applied sciences

MDPI

Article

A Multi-User Interactive Coral Reef Optimization Algorithm for Considering Expert Knowledge in the Unequal Area Facility Layout Problem

Lorenzo Salas-Morera *, **Laura García-Hernández** and **Carlos Carmona-Muñoz**

Area of Project Engineering, University of Cordoba, 14071 Córdoba, Spain; ir1gahel@uco.es (L.G.-H.); i32camuc@uco.es (C.C.-M.)
* Correspondence: lsalas@uco.es

Abstract: The problem of Unequal Area Facility Layout Planning (UA-FLP) has been addressed by a large number of approaches considering a set of quantitative criteria. Moreover, more recently, the personal qualitative preferences of an expert designer or decision-maker (DM) have been taken into account too. This article deals with capturing more than a single DM's personal preferences to obtain a common and collaborative design including the whole set of preferences from all the DMs to obtain more complex, complete, and realistic solutions. To the best of our knowledge, this is the first time that the preferences of more than one expert designer have been considered in the UA-FLP. The new strategy has been implemented on a Coral Reef Optimization (CRO) algorithm using two techniques to acquire the DMs' evaluations. The first one demands the simultaneous presence of all the DMs, while the second one does not. Both techniques have been tested over three well-known problem instances taken from the literature and the results show that it is possible to obtain sufficient designs capturing all the DMs' personal preferences and maintaining low values of the quantitative fitness function.

Keywords: UA-FLP; coral reef optimization; interactive algorithm; multi-user; meta-heuristics

check for updates

Citation: Salas-Morera, L.; García-Hernández, L.; Carmona-Muñoz, C. A Multi-User Interactive Coral Reef Optimization Algorithm for Considering Expert Knowledge in the Unequal Area Facility Layout Problem. *Appl. Sci.* **2021**, *11*, 6676. https://doi.org/10.3390/app11156676

Academic Editor: Giancarlo Mauri

Received: 2 June 2021
Accepted: 17 July 2021
Published: 21 July 2021

1. Introduction

The rapid technological change that the contemporary world is undergoing poses a series of opportunities and challenges for companies and for the productive structures of countries. Furthermore, the reduction of costs is a determining factor in the success and profitability of companies, so it has become one of the most important challenges. In this sense, an optimal plant layout design can affect the manufacturing efficiency, with savings from 20% to 50% in the total production costs of industrial companies [1]. Therefore, a poor or irregular plant layout design can lead to longer waiting times, higher material handling costs (MHC), and reduced worker efficiency [2]. For these reasons, the facility layout design is one important issue in manufacturing efficiency and industry competitiveness. In general, the facility layout problem (FLP) deals with the determination of facilities' placement within a known rectangular plant floor so as to satisfy certain given objectives and restrictions and it is considered to be NP-hard [3,4]. The most commonly used problem configuration is to find the best layout distribution with Unequal Area Facility Layout Planning (UA-FLP) [5], since considering all the facilities with the same area (EA-FLP) is thought to be an unfeasible simplification in practice [6]. Generally, a set of objectives can be taken into account, such as material handling costs, closeness relationships or distance requirements, among others, but it is common to consider only the former [7], as well as a set of constraints, such as the total available area, the area needed by each of the facilities, the restriction that overlapping facilities are not allowed, and the highest allowed ratio between each facility's largest and shortest side, which is called the aspect ratio [8].

Moreover, in recent years, qualitative aspects have been taken into account too [9–11] with the aim of incorporating expert human knowledge into the design [12]. These qualitative aspects can be related to the relative position between facilities, aesthetic aspects, or personal subjective preferences of the designers, based on their personal background and experience, and can be previously known or simply appear during the algorithm evolution process, or they can even change during the execution time [13]. In the same way, more recently, the intervention of several human designers has been introduced in some research fields. For example, Refs. [14,15] used the combined intervention of several evaluators who judged two solutions and chose the best of them in a sort of tournament competition, obtaining a good representation of evaluators' opinions in the algorithms' results. Similarly, Refs. [16,17] used a distributed interactive genetic algorithm to incorporate the feelings of several users into a musical melody. Each user evolved a population of the algorithm and, after a certain number of generations, an exchange of solutions was done, with the result that the final melody satisfied all users' preferences. In these kinds of designs, it is necessary to reflect on the intervention of multiple designers or evaluators. In the first one, a voting system must be implemented in order to collect users' opinions and to obtain a compromise evaluation of each solution, in such a way that only the best one is selected. In this case, the simultaneous presence of all the evaluators is needed, even when the evaluation is done via the web. In the second one, a model similar to an island algorithm [18] is put into practice. In this model, the concurrent presence of each of the evaluators is needed too.

Concerning the UA-FLP problem, many approaches have been used in order to solve it. Initially, some exact methods were devised and carried out, but the NP-hard character of the problem [19] makes it unapproachable for large problems in terms of computational cost. Nevertheless, Ref. [20] proposed an integer programming method, limited to problems of up to five departments, and [21] improved the proposal to solve instances of up to eight departments. By their part, Refs. [22,23] presented a mixed integer programming formulation limited to problems with up to fourteen departments, and, more recently, Ref. [24] found optimal solutions for problems with up to twelve departments. Given the aforementioned issues, heuristic and meta-heuristic approaches have arisen. Among them, Ref. [25] suggested a multiobjective genetic algorithm combining material handling costs, adjacency, and distance between departments; meanwhile, ref. [26,27] adopted Ant Colony Optimization Algorithms, obtaining good results on well-known problem sets. In the task of improving the algorithm's efficiency, island optimization algorithms have been implemented too [18,28]. These kinds of algorithms have in common an improvement of the exploration of the searching space by means of maintaining the diversity of the individuals in the population, which allows the achievement of good results within a reasonable computational time. Other remarkable approaches to the problem have been Simulated Annealing [29–32], Tabu Search [33–35], Wang–Landau sampling algorithms [36], and Particle Swarm Optimization [37–39], among others. More recently, Coral Reef Optimization (CRO) algorithms have been used to improve the solutions obtained. These kinds of algorithms can be classified as bio-inspired evolutionary algorithms and they simulate the processes occurring in natural coral reefs, such as reproduction, fighting for space, and depredation, which results in a sort of combination of Simulating Annealing and a Hybrid Evolutionary Algorithm [40]. The CRO has been successfully applied to a number of areas, such as the analysis of time series [41], optimization of antennas for LTE and 5G services [42], selection of representative measuring points for temperature field reconstruction [43], data clustering [44], Wi-Fi channel assignment [45], materialized view selection in a data warehouse [46], and optimization of convolutional neural networks [47], among others.

Looking at the application of CRO to the UA-FLP, it has been already used successfully in several variants. Firstly, García-Hernández et al. [48] put into practice a basic CRO algorithm applied to the UA-FLP, improving the solutions previously found in a well-known set of problems. This basic CRO was subsequently improved by means of an island evolution model [49] and a combined evolution model called a Substrate Layers model CRO (CRO-SL) [50]. Recently, a multi-objective interactive CRO algorithm has been developed to take into account the subjective preferences of an expert designer [10]. Nevertheless, to the best of our knowledge, the joint collaboration of several human designers has not been implemented yet over the UA-FLP. The aim of the multiple users' evaluation is to capture a diversity of criteria that enrich the design from several points of view derived from the designers' expert knowledge.

Therefore, this paper deals with the collaborative incorporation of several expert designers' knowledge into the design of a plant layout. This is the first time that multiple users' collaboration has been applied to the problem of UA-FLP.

The rest of the paper is organized as follows: Section 2 details the problem formulation. Section 3 describes the novel multi-user approach. Section 4 explains the set of experiments performed to validate the approach, presenting and analyzing the results obtained. Finally, in Section 5, the main conclusions and future lines of work are given.

2. Problem Formulation

2.1. Fitness Function

The UA-FLP deals with the arrangement of n rectangular facilities, each one of a given area A_i within an available space of dimensions $W \times H$ (Equation (1)) and without overlapping between them.

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

Furthermore, there is one constraint related to the geometrical proportions of each of the facilities. In this sense, the parameter *aspect ratio* is defined as the quotient of the largest side of a facility by the shortest one, and it can be a general restriction that is equal for all the plant's facilities or a particular limit for each one of them.

With these restrictions, the parameter to be optimized (minimized) is the material handling cost between facilities, given the distance between the centroids of every two facilities d_{ij}, and the material flow between them (units of product per unit of time, kg per unit of time, etc.), f_{ij}. The problem consists of finding the disposition of the n facilities within the plant while respecting the restrictions and minimizing the total material handling cost (Equation (2)).

$$minimize \sum_{i=1}^{n} \sum_{j \neq i}^{n} f_{ij} \times d_{ij} \tag{2}$$

The general fitness function must consider the minimization of the material handling cost, as shown in Equation (2), and penalizing candidate solutions that do not respect the aspect ratio constraints. On the contrary, if the fitness function only considers the minimization of the material handling costs, the search could converge quickly in unfeasible solutions from the point of view of the aspect ratio of one or more facilities. To avoid this problem, Tate and Smith [51] proposed the use of a penalization for the solutions that contain these unfeasible facilities, adding a penalty value proportional to the number of unfeasible facilities (if any). In this way, the fitness function finally considered is shown in Equation (3).

$$V_t = \sum_{i=1}^{n} \sum_{j \neq i}^{n} f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all}) \tag{3}$$

where:

V_t is the value of the fitness function with penalty for a specific plant.

f_{ij} and d_{ij} are the material flows between departments and the distances between centroids, respectively.

D_{inf} is the number of unfeasible facilities.

k is a penalty parameter that adjusts the grade of penalization (set to 3, according to Tate and Smith [51]).

V_{feas} is the best fitness value found in the set of feasible solutions.
V_{all} is the overall best fitness value found.

Note that in the case of $D_{inf} = 0$, the value of the fitness function is simply the material handling cost.

Nevertheless, since the collaborative intervention of several expert designers (or decision-makers, DMs) has to be incorporated into the evaluation, the fitness function must be modified [10]. The expert designers' evaluation is a discrete value on a Likert scale from 1 (not satisfactory) to 5 (very satisfactory). To fulfill this new requirement, an extra penalty function has been added (Equation (4)):

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

where:

U_t is the factor for the extra penalty.
x is the evaluation assigned by the expert.
n is the plant's number of facilities.

Therefore, if the DM assigns a score of 5 (very satisfactory), the extra penalty factor is 0, and in the other extreme, if the DM assigns a valuation of 1 (not satisfactory), the extra penalty factor is n. The final fitness function to be applied is shown in Equation (5):

$$V_t = (1 + (U_t)^k) \sum_i^n \sum_j^n f_{ij} d_{ij} + (D_{inf})^k (V_{feas} - V_{all}) \tag{5}$$

In this way, if the evaluation assigned by the DM is 5 and the number of unfeasible facilities is 0, the value obtained by the plant is equal to the material handling cost; moreover, if the evaluation assigned by the DM is 1, the material handling cost is multiplied by $1 + n$. Finally, if Equation (5) is minimized, the solution found would satisfy, in the best possible way, the requirements that the material handling cost is minimum, the aspect ratio constraints are fulfilled, and the personal preferences of the DM are satisfied.

2.2. Evolutionary Strategy

A Multi-user Interactive Coral Reef Optimization Algorithm (MICRO) is suggested in order to solve the UA-FLP with the incorporation of several human experts' (users) preferences. The basic CRO was introduced by Salcedo-Sanz et al. [40] and its application to UA-FLP was described previously in detail by García-Hernández et al. [48]. In brief, it consists of the disposition of a number of evaluated individuals (candidate solutions to the problem) occupying a fraction ρ_0 of the available space in a rectangular matrix of dimensions $N \times M$ called *the reef* (phase I: initialization). Then, the evolution starts (phase II: evolution). In this phase, a reproduction of individuals from those already settled in the reef (corals) is carried out by means of two methods: broadcast spawning (similar to the crossover in genetic algorithms) and brooding (similar to mutation in genetic algorithms). Note that all corals reproduce by either of these methods, leading to two disjoint sets whose proportion is determined by f_b: the fraction of corals that will reproduce themselves by spawning. Afterwards, each one of the new individuals that appeared during the reproduction currently in the water (larvae) tries to take a place in the reef in a position chosen randomly (phase III: larvae setting), finding two possible situations. In the first case, the position is empty, so the larva settles freely on it; in the second one, the spot is occupied, so if the new larvae has better fitness than the one already occupying the spot, this last one is replaced; if this is not the case, it remains *in the water* searching for another place up to three times. If after these attempts, the larva does not find a place in the reef, it is discarded. Finally, a fraction f_a of the individuals with better fitness values are duplicated and try to find a position in the reef in the same way formerly described (phase IV: asexual reproduction), and a fraction f_d of the worst individuals are candidates to be discarded with a probability p_d (phase V: predation). The process is repeated until a stop condition is met (number of iterations, no improvement for x iterations, etc.). See Figure 1 for a graphic representation of each of the previously described phases, and Algorithm 1 for the algorithm's pseudocode. The method has been programmed in Python 3.

