitness of the best solutions obtained). Next, fitness value of the best solution obtained by previous only-qualitative approaches are listed. Note that these solutions reached the maximum user score in their evaluations. Finally, the last column shows the best fitness values found in the literature for these instances, focusing exclusively in material flow and feasibility (quantitative optimization). The conclusion is clear: the proposed ICRO approach is capable of optimize plant layouts while keeping the DM's constraints. That is why CartonPacks and ChoppedPlastic instances have a better performance compared to the best known solution and Slaughterhouse does not: the DM's preferences are guiding the search, and if the DM's preferences conflict with the shape of the best solution considering only MHC those values will never be reached. That is also suitable for the other two instances: in theses cases, the DM's preferences have led to find better solutions than the ones previously considered as the "best" ones.

On the other hand, Figures 11, 12 and 13 summarize the evolution of the ICRO for each instance launch. Four lines are showed: the first two ones correspond to mean fitness per generation and mean user evaluation

Table 7: Result table for CartonPacks instance launches

<table>
<tr><th></th><th colspan="5">First solution with user score 5</th><th colspan="6">Best solution found</th></tr>
<tr><th>Launch</th><th>Gen.</th><th>Layout</th><th>MHC</th><th>D_{inf}</th><th>Fitness</th><th>Gen.</th><th>Layout</th><th>MHC</th><th>D_{inf}</th><th>Eval</th><th>Fitness</th></tr>
<tr><td>CartonPacks-L1</td><td>3</td><td>3, 7, 9, 6 | 8, 10, 5, 11, 4 | 1, 2</td><td>98.214</td><td>2</td><td>98.214</td><td>23</td><td>3, 9, 7, 6 | 8, 11, 2 | 10, 4 | 1, 5</td><td>73.480</td><td>0</td><td>5</td><td>73.480</td></tr>
<tr><td>CartonPacks-L2</td><td>1</td><td>6 | 4, 1 | 7, 11, 10, 9 | 5 | 3 | 2, 8</td><td>95.846</td><td>3</td><td>95.846</td><td>31</td><td>1, 4 | 10, 6 | 9, 2, 5 | 8, 11, 7, 3</td><td>59.525</td><td>0</td><td>4.985</td><td>59.529</td></tr>
<tr><td>CartonPacks-L3</td><td>1</td><td>5, 4, 2, 1 | 11, 8 | 6 | 10, 3, 7 | 9</td><td>111.254</td><td>4</td><td>111.254</td><td>33</td><td>8, 9, 10, 3 | 11, 1 | 7, 5 | 2, 6, 4</td><td>61.312</td><td>0</td><td>4.995</td><td>61.312</td></tr>
<tr><td>CartonPacks-L4</td><td>2</td><td>11, 7, 3, 1 | 10 | 9, 5, 4, 6, 8, 2</td><td>107.915</td><td>6</td><td>5.98e5</td><td>32</td><td>5, 6 | 9, 4 | 7, 8, 10, 2 | 3, 11, 1</td><td>88.308</td><td>0</td><td>5</td><td>88.308</td></tr>
<tr><td>CartonPacks-L5</td><td>1</td><td>3, 8, 9, 7 | 11 | 2, 10, 1 | 5, 4 | 6</td><td>69.648</td><td>2</td><td>69.648</td><td>8</td><td>3, 8 | 9, 7, 11 | 2, 10, 1 | 5, 6, 4</td><td>79.838</td><td>0</td><td>4.9814</td><td>79.848</td></tr>
</table>

Table 8: Result table for ChoppedPlastic instance launches

<table>
<tr><th></th><th colspan="5">First solution with user score 5</th><th colspan="6">Best solution found</th></tr>
<tr><th>Launch</th><th>Gen.</th><th>Layout</th><th>MHC</th><th>D_{inf}</th><th>Fitness</th><th>Gen.</th><th>Layout</th><th>MHC</th><th>D_{inf}</th><th>Eval</th><th>Fitness</th></tr>
<tr><td>ChoppedPlastic-L1</td><td>0</td><td>3, 1 | 2 | 6, 4, 8 | 9, 10, 5 | 7, 11</td><td>571.894</td><td>5</td><td>1.15e7</td><td>17</td><td>8, 10, 2, 3, 4, 5, 7—11, 1, 9, 6</td><td>282.777</td><td>0</td><td>4.903</td><td>286.774</td></tr>
<tr><td>ChoppedPlastic-L2</td><td>0</td><td>11 | 10 | 3, 5, 4, 8 | 7, 9, 6 | 2, 1</td><td>805.615</td><td>5</td><td>6.49e6</td><td>30</td><td>1, 2, 3, 4, 5, 6, 7 | 11, 8, 9, 10</td><td>257.943</td><td>0</td><td>5</td><td>257.943</td></tr>
<tr><td>ChoppedPlastic-L3</td><td>0</td><td>11 | 9, 10, 7, 4, 5, 1 | 6, 8 | 3 | 2</td><td>491.845</td><td>5</td><td>9.6e6</td><td>35</td><td>11, 7, 9, 8 | 6, 4, 3, 2 | 5, 10, 1</td><td>346.228</td><td>0</td><td>5</td><td>346.228</td></tr>
<tr><td>ChoppedPlastic-L4</td><td>0</td><td>10, 2, 8, 5, 1, 7, 3, 11 | 6, 4, 9</td><td>776.729</td><td>2</td><td>4.87e5</td><td>45</td><td>6, 5, 4, 3, 2, 10, 8 | 7, 9, 1, 11</td><td>271.453</td><td>0</td><td>5</td><td>271.453</td></tr>
<tr><td>ChoppedPlastic-L5</td><td>0</td><td>3, 1, 11 | 2 | 9 | 10 | 4 | 6, 5, 7, 8</td><td>547.023</td><td>5</td><td>8.01e6</td><td>25</td><td>7, 6, 3, 2, 8 | 10, 5, 4, 1, 9, 11</td><td>330.142</td><td>0</td><td>5</td><td>330.142</td></tr>
</table>

per generation, respectively. The other two are vertical lines and point to the generations when a solution was given a 5 score by the DM and when the best overall solution was found. As can be observed, the learning process is usually fast and, in cases where not all the preferences are satisfied, it is possible to correct the course (Slaughterhouse - Launch 1) if a solution which fulfills all preferences is shown to the DM. In this type of scenario, random larvae creation proves its usefulness. Another remarkable fact is that there is a strong negative correlation between fitness value and user evaluation: the DM's satisfaction is critical for the approach presented in this work.

Table 9: ICRO algorithm performance comparison.

Problem name	Best ICRO Fitness	Mean Best ICRO Fitness	Best Multi-objective IGA Fitness	Best known Fitness
Slaughterhouse	4136.462	4879.047	5772 [a]	**3854.00** [c]
CartonPacks	**50.529**	72.495	61.449 [b]	94.10 [c]
ChoppedPlastic	**257.943**	298.508	380.736 [b]	377.18 [b]

[a] García-Hernández et al. [44]
[b] García-Hernández et al. [60]
[c] Enea et al. [40]