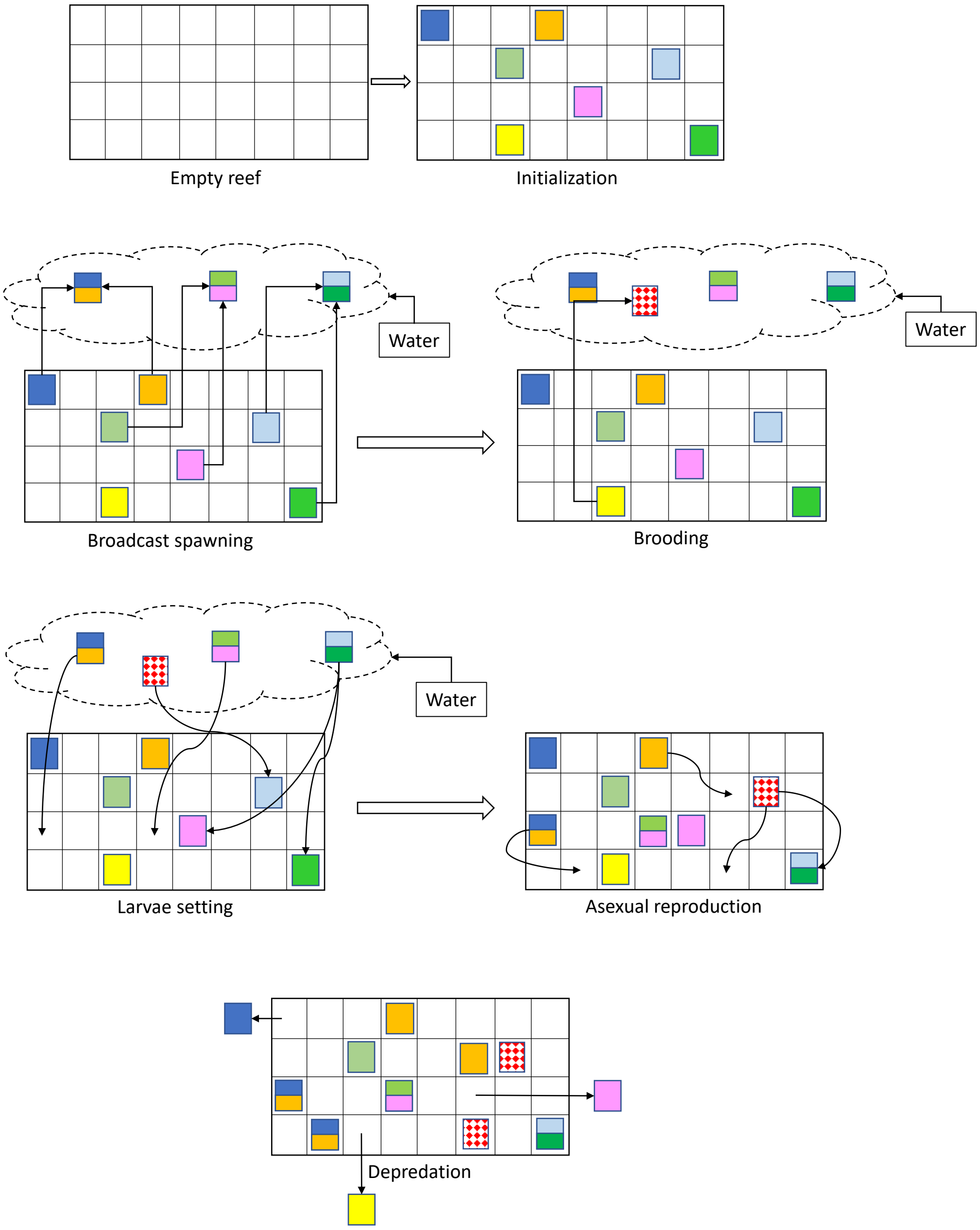


Figure 1. Phases of the basic CRO algorithm.

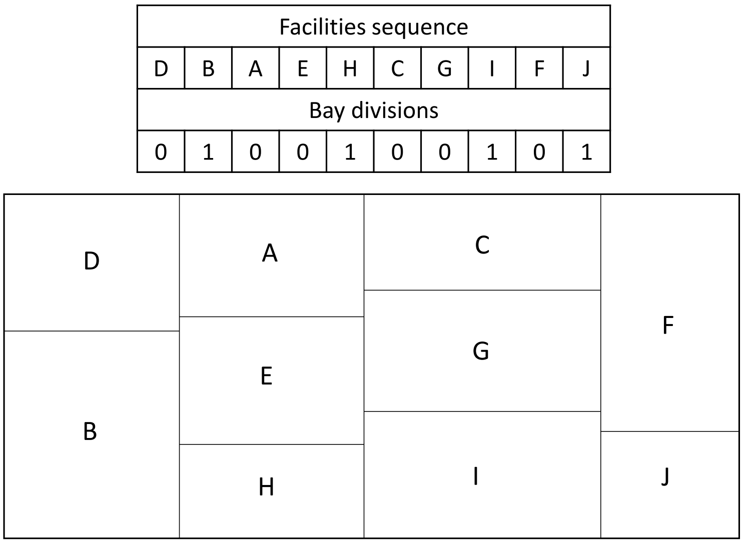
Algorithm 1 CRO algorithm pseudocode

Input Reef dimensions $m \times n$, Initial occupation rate ρ_0, Fraction of broadcast spawning f_b, Fraction of asexual reproduction f_a, Predation fraction f_d, Predation probability p_d
Output Solution with best *fitness*

```
procedure CRO
    $reef \leftarrow$ initialization$(n, m, \rho_0)$
    repeat
        $spawning\_set, brooding\_set \leftarrow$ partition$(reef, f_b)$
        $water \leftarrow$ broadcast_spawning$(spawning\_set)$
        $water \leftarrow \{water,$ brooding$(brooding\_set)\}$
        $water \leftarrow$ evaluation$(water)$
        $reef \leftarrow$ larvae_setting$(water, reef)$
        $reef \leftarrow$ asexual reproduction$(reef, f_a)$
        $reef \leftarrow$ predation$(reef, f_d, p_d)$
    until $stop\_condition$
    return best_solution$(reef)$
end procedure
```

2.3. Individual Codification

Individuals in the reef are codified following a bay structure according to Gomez et al. [52], using a two-vector structure. The first one contains the sequence in which facilities are placed in the plant, from top to bottom and from left to right. The second vector contains a sequence of as much 0 and 1 as the number of facilities in the plant. The positions with 1 mark the end of each bay (see Figure 2).



Facilities sequence									
D	B	A	E	H	C	G	I	F	J
Bay divisions									
0	1	0	0	1	0	0	1	0	1

Figure 2. Facility layout codification example.

2.4. Multi-User Evaluation

The fitness function stated in Equation (5) has two clearly different parts: a quantitative evaluation of solutions given by material handling costs and geometrical feasibility, and a qualitative evaluation assigned by the DM. Therefore, it would be necessary that the DM would evaluate each one of the candidate solutions in the whole reef and water but, since the number of individuals to be evaluated can be high, human intervention is critically limited due to the necessary time to perform it and by the fatigue caused to the DM. Hence, it is necessary to implement a strategy to free the DM from such a tiresome task. In this way, a clustering method has been implemented in order to show to the DM only a small group formed by nine solutions, distributing the valuation to the rest of the individuals proportionally to their similarity to the main nine that have been actually assessed by the DM. The clustering algorithm used is Fuzzy c-Means [53] and it is explained in detail in García-Hernández et al. [54]. This work has used the implementation of c-Means developed in the library Peach.

The strategy followed to organize the human intervention during the algorithm execution is as follows: during the first generations, the intervention of the DMs is required in every one of them until one solution that satisfies completely the DM's requirements appears. Once this occurs, the algorithm runs by itself, without the intervention of the DMs, for a prefixed number of iterations (this number can be set for each problem and can be changed during the execution time) in which human evaluations are automatically assigned according to the previously assigned ones. Therefore, the DM's fatigue is limited as much as possible without losing his/her contribution to guide the process. Furthermore, the intervention of several DMs who have their own preferences regarding the characteristics of the problem is allowed. Thus, every time that the intervention of the DM is required during the algorithm process, it is possible to choose a different one. In this way, it is possible to organize the intervention of 1 to m human evaluators sequentially one by one, changing the DM each time that the algorithm requires the human intervention, or to allow a particular DM to guide the evolution of the algorithm during several generations in a row, or even to allow the intervention of the m human evaluators in a random way (see flow diagram in Figure 3).

According to the flow diagram of Figure 3, the algorithm begins with a random generation of individuals representing the whole population. Afterwards, the individuals of the population are grouped into nine clusters according to García-Hernández et al. [54] and the human intervention can be required or not. The human intervention is required for every generation in which no individuals have appeared satisfying the DMs' requirements, and every certain number of prefixed generations if any individual satisfying the DMs' requirements has already appeared during the previous generations. Furthermore, every time that the human intervention is required, the algorithm asks whether the same DM will be maintained or a change will take place and the DM who will perform the evaluation is indicated if there is a change. Continuing with the development of the algorithm, the quantitative values of every one of the individuals in the population are calculated and incorporated into the general evaluation function. Finally, the reproduction and selection processes are performed, and the algorithm finishes, or the depredation is carried out and a new generation starts.

The way in which the scores assigned by each one of the DMs are distributed among the corals is as follows: in the case where there is a single evaluator, only the nine individuals representative of the clusters (centers) are actually evaluated, and the scores are distributed to the rest of the population proportionally to their similitude with respect to the nine centers, as mentioned before. Nevertheless, in the general case where there is more than one DM, it is possible that each one of the centers of the clusters has been evaluated by one or up to m of the DMs. Therefore, each one of the centers is assigned a vector of evaluations with the same number of positions as the number of DMs, and where at least one of the positions is not 0. The valuation distributed to the rest of the population is the average of the non-0 scores. In this way, the valuation distributed to the rest of the population contains the preferences of all the DMs that have actually evaluated each one of the centers at any moment in the algorithm's evolution. This strategy helps to consider the preferences of all the DMs present in the system even though their intervention is not synchronous.