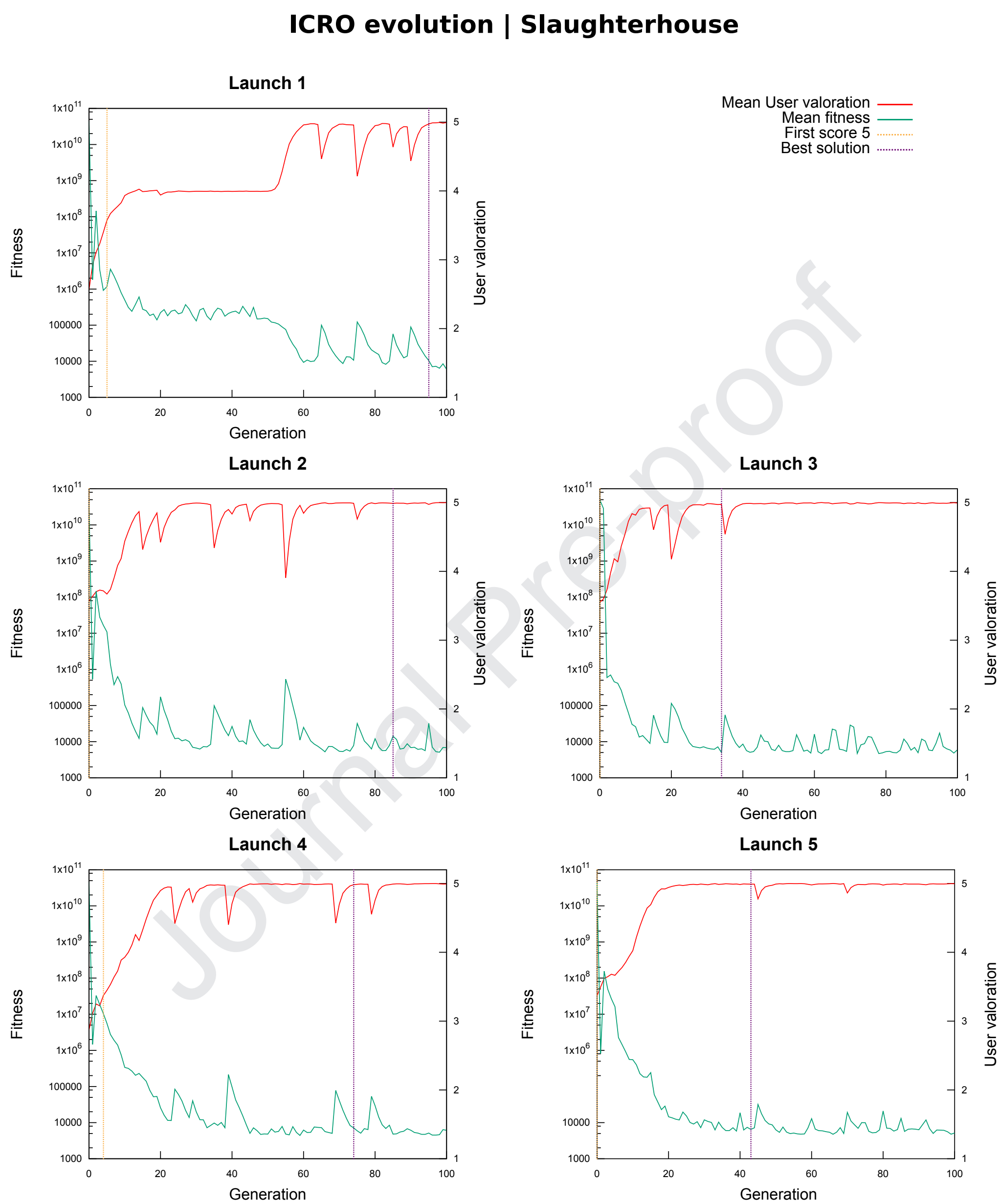


Figure 11: ICRO Evolution per launch in Slaughterhouse instance.