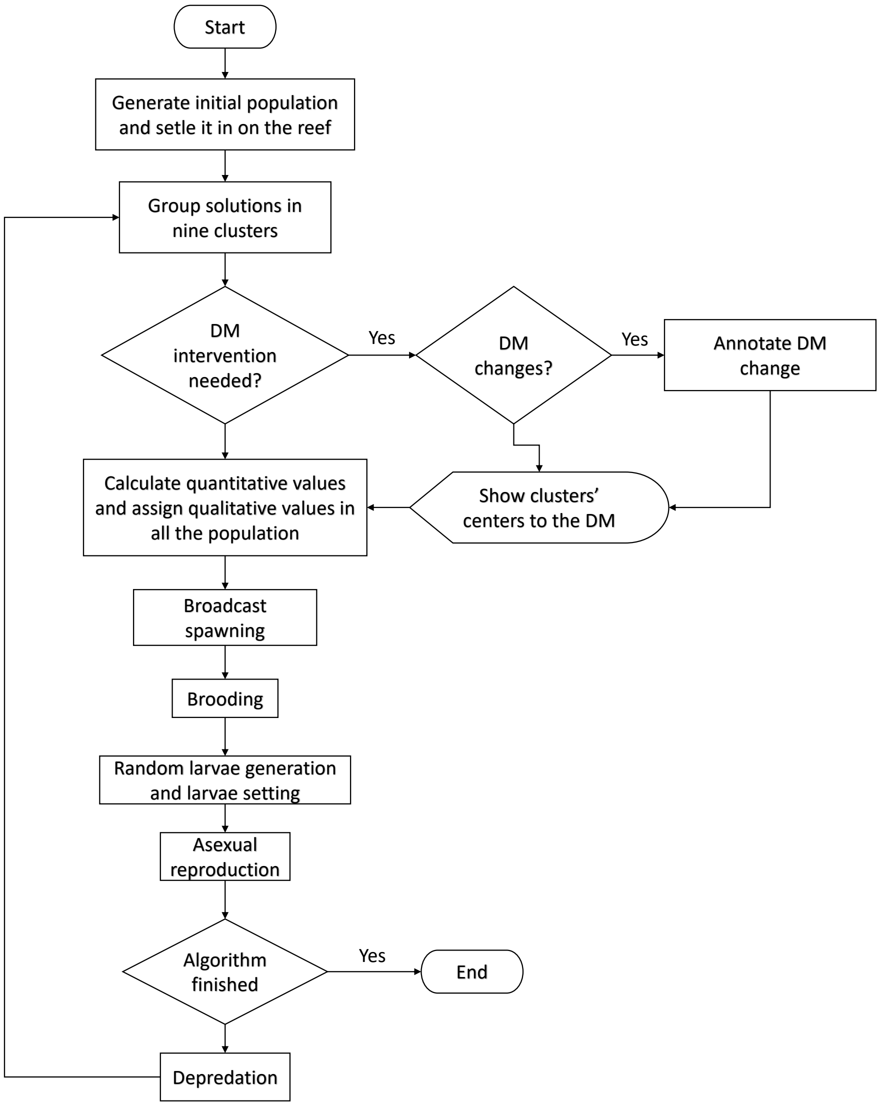

Start
Generate initial population and setle it in on the reef
Group solutions in nine clusters
DM intervention needed?
Yes
DM changes?
Yes
Annotate DM change
Show clusters' centers to the DM
Calculate quantitative values and assign qualitative values in all the population
Broadcast spawning
Brooding
Random larvae generation and larvae setting
Asexual reproduction
Algorithm finished
Yes
End
Depredation

Figure 3. Flow diagram of the algorithm with DMs' intervention.

3. Experimentation

The performance of the suggested multi-user interactive CRO is evaluated in this section. Although it is difficult to compare the results with other approaches since this is the first time that a multi-user strategy has been implemented in UA-FLP, the results obtained have been compared with those previously obtained in a single-user strategy [10]. The three problem instances chosen for evaluating the algorithm's performance are the following ones: *Slaughterhouse*, proposed in Salas-Morera et al. [55] and explained in detail in García-Hernández et al. [56]; *CartonPacks*, proposed in García-Hernández et al. [56]; and *ChoppedPlastic*, from García-Hernández et al. [13].

The tests' strategy set up to analyze the algorithm performance is as follows: the number of different DMs chosen for the tests was three. Two sets of tests have been carried out with the three instances, one of them requiring the alternative intervention of each one of the DMs, in such a way that DM1 assessed the generations $3g + 1$; DM2, the generations $3g + 2$; and the DM3, the generations $3g + 3$, with g taking values from 0 to 32, which led to a total of 99 generations of the algorithm. However, it was not necessary for each of the DMs to evaluate 33 generations, since, from the moment that one individual appears satisfying all of one DM's preferences, the algorithm is configured in a way that only requires this DM's intervention from some generations to some generations, as mentioned before (this strategy will be called *alternative*). The second one (called *sequential*) requires the intervention of the three DMs in a sequential way. Thus, DM1 evaluates the centers as many times as required by the algorithm until it reaches generation 33, DM2 does so from generation 34 to 66, and DM3 from generation 67 to 99. The tests' strategy is configured in this way to determine if the order of intervention of the DMs is significant in the result or not, which could lead to the conclusion that the order of the DMs' intervention could be even random. In the same way, the *alternative* strategy requires the coordinated intervention of the DMs, which demands the simultaneous presence of all of them, while the *sequential* one permits the separate intervention of each one of them. Each set of proofs has been repeated five times.

The parameters of the algorithm have been configured as shown in Table 1.

Table 1. Chosen MICRO 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 → 5

The first problem, *Slaughterhouse*, consists of 12 facilities that have to be set up in a plant of 30 × 51.14 m^2 with an aspect ratio per facility limited to 4. The characteristics of this problem are summarized in Table 2 and Figure 4, and the particular preferences of each one of the DMs are shown in Table 3. The reasons that the DMs wish for the two facilities to either be far from each other or close to each other, to be in the perimeter or in the interior part of the plant, can be diverse. In this case, DM1 considers that A (*Stables*) must be in the perimeter as this is part where new animals enter the plant and it needs a connection with the exterior part to unload. Analogously, DM1 wants L (*Byproduct shipping*) and J (*Shipping*) to be in the perimeter for the same reasons. These reasons could be classified together as *process*. By his/her part, DM2 thinks that K (*Offices*) must be far from A (*Stables*). In this case, the reason is likely due to noise and the bad smell. Equally, he/she thinks that K must be far from I (*Compressor room*), due to machinery noise. DM2 thinks too that J (*Shipping*) and L (*Byproduct shipping*) must be far from each other due to marketing reasons. Finally, DM3 considers that B (*Slaughter*) must be far from J (*Shipping*) due to hygiene reasons; that I (*Compressor room*) must be far from A (*Stables*) due to the machinery's noise, which could disturb the animals; and that H (*Boiler room*) must be close to C (*Entrails*) due to process reasons.

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 and 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

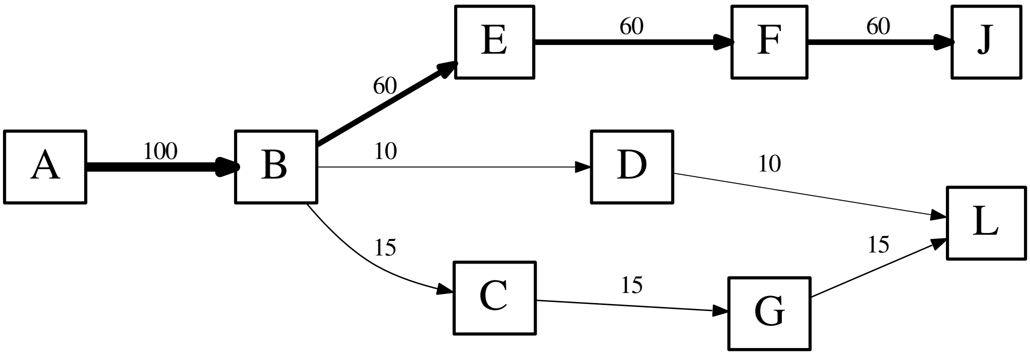


Figure 4. Material flow requirements for the *Slaughterhouse* problem.

Table 3. DMs' particular preferences for *Slaughterhouse* problem.

DM	Preference 1	Preference 2	Preference 3
DM1	A must be in the perimeter of the plant	L must be in the perimeter of the plant	J must be in the perimeter of the plant
DM2	K must be far from A	K must be far from I	L must be far from J
DM3	B must be far from J	I must be far from A	H must be close to C

The second instance, *CartonPacks*, is a carton recycling plant, and has 11 departments with an available area of 20 × 14.5 m^2, with an aspect ratio limit of 4 per facility. Facilities' areas are presented in Table 4 and the flows between them appear in Figure 5. The preferences of DMs are shown in Table 5. DM1's preferences indicate that facilities A (*Raw material*), F (*Expedition*), and D (*Offices*) must be in the perimeter due to process reasons. DM2 prefers that D (*Offices*) must be far from C (*Repair shop*) as this can be a noisy activity and close to A and F (*Raw material and Expedition*) due to process reasons; and DM3 points to noise reasons in asking that D (*Offices*) be far from G, H, and I (*Hydraulic 1, Hydraulic 2, and Crushing*).

Table 4. Facility features for the *CartonPacks* problem.

Id	Facility	Area (m^2)	Aspect Ratio
A	Raw Material	40	4
B	Finished products	40	4
C	Repair shop	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

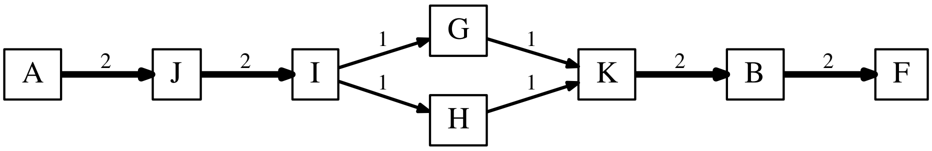


Figure 5. Material flow requirements for the *CartonPacks* problem.

Table 5. DMs' particular preferences for *CartonPacks* problem.

DM	Preference 1	Preference 2	Preference 3
DM1	A must be in the perimeter of the plant	F must be in the perimeter of the plant	D must be in the perimeter of the plant
DM2	D must be far from C	D must be close to A	D must be close to F
DM3	D must be far from G	D must be far from H	D must be far from I

The third problem, *ChoppedPlastic*, refers to a plant of plastic recycling and consists of 10 facilities to be settled in an area of 30 × 10 m^2, with an empty area of 21 m^2. The facilities' areas and aspect ratio limits appear in Table 6, and the flows between facilities are shown in Figure 6. The DMs' preferences are listed in Table 7. In this case, DM1 considers that facility I (*Office*) must be in one end of the plant to protect it from the possible annoying activity of the rest of the plant, the facility K (*Repair shop*) must be in the perimeter as it needs a loading dock for receiving replacement parts, and J (*Toilets*) must be close to I (*Offices*) due to personal commodities. DM2 wants K (*Repair shop*) to be close to E (*Chopped*) for process reasons, I (*Office*) to be close to G (*Expedition*) due to logistics, and G to be at one end of the plant for process reasons. Finally, DM3 considers that the empty space (Z) must be at one end for aesthetic reasons, as well as close to the *Offices* in order to allow possible future expansions, and that E (*Chopped*) must be far from I (*Offices*) for personal comfort.

Table 6. 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 and 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
Z	Empty space	21	

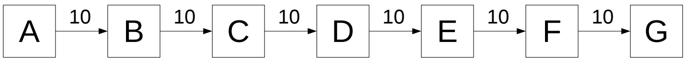


Figure 6. Material flow requirements for the *ChoppedPlastic* problem.

Table 7. DMs' particular preferences for *ChoppedPlastic* problem.

DM	Preference 1	Preference 2	Preference 3
DM1	I must be in the end of the plant	K must be in the perimeter of the plant	J must be close to I
DM2	K must be close to E	I must be close to G	G must be in the end of the plant
DM3	Z must be in the end of the plant	Z must be close to I	E must be far from I

4. Results

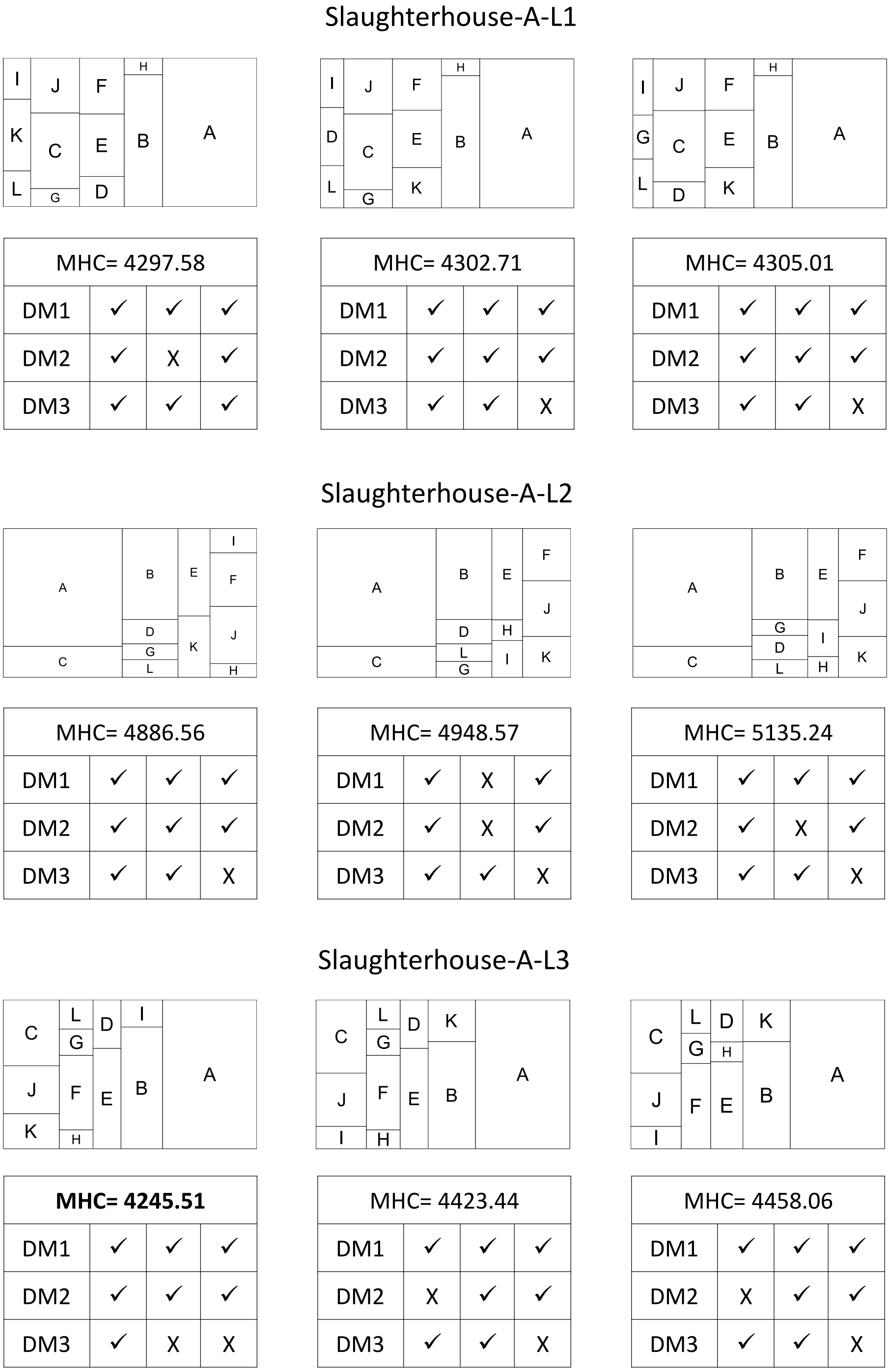
This section summarizes the results obtained in the tests carried out to prove the performance of the algorithm. Figures 7–18 show the three best solutions found per launch of 100 generations, giving the fitness function value of each one of the solutions (coinciding with MHC) as well as the fulfillment of DMs' subjective preferences. Solutions with the minimum value of the fitness function and fulfilling most of the DM preferences are shown in bold. The best solutions shown in Figures 7–18 are summarized in Table 8. In all the cases tested, good designs were found considering the material handling cost and introducing the subjective preferences of all three DMs. It is worth mentioning the fact that new constraints have been introduced without needing to formulate them in a formal way and that the designs found do not only satisfy these quantitative criteria but obtain low enough values of material handling cost. Thus, this interactive approach introduces the possibility of making the design much more realistic and applicable to real engineering problems without neglecting the need to minimize quantitative costs.

With respect to the comparison of results with the previous single-user approach, it is worth mentioning that, in this case, the subjective preferences of three different DMs have been captured. The goal of introducing more than one human DM is to enrich the design with the opinions of a team of designers that can bring different points of view to the project, giving it a more complex and complete perspective. Obviously, giving the design a greater set of qualitative characteristics has the risk of sacrificing part of the quantitative optimization, and the designing team will have to assess if this is acceptable or not, but in real industrial projects, it is possible that the exact result of the quantitative optimization cannot be directly applied due to safety, hygiene, or other factors that are impossible to consider in a simple fitness function. Nonetheless, in the three cases tested, the loss of quantitative optimization seems not to be too significant, and in one of them, improvements have been found with respect to the best results previously known. Concretely, the best solution found in [10] for the Slaughterhouse problem had an MHC of 4136.46, while, according to Figures 7–18, considering the preferences of the three designers, a solution with 4245.51 was found, fulfilling 7 of the 9 requirements, and other one with 4574.24, fulfilling all nine requirements. In the same way, the best solution found in [10] for the CartonPacks problem had an MHC of 59.52, while here, a solution fulfilling 7 of the 9 requirements with a MHC of 57.29 has been found, and other one fulfilling all nine requirements and with an MHC of 68.19. Finally, for the ChoppedPlastic problem, the best solution found in [10] had an MHC of 257.94, while here, a solution fulfilling 7 of the 9 requirements has been found with an MHC of 269, and other one fulfilling all nine requirements with an MHC of 332.3.

Furthermore, the most interesting issue of this paper is the collaborative intervention of more than one human designer to obtain a more complex and complete design. This possibility represents a substantial advance in this kind of problem since it has not been achieved before. The test strategy followed attempted to check whether the order in which the intervention of each one of the human designers is required affects the results or not. Therefore, a double method of intervention was used. Firstly, a more simple strategy was implemented, requiring the alternative intervention of the three human designers. This strategy requires the simultaneous presence of the three (m, in a general way) human designers, in a way that could be fatiguing or, at least, annoying, since all the human designers have to be present in the same space simultaneously, while only one of them is actually working and the others are waiting. The second strategy does not require the simultaneous presence of the three designers, since each one of them guides the third part of the generations of the algorithm in a row, but this strategy could lead to the problem that the algorithm might *forget* the preferences of the previous designer while following the current designer's preferences. Therefore, a set of tests has been performed to assess this, and the results are promising since there is no evidence that the order in which the designers perform their evaluations affects the results in any way. As shown in Table 8, in all cases, good solutions fulfilling the DMs' requirements have been found, sacrificing only a minor part of the material handling costs.

Table 8. Summary of best layouts found applying only quantitative criteria and including qualitative criteria.

Problem/ DMs Intervention	Best MHC	Layout	MHC of Best Qualitative	Layout
Slaughterhouse/ Alternative	4245.51	CJK | LGFH | DE | IB | A	4604.71	IJFK | EDL | BHG | AC
Slaughterhouse/ Sequential	4297.58	A | HB | FED | JCG | IKL	4574.24	KLGJ | CF | IHE | DB | A
Cartonpacks/ Alternative	57.29	DC | EF | HKB | GIJA	60.51	FC | BKHG | AJI | DE
Cartonpacks/ Sequential	61.59	DC | EAJ | FHI | DKG	68.19	AE | IJD | GF | CHKB
ChoppedPlastic/ Alternative	299	A | BC | JD | E | F | G | ZKI	334.7	A | B | CD | KE | JF | ZIG
ChoppedPlastic/ Sequential	269	G | F | E | KD | JC | B | A | I | Z	332.3	BA | CD | KE | JF | IG | Z



Slaughterhouse-A-L1

MHC= 4297.58			
DM1	✓	✓	✓
DM2	✓	X	✓
DM3	✓	✓	✓

MHC= 4302.71			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	X

MHC= 4305.01			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	X

Slaughterhouse-A-L2

MHC= 4886.56			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	X

MHC= 4948.57			
DM1	✓	X	✓
DM2	✓	X	✓
DM3	✓	✓	X

MHC= 5135.24			
DM1	✓	✓	✓
DM2	✓	X	✓
DM3	✓	✓	X

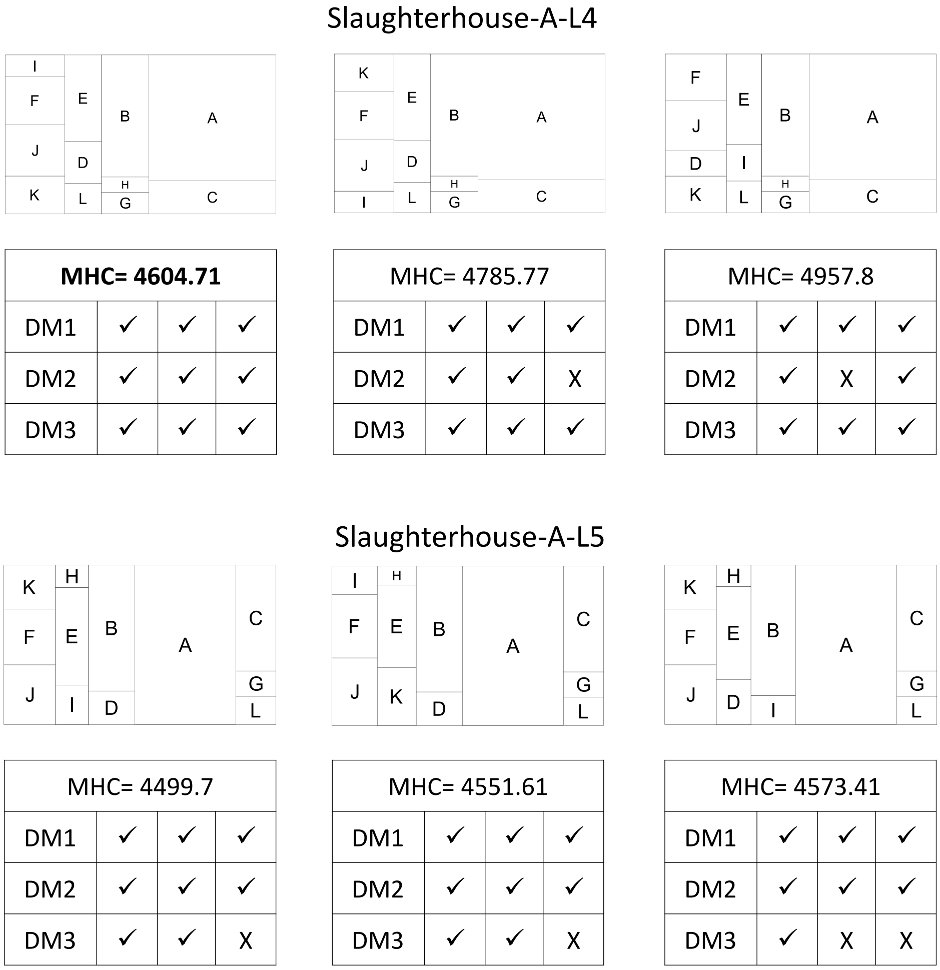
Slaughterhouse-A-L3

MHC= 4245.51			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	X	X

MHC= 4423.44			
DM1	✓	✓	✓
DM2	X	✓	✓
DM3	✓	✓	X

MHC= 4458.06			
DM1	✓	✓	✓
DM2	X	✓	✓
DM3	✓	✓	X

Figure 7. Best solutions found in *Alternative* launch 1 to 3 of *Slaughterhouse* problem.