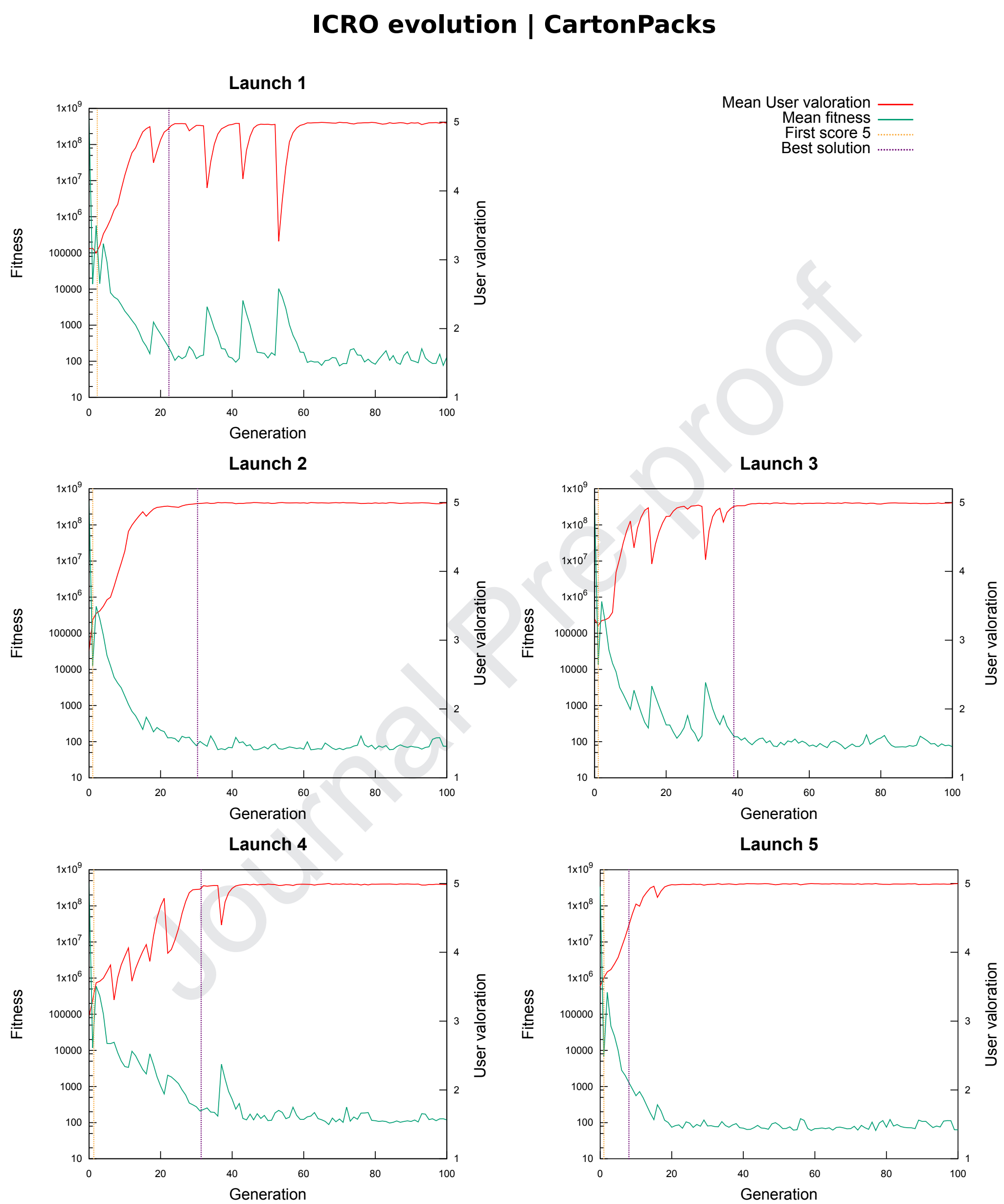


Figure 12: ICRO Evolution per launch in CartonPacks instance.

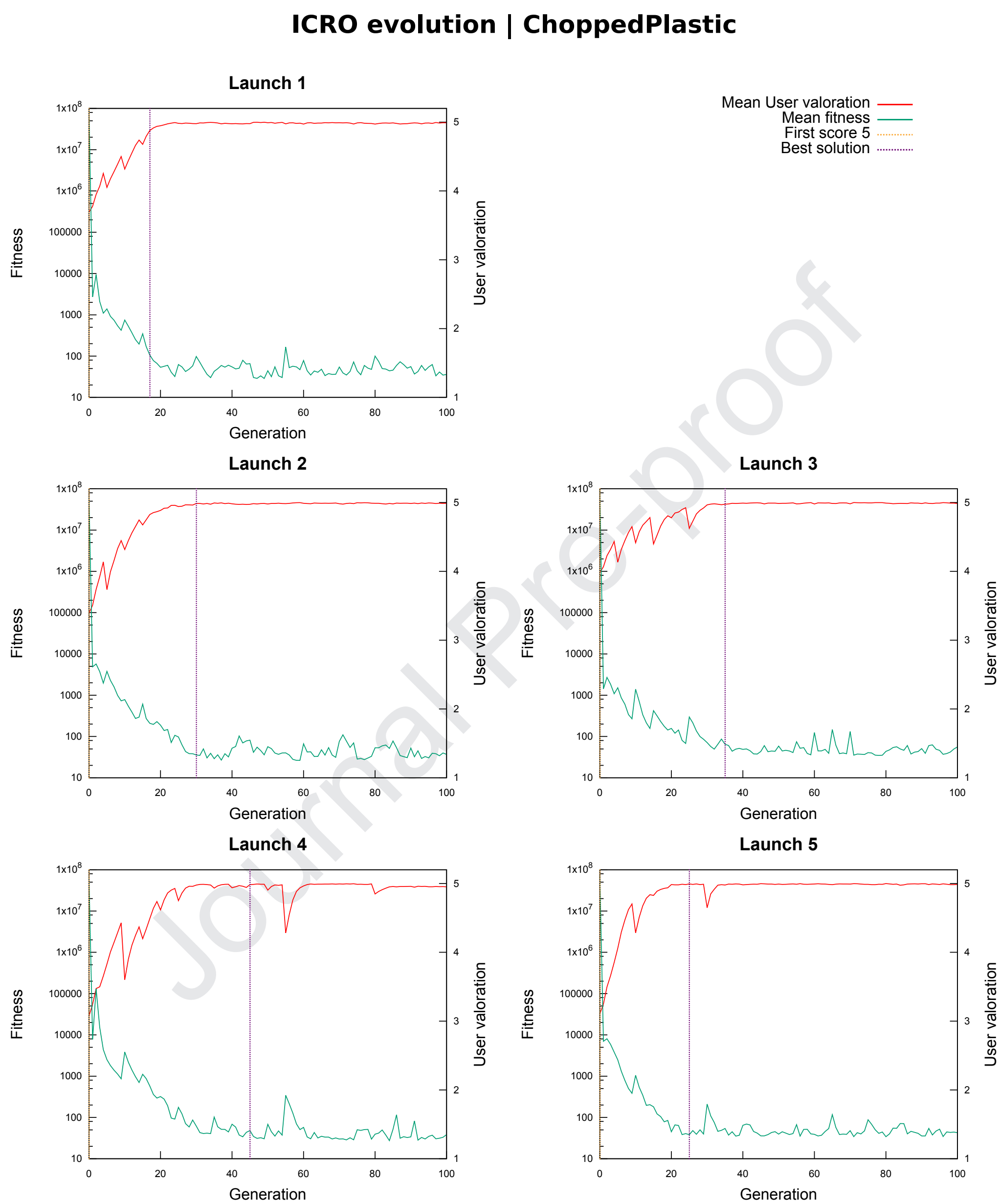


Figure 13: ICRO Evolution per launch in ChoppedPlastic instance.