Slaughterhouse-A-L4

MHC= 4604.71			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	✓

MHC= 4785.77			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

MHC= 4957.8			
DM1	✓	✓	✓
DM2	✓	X	✓
DM3	✓	✓	✓

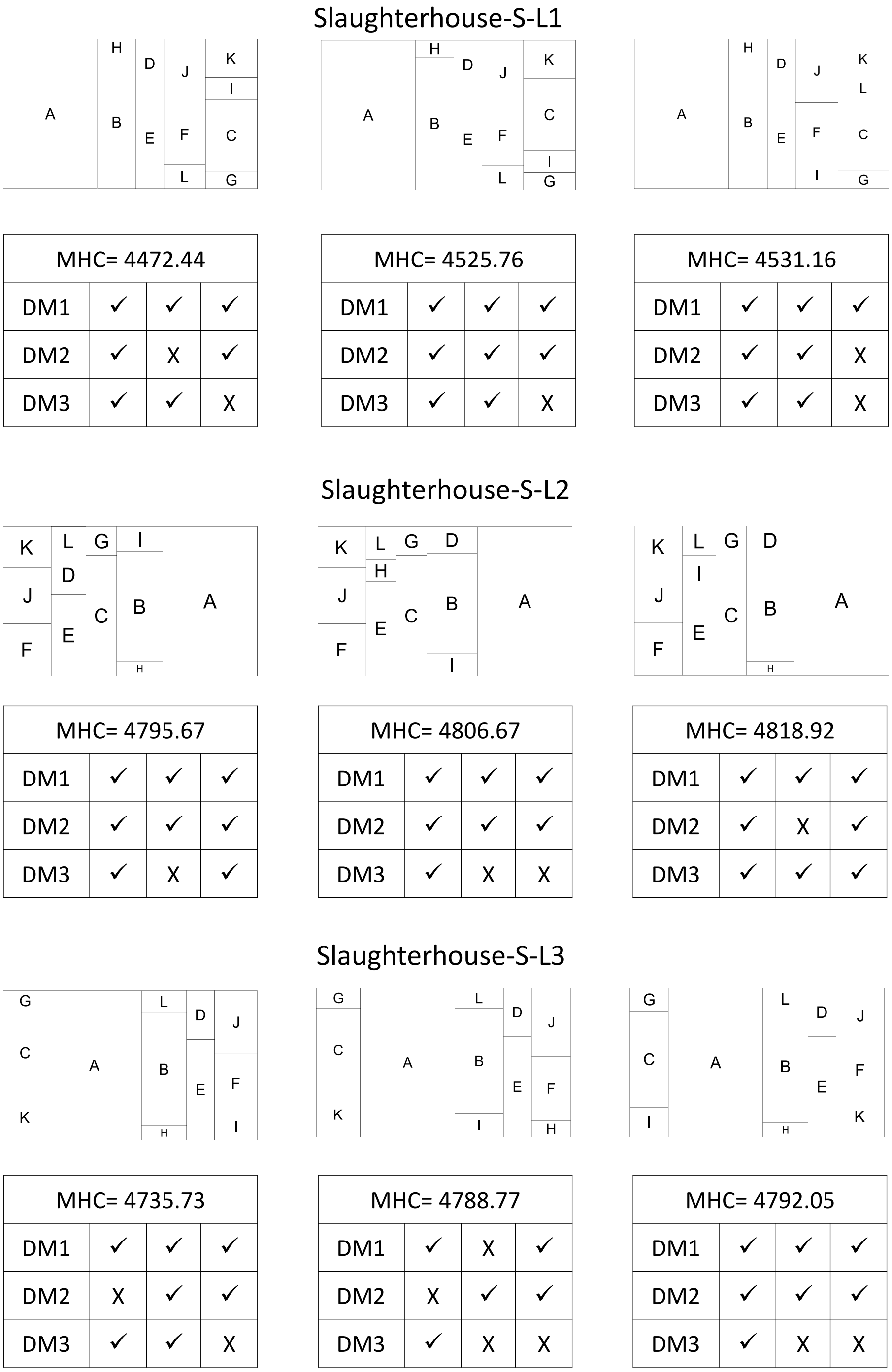
Slaughterhouse-A-L5

MHC= 4499.7			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	X

MHC= 4551.61			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	X

MHC= 4573.41			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	X	X

Figure 8. Best solutions found in *Alternative* launch 4 to 5 of *Slaughterhouse* problem.



Slaughterhouse-S-L1

MHC= 4472.44			
DM1	✓	✓	✓
DM2	✓	X	✓
DM3	✓	✓	X

MHC= 4525.76			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	X

MHC= 4531.16			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	X

Slaughterhouse-S-L2

MHC= 4795.67			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	X	✓

MHC= 4806.67			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	X	X

MHC= 4818.92			
DM1	✓	✓	✓
DM2	✓	X	✓
DM3	✓	✓	✓

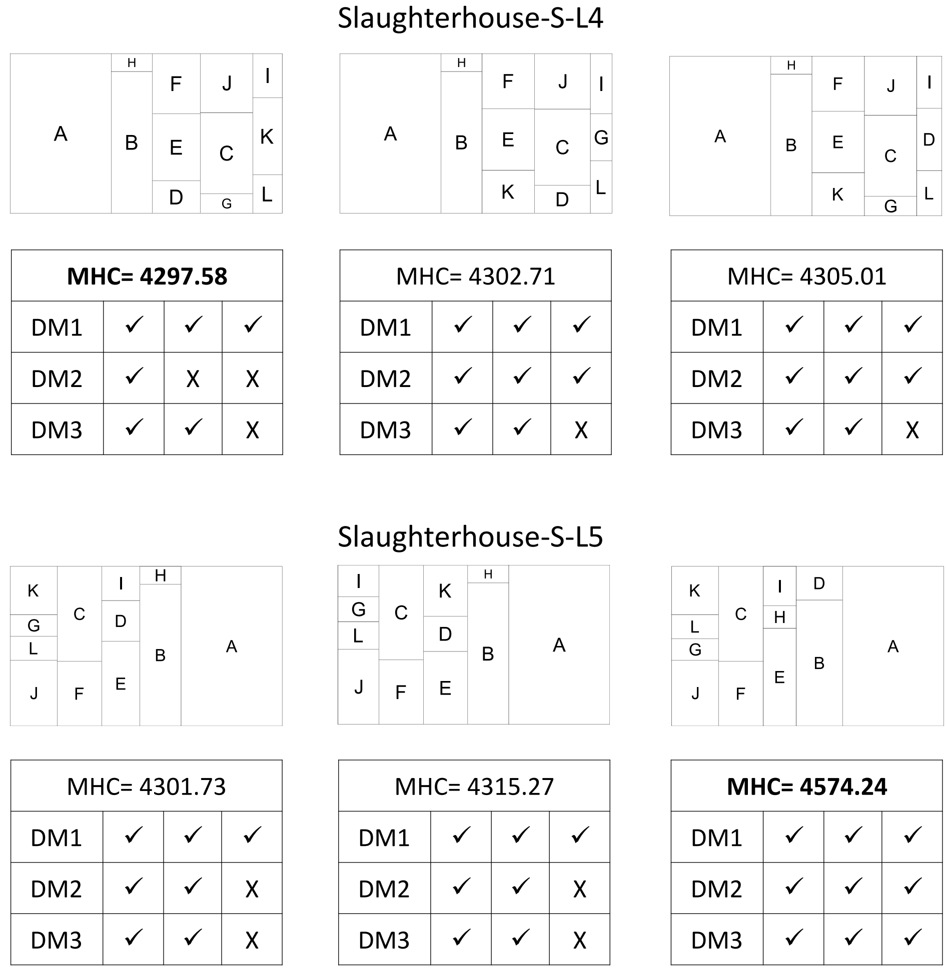
Slaughterhouse-S-L3

MHC= 4735.73			
DM1	✓	✓	✓
DM2	X	✓	✓
DM3	✓	✓	X

MHC= 4788.77			
DM1	✓	X	✓
DM2	X	✓	✓
DM3	✓	X	X

MHC= 4792.05			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	X	X

Figure 9. Best solutions found in *Sequential* launch 1 to 3 of *Slaughterhouse* problem.



MHC= 4297.58			
DM1	✓	✓	✓
DM2	✓	X	X
DM3	✓	✓	X

MHC= 4302.71			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	X

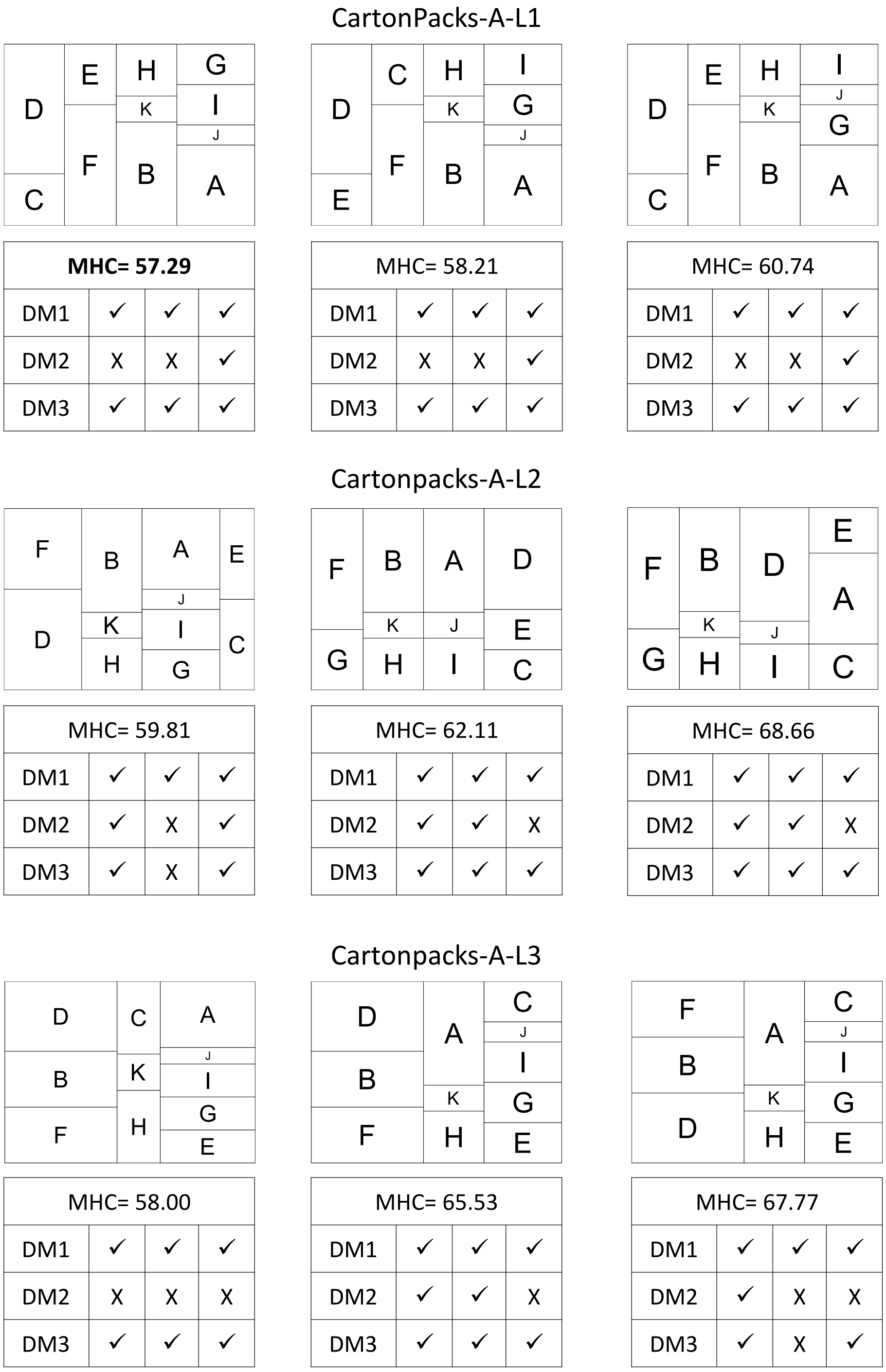
MHC= 4305.01			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	X

MHC= 4301.73			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	X

MHC= 4315.27			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	X

MHC= 4574.24			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	✓

Figure 10. Best solutions found in *Sequential* launch 4 to 5 of *Slaughterhouse* problem.



CartonPacks-A-L1

MHC= 57.29			
DM1	✓	✓	✓
DM2	X	X	✓
DM3	✓	✓	✓

MHC= 58.21			
DM1	✓	✓	✓
DM2	X	X	✓
DM3	✓	✓	✓

MHC= 60.74			
DM1	✓	✓	✓
DM2	X	X	✓
DM3	✓	✓	✓

Cartonpacks-A-L2

MHC= 59.81			
DM1	✓	✓	✓
DM2	✓	X	✓
DM3	✓	X	✓

MHC= 62.11			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

MHC= 68.66			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

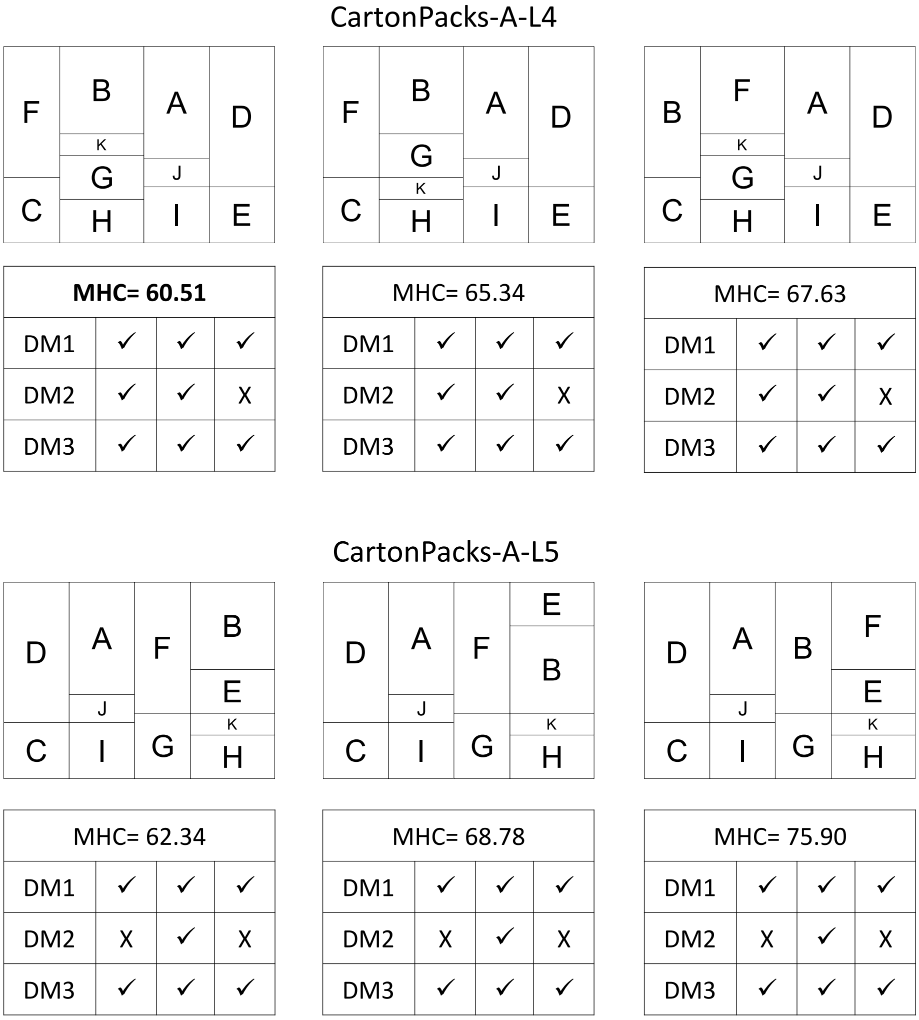
Cartonpacks-A-L3

MHC= 58.00			
DM1	✓	✓	✓
DM2	X	X	X
DM3	✓	✓	✓

MHC= 65.53			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

MHC= 67.77			
DM1	✓	✓	✓
DM2	✓	X	X
DM3	✓	X	✓

Figure 11. Best solutions found in *Alternative* launch 1 to 3 of *CartonPacks* problem.



CartonPacks-A-L4

MHC= 60.51			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

MHC= 65.34			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

MHC= 67.63			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

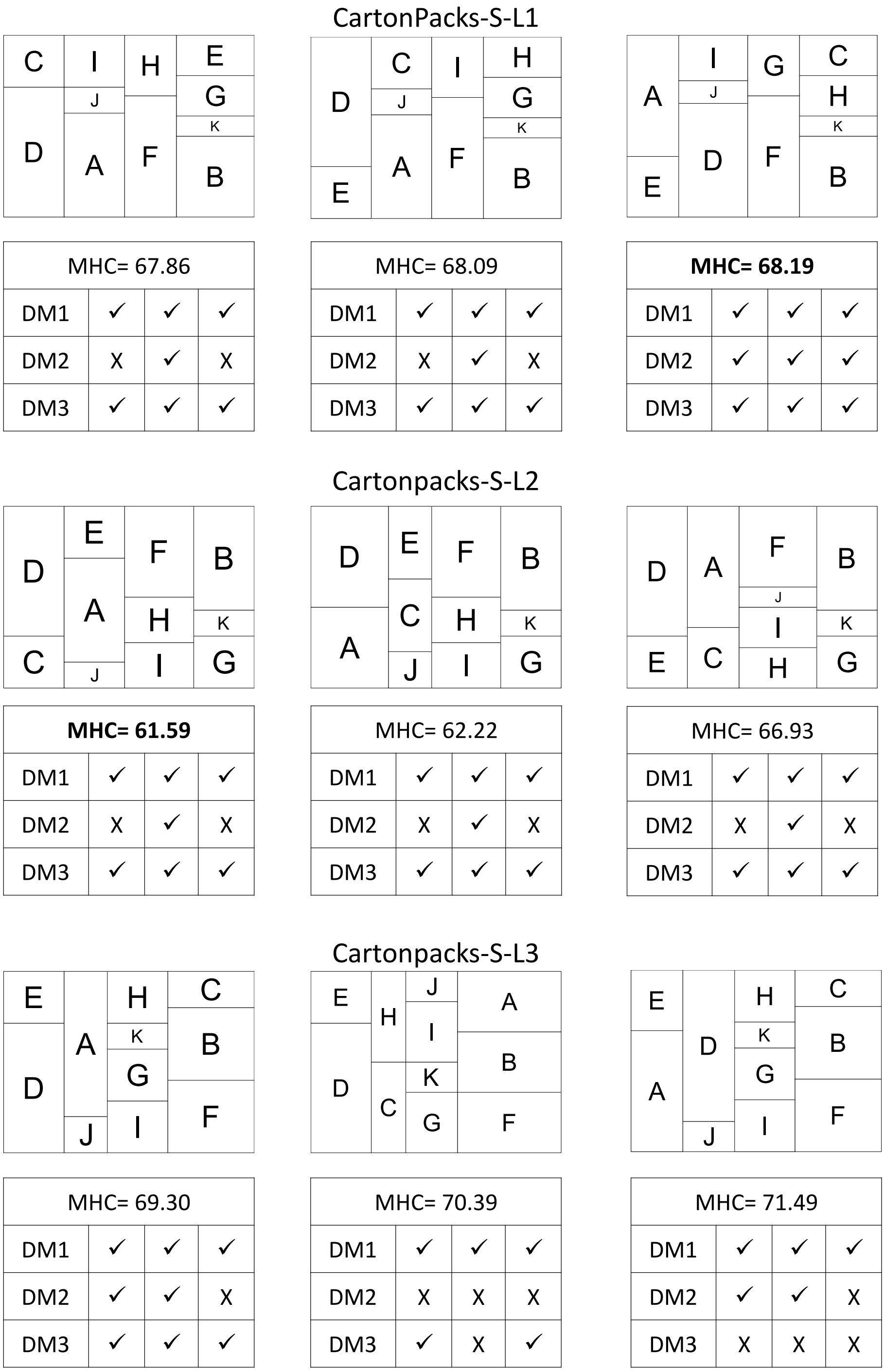
CartonPacks-A-L5

MHC= 62.34			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 68.78			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 75.90			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

Figure 12. Best solutions found in *Alternative* launch 4 to 5 of *CartonPacks* problem.



CartonPacks-S-L1

MHC= 67.86			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 68.09			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 68.19			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	✓

Cartonpacks-S-L2

MHC= 61.59			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 62.22			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 66.93			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

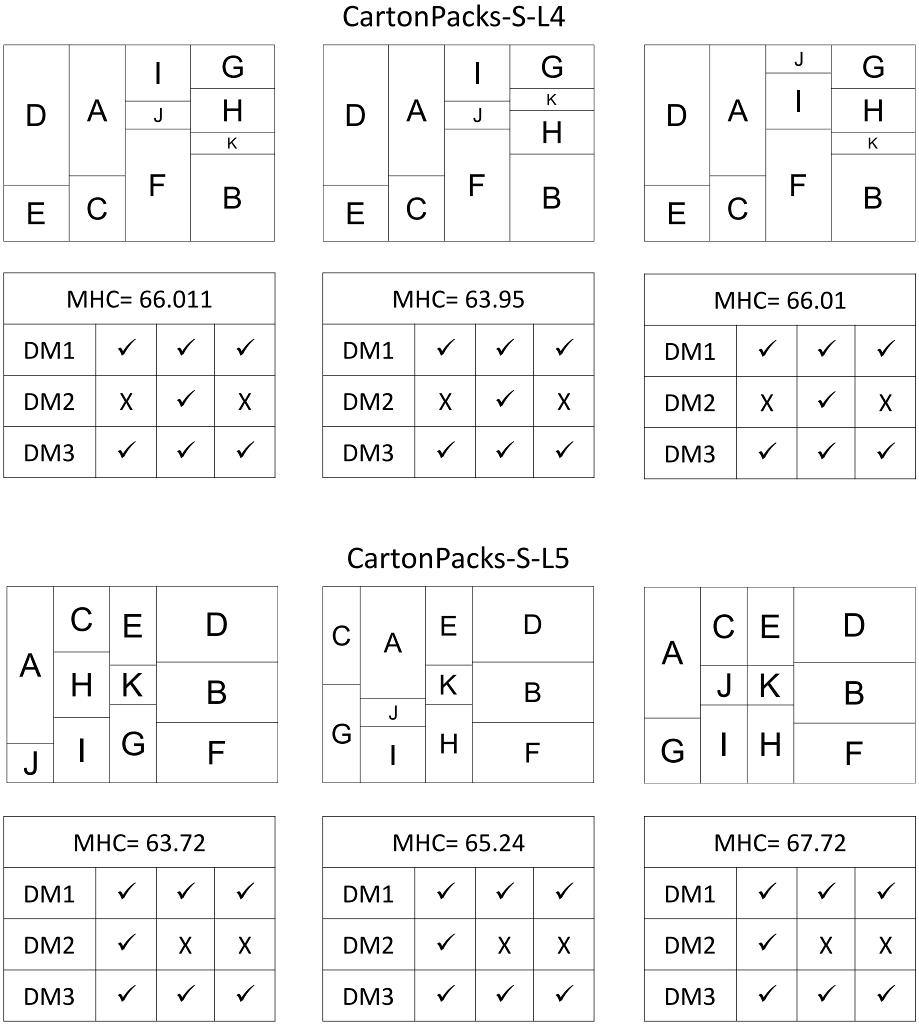
Cartonpacks-S-L3

MHC= 69.30			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

MHC= 70.39			
DM1	✓	✓	✓
DM2	X	X	X
DM3	✓	X	✓

MHC= 71.49			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	X	X	X

Figure 13. Best solutions found in *Sequential* launch 1 to 3 of *CartonPacks* problem.



MHC= 66.011			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 63.95			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

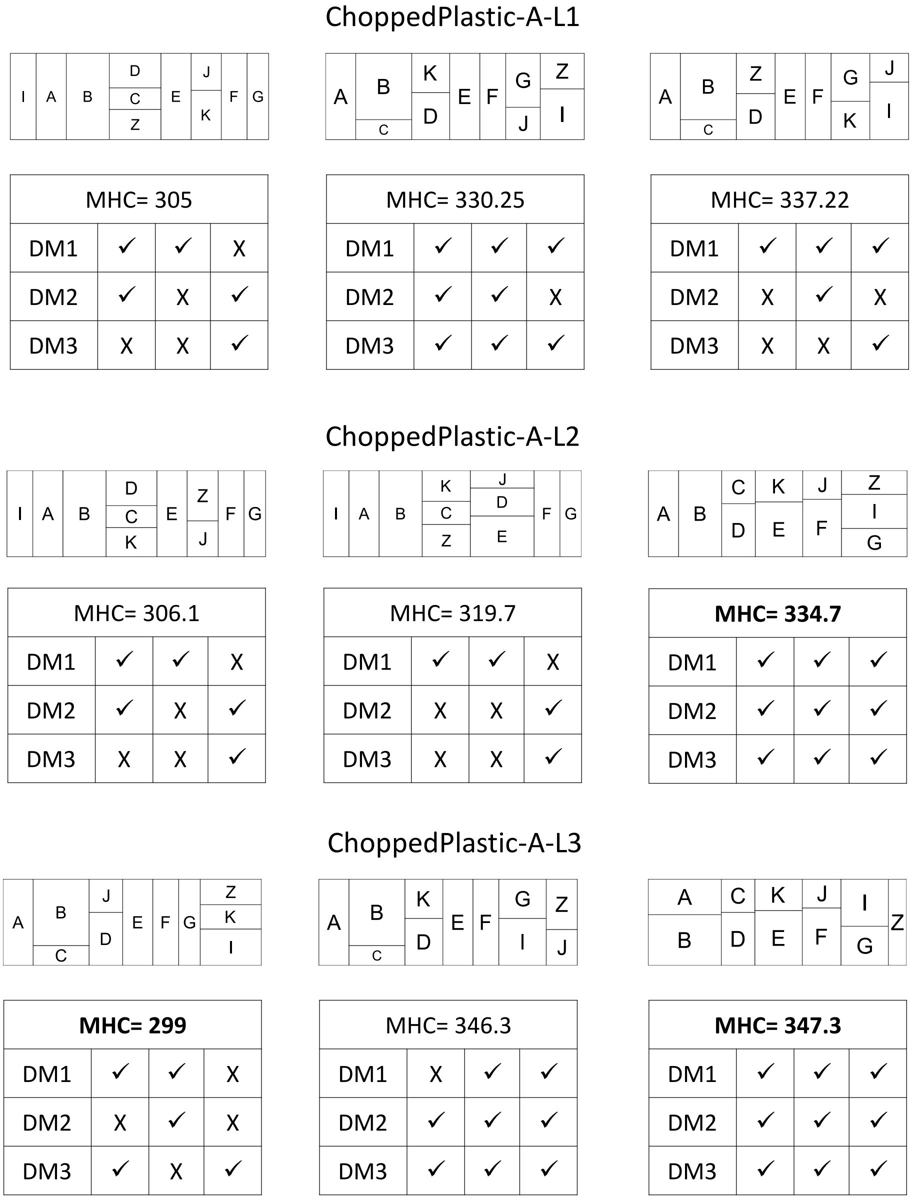
MHC= 66.01			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 63.72			
DM1	✓	✓	✓
DM2	✓	X	X
DM3	✓	✓	✓

MHC= 65.24			
DM1	✓	✓	✓
DM2	✓	X	X
DM3	✓	✓	✓

MHC= 67.72			
DM1	✓	✓	✓
DM2	✓	X	X
DM3	✓	✓	✓

Figure 14. Best solutions found in *Sequential* launch 4 to 5 of *CartonPacks* problem.