Finally, Figures 14, 15 and 16 show the layouts of the best solutions obtained by the ICRO algorithm.

It is a well known fact that plant layout design is a problem that, when properly resolved, increases efficiency and optimizes industrial production costs in a very remarkable way. In this sense, the algorithm proposed and the results obtained contribute to innovate the way in which plant layouts are designed, and improve their subsequent performance in the real world.

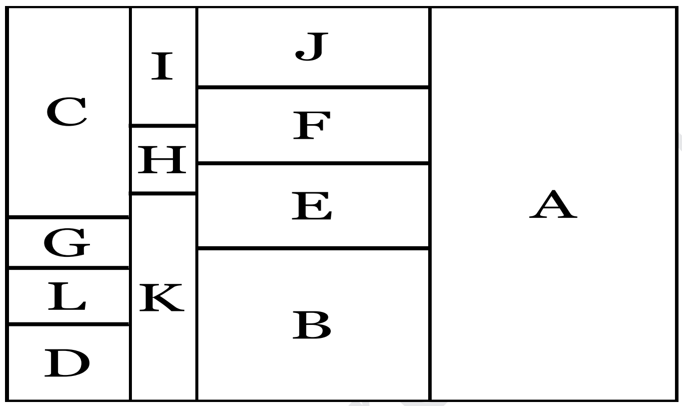


Figure 14: Best layout found by the ICRO algorithm for Slaughterhouse instance.

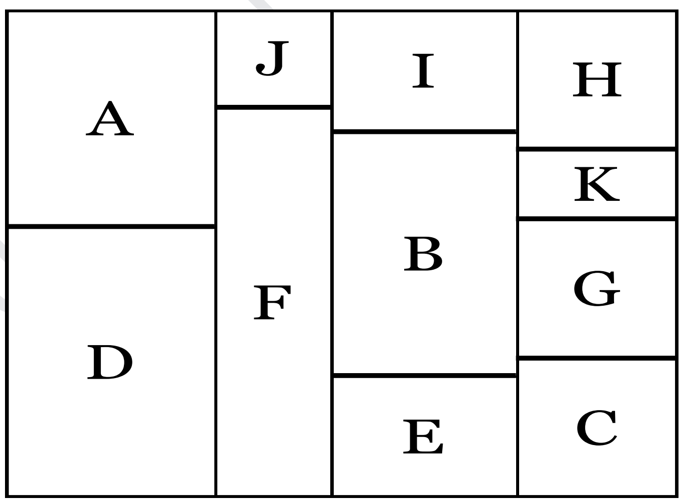


Figure 15: Best layout found by the ICRO algorithm for CartonPacks instance.

5. Conclusions

In this article, a novel meta-heuristic that allows considering both qualitative and quantitative aspects in Unequal Area Facility Layout Problems

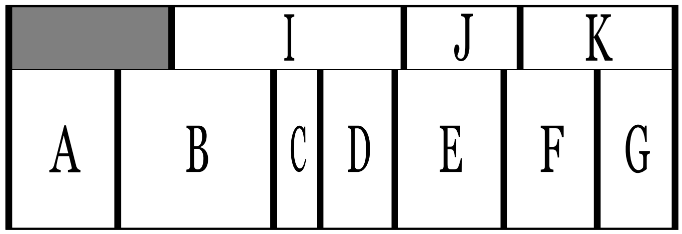


Figure 16: Best layout found by the ICRO algorithm for ChoppedPlastic instance.

(UA-FLP) has been proposed. Specifically, an Interactive Coral Reefs Optimization (ICRO) approach has been proposed for this problem.

A new interactive strategy, which allows discontinued user intervention and minimizes visual fatigue without losing precision in layout evaluation has been designed to be included in the CRO. This way a guidance to the search process that aims to solutions that fulfil user subjective preferences is achieved, not forgetting objective quantitative design aspects optimization.