ChoppedPlastic-A-L1

MHC= 305			
DM1	✓	✓	X
DM2	✓	X	✓
DM3	X	X	✓

MHC= 330.25			
DM1	✓	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

MHC= 337.22			
DM1	✓	✓	✓
DM2	X	✓	X
DM3	X	X	✓

ChoppedPlastic-A-L2

MHC= 306.1			
DM1	✓	✓	X
DM2	✓	X	✓
DM3	X	X	✓

MHC= 319.7			
DM1	✓	✓	X
DM2	X	X	✓
DM3	X	X	✓

MHC= 334.7			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	✓

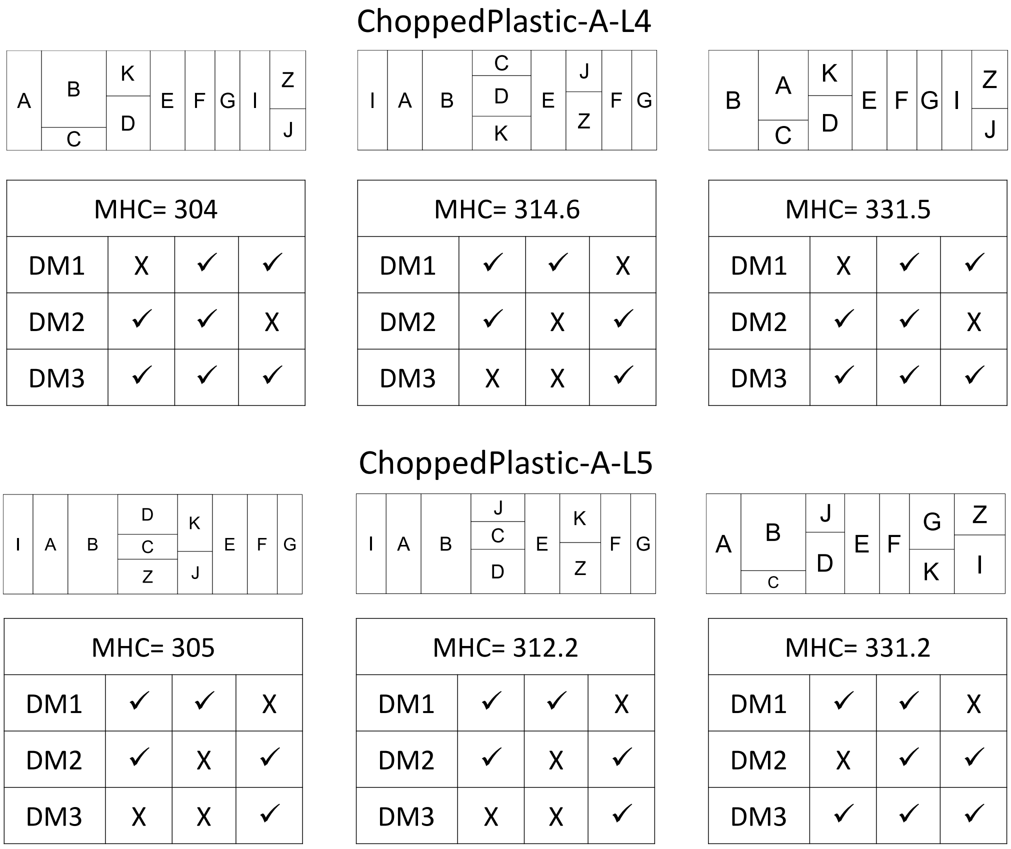
ChoppedPlastic-A-L3

MHC= 299			
DM1	✓	✓	X
DM2	X	✓	X
DM3	✓	X	✓

MHC= 346.3			
DM1	X	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	✓

MHC= 347.3			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	✓

Figure 15. Best solutions found in *Alternative* launch 1 to 3 of *ChoppedPlastic* problem.



MHC= 304			
DM1	X	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

MHC= 314.6			
DM1	✓	✓	X
DM2	✓	X	✓
DM3	X	X	✓

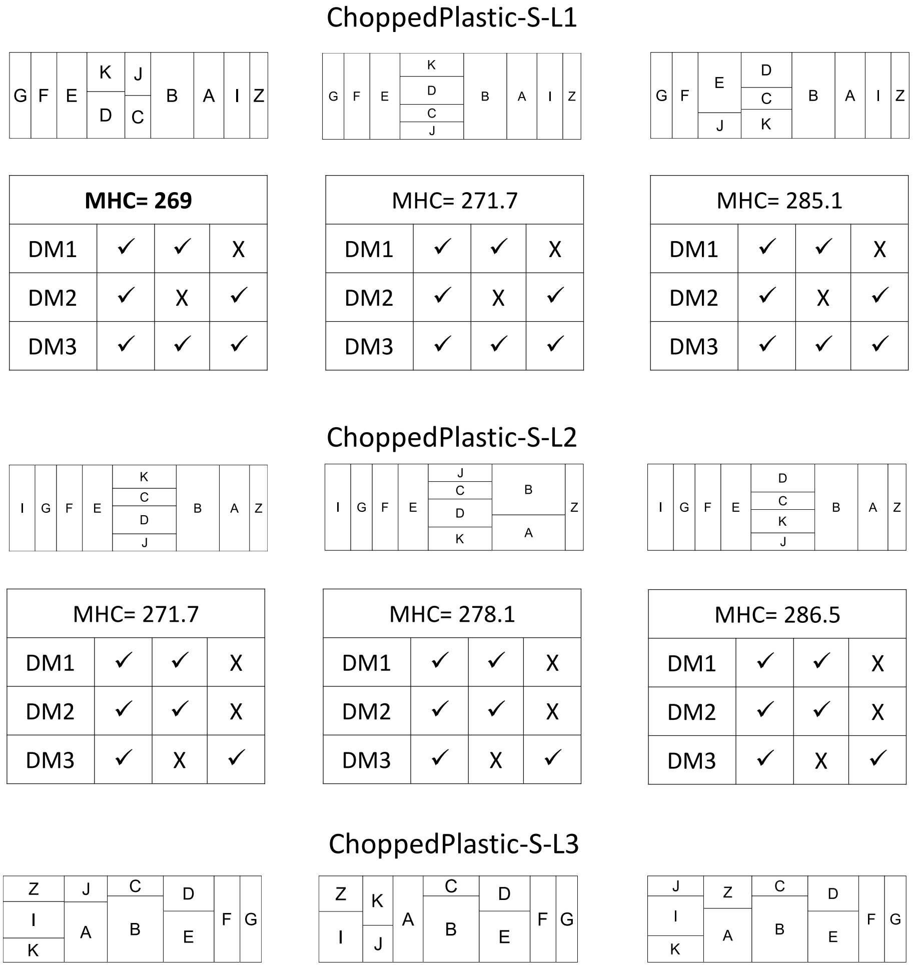
MHC= 331.5			
DM1	X	✓	✓
DM2	✓	✓	X
DM3	✓	✓	✓

MHC= 305			
DM1	✓	✓	X
DM2	✓	X	✓
DM3	X	X	✓

MHC= 312.2			
DM1	✓	✓	X
DM2	✓	X	✓
DM3	X	X	✓

MHC= 331.2			
DM1	✓	✓	X
DM2	X	✓	✓
DM3	✓	✓	✓

Figure 16. Best solutions found in *Alternative* launch 4 to 5 of *ChoppedPlastic* problem.



ChoppedPlastic-S-L1

MHC= 269			
DM1	✓	✓	X
DM2	✓	X	✓
DM3	✓	✓	✓

MHC= 271.7			
DM1	✓	✓	X
DM2	✓	X	✓
DM3	✓	✓	✓

MHC= 285.1			
DM1	✓	✓	X
DM2	✓	X	✓
DM3	✓	✓	✓

ChoppedPlastic-S-L2

MHC= 271.7			
DM1	✓	✓	X
DM2	✓	✓	X
DM3	✓	X	✓

MHC= 278.1			
DM1	✓	✓	X
DM2	✓	✓	X
DM3	✓	X	✓

MHC= 286.5			
DM1	✓	✓	X
DM2	✓	✓	X
DM3	✓	X	✓

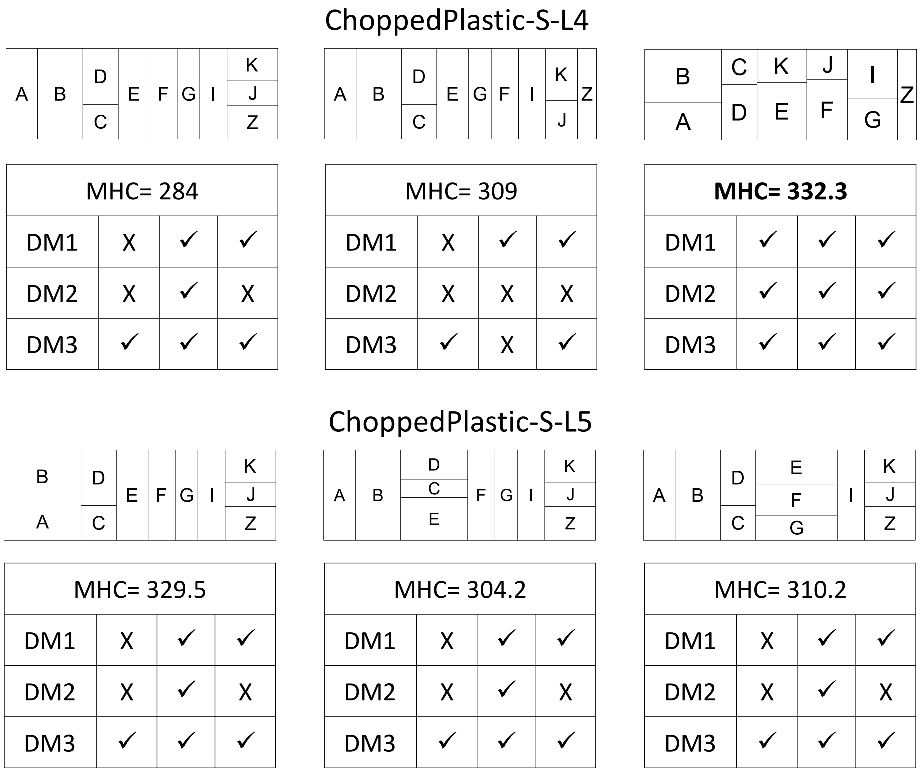
ChoppedPlastic-S-L3

MHC= 324.1			
DM1	✓	✓	✓
DM2	X	X	✓
DM3	✓	✓	✓

MHC= 324.7			
DM1	✓	✓	✓
DM2	X	X	✓
DM3	✓	✓	✓

MHC= 330.8			
DM1	✓	✓	✓
DM2	X	X	✓
DM3	X	✓	✓

Figure 17. Best solutions found in *Sequential* launch 1 to 3 of *ChoppedPlastic* problem.