The novel ICRO has been tested using three different UA-FLPs taken from related references. The evaluation and analysis performed using the reached solutions demonstrate that our proposed approach is able to reach excellent design solutions in all the tested UA-FLPs. This is due to the achieved solutions are considered as satisfactory for the Decision Maker (DM), i.e. they satisfy all the DM requirements, and also, these solutions have good material flow performance.

Possible future research could be to include alternative methods of layout representation together with the ICRO algorithm for solving UA-FLPs, and test advanced versions of the CRO approach such a ensemble CRO version with several *substrates* [69] in this problem. Similarly, it would be possible to allow the user to change his/her mind during the evolution process because the user is displayed a new solution with desirable features that had not been taken into account until then. Another improvement would be to handle in a more effective way the evaluated solution set, discarding the ones less significant, in order to reduce execution time per generation. Additionally, new ways to incorporate the expert knowledge into the approach could be investigated.

Acknowledgements

This work has been partially supported by the project TIN2017-85887-C2-2-P of the Spanish Ministry of Economy, Industry and Competitiveness, Government of Spain. The authors acknowledge the funding by the program to promote research of the University of Córdoba (Spain), under the Research Project XXII PP. Mod. 4.1 and the Junta de Andalucía, under the Research Project UCO-FEDER 18 REF. 1265277 MD A1.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Tompkins, J. White, Y. Bozer, J. Tanchoco, Facilities Planning, Wiley, New York, 4rd ed. edition, 2010.

[2] P. Kouvelis, A. A. Kurawarwala, G. J. Gutierrez, Algorithms for robust single and multiple period layout planning for manufacturing systems, European Journal of Operational Research 63 (1992) 287–303.

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

[4] M. F. Anjos, M. V. Vieira, Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions, European Journal of Operational Research 261 (2017) 1 – 16.

[5] H. Hosseini-Nasab, S. Fereidouni, S. M. T. Fatemi Ghomi, M. B. Fakhrzad, Classification of facility layout problems: a review study, The International Journal of Advanced Manufacturing Technology 94 (2018) 957–977.

[6] G. C. Armour, E. S. Buffa, A heuristic algorithm and simulation approach to relative location of facilities, Management Science 9 (1963) 294–309.

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

[8] M. Babbar-Sebens, B. S. Minsker, Interactive genetic algorithm with mixed initiative interaction for multi-criteria ground water monitoring design, Applied Soft Computing 12 (2012) 182 – 195.

[9] U. Tuzkaya, T. Ertay, An application of fuzzy ahp/dea methodology for the facility layout design in the presence of both quantitative and qualitative data, in: Applied Computational Intelligence - Proceedings of the 6th International FLINS Conference, pp. 507–512.

[10] A. M. Brintup, J. Ramsden, A. Tiwari, An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization, Computers in Industry 58 (2007) 279–291.

[11] L. García-Hernández, H. Pierreval, L. Salas-Morera, A. Arauzo-Azofra, Handling qualitative aspects in unequal area facility layout problem: An interactive genetic algorithm, Appl. Soft Comput. 13 (2013) 1718–1727.

[12] S. Salcedo-Sanz, J. Del Ser, I. Landa-Torres, S. Gil-López, J. A. Portilla-Figueras, The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems, Scientific World Journal 2014 (2014).

[13] E. Bermejo, M. Chica, S. Damas, S. Salcedo-Sanz, O. Cordón, Coral reef optimization with substrate layers for medical image registration, Swarm and Evolutionary Computation 42 (2018) 138 – 159.

[14] C. Yan, J. Ma, H. Luo, A. Patel, Hybrid binary coral reefs optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical datasets, Chemometrics and Intelligent Laboratory Systems 184 (2019) 102 – 111.

[15] S. Salcedo-Sanz, J. Sanchez-Garcia, P.-F. J.A., S. Jimenez-Fernandez, A. Ahmadzadeh, A coral-reef optimization algorithm for the optimal service distribution problem in mobile radio access networks, Transactions on Emerging Telecommunications Technologies 25 (2014) 1057–1069.

[16] S. Salcedo-Sanz, P. García-Díaz, J. Portilla-Figueras, J. D. Ser, S. Gil-López, A coral reefs optimization algorithm for optimal mobile network deployment with electromagnetic pollution control criterion, Applied Soft Computing 24 (2014) 239 – 248.

[17] S. Salcedo-Sanz, C. Camacho-Gómez, A. Magdaleno, E. Pereira, A. Lorenzana, Structures vibration control via tuned mass dampers using a co-evolution coral reefs optimization algorithm, Journal of Sound and Vibration 393 (2017) 62 – 75.

[18] C. Camacho-Gómez, X. Wang, E. Pereira, I. Díaz, S. Salcedo-Sanz, Active vibration control design using the coral reefs optimization with substrate layer algorithm, Engineering Structures 157 (2018) 14 – 26.

[19] S. Salcedo-Sanz, A. Pastor-Sánchez, L. Prieto, A. Blanco-Aguilera, R. García-Herrera, Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization – extreme learning machine approach, Energy Conversion and Management 87 (2014) 10 – 18.

[20] S. Salcedo-Sanz, C. Camacho-Gómez, R. Mallol-Poyato, S. Jiménez-Fernández, J. DelSer, A novel coral reefs optimization algorithm with substrate layers for optimal battery scheduling optimization in micro-grids, Soft Computing 20 (2016) 4287–4300.

[21] M. Ficco, C. Esposito, F. Palmieri, A. Castiglione, A coral-reefs and game theory-based approach for optimizing elastic cloud resource allocation, Future Generation Computer Systems 78 (2018) 343 – 352.

[22] Z. Yang, T. Zhang, D. Zhang, A novel algorithm with differential evolution and coral reef optimization for extreme learning machine training, Cognitive Neurodynamics 10 (2016) 73–83.

[23] I. G. Medeiros, J. C. Xavier, A. M. P. Canuto, Applying the coral reefs optimization algorithm to clustering problems, in: 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–8.

[24] A. M. Durán-Rosal, P. A. Gutiérrez, S. Salcedo-Sanz, C. Hervás-Martínez, A statistically-driven coral reef optimization algorithm for optimal size reduction of time series, Applied Soft Computing 63 (2018) 139 – 153.

[25] L. Garcia-Hernandez, L. Salas-Morera, J. Garcia-Hernandez, S. Salcedo-Sanz, J. V. de Oliveira, Applying the coral reefs optimization algorithm for solving unequal area facility layout problems, Expert Systems with Applications 138 (2019) 112819.

[26] Komarudin, K. Y. Wong, Applying ant system for solving unequal area facility layout problems, European Journal of Operational Research 202 (2010) 730 – 746.

[27] R. Meller, V. Narayanan, P. Vance, Optimal facility layout design, Operations Research Letters 23 (1999) 117 – 127.

[28] B. Montreuil, A modelling framework for integrating layout design and flow network design, in: Material Handling '90, Springer Berlin Heidelberg, Berlin, Heidelberg, 1991, pp. 95–115.

[29] A. Konak, S. Kulturel-Konak, B. Norman, A. Smith, A new mixed integer formulation for optimal facility layout, Operation Research Letters 34 (2006) 660 – 672.

[30] R. Meller, V. Narayanan, P. Vance, Optimal facility layout design, Operations Research Letters 23 (1998) 117–127.

[31] H. Sherali, B. Fraticelli, R. Meller, Enhanced model formulations for optimal facility layout, Operations Research 51 (2003) 629 – 644.

[32] I. Castillo, J. Westerlund, S. Emet, T. Westerlund, Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods, Computers and Chemical Engineering 30 (2005) 54–69.

[33] J. Chae, A. C. Regan, Layout design problems with heterogeneous area constraints, Computers & Industrial Engineering 102 (2016) 198 – 207.

[34] K. Y. Tam, A simulated annealing algorithm for allocating space to manufacturing cells, International Journal of Production Research 30 (1992) 63–87.

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

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

[37] D. M. Tate, A. E. Smith, Unequal area facility layout using genetic search, IIE Transactions 27 (1995) 465–472.

[38] Y. Wu, E. Appleton, The optimisation of block layout and aisle structure by a genetic algorithm, Comput. Ind. Eng. 41 (2002) 371–387.

[39] A. Gomez, Q. Fernandez, D. D. la Fuente Garcia, P. Garcia, Using genetic algorithms to resolve layout problems in facilities where there are aisles, International Journal of Production Economics 84 (2003) 271 – 282.

[40] M. Enea, G. Galante, E. Panascia, The facility layout problem approached using a fuzzy model and a genetic search, Journal of Intelligent Manufacturing 16 (2005) 303–316. 10.1007/s10845-005-7025-4.

[41] G. Aiello, M. Enea, G. Galante, A multi-objective approach to facility layout problem by genetic search algorithm and electre method, Robotics and Computer-Integrated Manufacturing 22 (2006) 447–455.

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

[43] L. García-Hernández, J. M. Palomo-Romero, L. Salas-Morera, A. Arauzo-Azofra, H. Pierreval, A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem, Expert Systems with Applications 42 (2015) 4697 – 4708.

[44] L. García-Hernández, A. Arauzo-Azofra, L. Salas-Morera, H. Pierreval, E. Corchado, Facility layout design using a multi-objective interactive genetic algorithm to support the DM, Expert Systems 32 (2015) 94–107.

[45] J. M. Palomo-Romero, L. Salas-Morera, L. García-Hernández, An island model genetic algorithm for unequal area facility layout problems, Expert Systems with Applications 68 (2017) 151 – 162.

[46] K. Y. Wong, Komarudin, Solving facility layout problems using flexible bay structure representation and ant system algorithm, Expert Systems with Applications 37 (2010) 5523 – 5527.

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

[48] J. Liu, J. Liu, Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems, Applied Soft Computing 74 (2019) 167 – 189.

[49] B. H. Ulutas, S. Kulturel-Konak, An artificial immune system based algorithm to solve unequal area facility layout problem, Expert Systems with Applications 39 (2012) 5384 – 5395.

[50] A. M. E. Sikaroudi, K. Shahanaghi, Facility layout by collision detection and force exertion heuristics, Journal of Manufacturing Systems 41 (2016) 21 – 30.

[51] F. G. Paes, A. A. Pessoa, T. Vidal, A hybrid genetic algorithm with decomposition phases for the unequal area facility layout problem, European Journal of Operational Research 256 (2017) 742 – 756.

[52] S. Kang, J. Chae, Harmony search for the layout design of an unequal area facility, Expert Systems with Applications 79 (2017) 269 – 281.

[53] E. Shayan, A. Chittilappilly, Genetic algorithm for facilities layout problems based on slicing tree structure, International Journal of Production Research 42 (2004) 4055–4067.

[54] G. Aiello, G. L. Scalia, M. Enea, A non dominated ranking multi objective genetic algorithm and electre method for unequal area facility layout problems, Expert Systems with Applications 40 (2013) 4812 – 4819.

[55] G. Aiello, G. L. Scalia, M. Enea, A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding, Expert Systems with Applications 39 (2012).

[56] K. S. N. Ripon, K. Glette, K. N. Khan, M. Hovin, J. Torresen, Adaptive variable neighborhood search for solving multi-objective facility layout problems with unequal area facilities, Swarm and Evolutionary Computation 8 (2013) 1 – 12.

[57] A. Saraswat, U. Venkatadri, I. Castillo, A framework for multi-objective facility layout design, Computers & Industrial Engineering 90 (2015) 167 – 176.

[58] M. R. A. Purnomo, Y. S. Wiwoho, 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 (2016) 012045.

[59] J. Liu, H. Zhang, K. He, S. Jiang, Multi-objective particle swarm optimization algorithm based on objective space division for the unequal-area facility layout problem, Expert Systems with Applications 102 (2018) 179 – 192.

[60] L. García-Hernández, A. Arauzo-Azofra, L. Salas-Morera, H. Pierreval, E. Corchado, Recycling plants layout design by means of an interactive genetic algorithm, Intelligent Automation & Soft Computing 19 (2013) 457–468.

[61] R. D. Meller, K.-Y. Gau, The facility layout problem: Recent and emerging trends and perspectives, Journal of Manufacturing Systems 15 (1996) 351–366.

[62] F. Azadivar, J. Wang, Facility layout optimization using simulation and genetic algorithms, International Journal of Production Research 38 (2000) 4369–4383.

[63] R. D. Meller, The multi-bay manufacturing facility layout problem, International Journal of Production Research 35 (1997) 1229–1237.

[64] X. Tong, SECOT: A Sequential Construction Technique For Facility Design, Doctoral Dissertation, University of Pittsburg, 1991.

[65] D. E. Goldberg, robert, Alleles, loci, and the traveling salesman problem, in: J. J. Grefenstette (Ed.), Proceedings of the First International

Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates, Publishers, 1985, pp. 154–159.

[66] A. Eiben, J. Smith, Introduction to Evolutionary Computing, SpringerVerlag, 2003.

[67] J. C. Dunn, A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters, Journal of Cybernetics 3 (1973) 32–57.

[68] L. Salas-Morera, A. Cubero-Atienza, R. Ayuso-Munoz, Computer-aided plant layout — Distribucion en planta asistida por ordenador, Informacion Tecnologica 7 (1996) 39–46.

[69] S. Salcedo-Sanz, A review on the coral reefs optimization algorithm: new development lines and current applications, Progress in Artificial Intelligence 6 (2017) 1–15.

Authors' individual contributions

Manuscript title: Applying the Coral Reefs Optimization Algorithm for solving Unequal Area Facility Layout Problems

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: Ajith Abraham

x Conceptualization

☐ Data curation

x Formal analysis

☐ 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

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