ChoppedPlastic-S-L4

MHC= 284			
DM1	X	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 309			
DM1	X	✓	✓
DM2	X	X	X
DM3	✓	X	✓

MHC= 332.3			
DM1	✓	✓	✓
DM2	✓	✓	✓
DM3	✓	✓	✓

ChoppedPlastic-S-L5

MHC= 329.5			
DM1	X	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 304.2			
DM1	X	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

MHC= 310.2			
DM1	X	✓	✓
DM2	X	✓	X
DM3	✓	✓	✓

Figure 18. Best solutions found in *Sequential* launch 4 to 5 of *ChoppedPlastic* problem.

5. Conclusions

This article presents the application of multi-user interactive evaluation to the UA-FLP problem. To the best of our knowledge, this application has never been used in this kind of problem. Nevertheless, the multi-user evaluation strategy could be applied similarly to other kinds of problems where the collaborative intervention of several human designers is needed. The multi-user intervention has been proven to be as useful in a synchronous (or *alternative*) as in an asynchronous (or *sequential*) way, properly capturing the designers' personal requirements in both cases.

Furthermore, the users' intervention is only required for a short number of solutions since they are grouped by similitude between them, and they are only needed for a reduced number of generations along with the algorithm evolution, since the algorithm evolves by itself during a predetermined number of generations from the moment at which the users' preferences are captured. In this way, the humans' intervention is reduced to the minimum possible to reduce their fatigue.

The proposal has been tested using three problem instances taken from the literature. The three cases refer to the layout distribution of realistic industrial plants, including an industrial slaughterhouse and two recycling plants for cartons and plastic. In the three cases tested, the algorithm has been shown to be able to yield good results for the qualitative fitness function but adding the personal preferences of three expert designers and with a moderate intervention of the human designers.

Future research should be devoted to further reducing the designers' fatigue by attempting to incorporate a more automated means of capturing their preferences. In the same way, new techniques of incorporating the combined evaluation of the DMs can be implemented. Finally, more investigation could be conducted about the way in which the DMs intervene during the algorithm's evolution, attempting to obtain more definitive conclusions about the influence of the order in which they act over the algorithm, as well as implementing a ubiquitous synchronous method via the web for capturing several DMs' preferences.

Author Contributions: Conceptualization, L.S.-M. and L.G.-H.; methodology, L.S-.M.; software, C.C.-M.; validation, L.S.-M., L.G.-H. and C.C.-M.; formal analysis, L.S.-M. and L.G.-H.; investigation, L.S.-M., L.G.-H. and C.C.-M.; resources, L.S.-M.; data curation, C.C.-M.; writing—original draft preparation, L.S.-M.; writing—review and editing, L.S.-M.; visualization, L.S.-M. and L.G.-H.; supervision, L.S.-M. and L.G.-H.; project administration, L.S.-M. and L.G.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tompkins, J.; White, J.; Bozer, Y.; Tanchoco, J. *Facilities Planning*, 4th ed.; Wiley: New York, NY, USA, 2010.
2. Kouvelis, P.; Kurawarwala, A.A.; Gutierrez, G.J. Algorithms for robust single and multiple period layout planning for manufacturing systems. *Eur. J. Oper. Res.* **1992**, *63*, 287–303. [CrossRef]
3. Aiello, G.; Scalia, G.L.; Enea, M. A non dominated ranking multi objective genetic algorithm and electre method for unequal area facility layout problems. *Expert Syst. Appl.* **2013**, *40*, 4812–4819. [CrossRef]
4. Sun, X.; Lai, L.-F.; Chou, P.; Chen, L.-R.; Wu, C.-C. On GPU implementation of the island model genetic algorithm for solving the unequal area facility layout problem. *Appl. Sci.* **2018**, *8*, 1604. [CrossRef]
5. Armour, G.C.; Buffa, E.S. A heuristic algorithm and simulation approach to relative location of facilities. *Manag. Sci.* **1963**, *9*, 294–309. [CrossRef]
6. Lin, Y.Z.; Lin, Y.C. Applying an immune ant colony system algorithm to solve an integrated flexible bay facility layout problem with input/output points design. *Lect. Notes Manag. Sci.* **2015**, *7*, 56–62.
7. Besbes, M.; Zolghadri, M.; Costa Affonso, R.; Masmoudi, F.; Haddar, M. A methodology for solving facility layout problem considering barriers: Genetic algorithm coupled with A* search. *J. Intell. Manuf.* **2020**, *31*, 615–640. [CrossRef]
8. Drira, A.; Pierreval, H.; Hajri-Gabouj, S. Facility layout problems: A survey. *Annu. Rev. Control* **2007**, *31*, 255–267. [CrossRef]
9. Aiello, G.; Enea, M.; Galante, G.; La Scalia, G. Multi objective genetic algorithms for unequal area facility layout problems: A survey. In Proceedings of the Summer School Francesco Turco, Senigallia, Italy, 11–13 September 2013; pp. 95–100.
10. García-Hernández, L.; Salas-Morera, L.; Carmona-Muñoz, C.; Abraham, A.; Salcedo-Sanz, S. A novel multi-objective Interactive Coral Reefs Optimization algorithm for the Unequal Area Facility Layout Problem. *Swarm Evol. Comput.* **2020**, *55*, 100688. [CrossRef]
11. Salas-Morera, L.; García-Hernández, L.; Antolxix-Cabrera, A.; Carmona-Muñoz, C. Using eye-tracking into decision makers evaluation in evolutionary interactive UA-FLP algorithms. *Neural Comput. Appl.* **2020**, *32*, 13747–13757. [CrossRef]
12. Mls, K.; Cimler, R.; Vaščák, J.; Puheim, M. Interactive evolutionary optimization of fuzzy cognitive maps. *Neurocomputing* **2017**, *232*, 58–68. [CrossRef]
13. García-Hernández, L.; Arauzo-Azofra, A.; Salas-Morera, L.; Pierreval, H.; Corchado, E. Recycling plants layout design by means of an interactive genetic algorithm. *Intell. Autom. Soft Comput.* **2013**, *19*, 457–468. [CrossRef]
14. Sakai, M.; Takenouchi, H.; Tokumaru, M. Design support system with votes from multiple people using digital signage. In Proceedings of the 2014 IEEE International Symposium on Independent Computing (ISIC), Orlando, FL, USA, 9–12 December 2014; pp. 7–12.
15. Takenouchi, H.; Tokomaru, M.; Muranaka, N. Tournament Evaluation System Considering Multiple People's Kansei Evaluation. *Kansei Eng. Int. J.* **2010**, *9*, 43–50. [CrossRef]
16. Nomura, K.; Fukumoto, M. Asynchronous distributed interactive genetic algorithm for creating music melody reflecting multiple users' feelings. In Proceedings of the 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2017, Kanazawa, Japan, 26–28 June 2017; pp. 645–650.
17. Nomura, K.; Fukumoto, M. Music Melodies Suited to Multiples Users' Feelings Composed by Asyncrhronous Distributed Interactive Genetic Algorithm. *Int. J. Softw. Innov.* **2018**, *6*, 26–36. [CrossRef]
18. Palomo-Romero, J.M.; Salas-Morera, L.; García-Hernández, L. An island model genetic algorithm for unequal area facility layout problems. *Expert Syst. Appl.* **2017**, *68*, 151–162. [CrossRef]
19. Hosseini-Nasab, H.; Fereidouni, S.; FatemiGhomi, S.M.T.; Fakhrzad, M.B. Classification of facility layout problems: A review study. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 957–977. [CrossRef]
20. Montreuil, B. A modelling framework for integrating layout design and flow network design. In *Material Handling '90*; Springer: Berlin/Heidelberg, Germany, 1991; pp. 95–115.
21. Meller, R.; Narayanan, V.; Vance, P. Optimal facility layout design. *Oper. Res. Lett.* **1998**, *23*, 117–127. [CrossRef]

22. Konak, A.; Kulturel-Konak, S.; Norman, B.; Smith, A. A new mixed-integer programming formulation for unequal area facility layout problem in a continuous plane. In Proceedings of the IIE Annual Conference and Exhibition 2004, Hilton Americas, Houston, TX, USA, 15–19 May 2004; p. 63.
23. Konak, A.; Kulturel-Konak, S.; Norman, B.; Smith, A. A new mixed integer formulation for optimal facility layout. *Oper. Res. Lett.* **2006**, *34*, 660–672. [CrossRef]
24. Chae, J.; Regan, A.C. Layout design problems with heterogeneous area constraints. *Comput. Ind. Eng.* **2016**, *102*, 198–207. [CrossRef]
25. Aiello, G.; Enea, M.; Galante, G. A multi-objective approach to facility layout problem by genetic search algorithm and electre method. *Robot. Comput. Integr. Manuf.* **2006**, *22*, 447–455. [CrossRef]
26. Wong, K.Y. Applying ant system for solving unequal area facility layout problems. *Eur. J. Oper. Res.* **2010**, *202*, 730–746.
27. Liu, J.; Liu, J. Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems. *Appl. Soft Comput.* **2019**, *74*, 167–189. [CrossRef]
28. Pourvaziri, H.; Naderi, B. A hybrid multi-population genetic algorithm for the dynamic facility layout problem. *Appl. Soft Comput. J.* **2014**, *24*, 457–469. [CrossRef]
29. Chwif, L.; Pereira Barretto, M.R.; Moscato, L.A. A solution to the facility layout problem using simulated annealing. *Comput. Ind.* **1998**, *36*, 125–132. [CrossRef]
30. Jerin Leno, I.; Saravana Sankar, S.; Ponnambalam, S.G. An elitist strategy genetic algorithm using simulated annealing algorithm as local search for facility layout design. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 787–799. [CrossRef]
31. Matai, R. Solving multi objective facility layout problem by modified simulated annealing. *Appl. Math. Comput.* **2015**, *261*, 302–311. [CrossRef]
32. Turgay, S. Multi objective simulated annealing approach for facility layout design. *Int. J. Math. Eng. Manag. Sci.* **2018**, *3*, 365–380. [CrossRef]
33. Al-Saleh, M.A.; Mir, M.; Hassanin, A. Comparison of enhanced constructive layout optimization technique with Tabu-search and Particle Swarm Optimization methodologies. In Proceedings of the IEOM 2015—5th International Conference on Industrial Engineering and Operations Management, Dubai, United Arab Emirates, 3–5 March 2015.
34. Jeong, D.; Seo, Y. Golden section search and hybrid tabu search-simulated annealing for layout design of unequal-sized facilities with fixed input and output points. *Int. J. Ind. Eng. Theory Appl. Pract.* **2018**, *25*, 297–315.
35. McKendall, A.R.; Hakobyan, A. Heuristics for the dynamic facility layout problem with unequal-area departments. *Eur. J. Oper. Res.* **2010**, *201*, 171–182. [CrossRef]
36. Liu, J.; Wang, D.; He, K.; Xue, Y. Combining Wang–Landau sampling algorithm and heuristics for solving the unequal-area dynamic facility layout problem. *Eur. J. Oper. Res.* **2017**, *262*, 1052–1063. [CrossRef]
37. Derakhshan Asl, A.; Wong, K.Y. Solving unequal-area static and dynamic facility layout problems using modified particle swarm optimization. *J. Intell. Manuf.* **2017**, *28*, 1317–1336. [CrossRef]
38. Jolai, F.; Tavakkoli-Moghaddam, R.; Taghipour, M. A multi-objective particle swarm optimisation algorithm for unequal sized dynamic facility layout problem with pickup/drop-off locations. *Int. J. Prod. Res.* **2012**, *50*, 4279–4293. [CrossRef]
39. Liu, J.; Zhang, H.; He, K.; Jiang, S. Multi-objective particle swarm optimization algorithm based on objective space division for the unequal-area facility layout problem. *Expert Syst. Appl.* **2018**, *102*, 179–192. [CrossRef]
40. Salcedo-Sanz, S.; Del Ser, J.; La-Torres, I.; Gil-López, S.; Portilla-Figueras, J.A. The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *Sci. World J.* **2014**, *2014*, 739768. [CrossRef]
41. Durán-Rosal, A.M.; Gutiérrez, P.A.; Salcedo-Sanz, S.; Hervxaxs-Martínez, C. Dynamical memetization in coral reef optimization algorithms for optimal time series approximation. *Prog. Artif. Intell.* **2019**, *8*, 253–262. [CrossRef]
42. Camacho-Gómez, C.; Marsa-Maestre, I.; Gimenez-Guzman, J.M.; Salcedo-Sanz, S. A Coral Reefs Optimization algorithm with substrate layer for robust Wi-Fi channel assignment. *Soft Comput.* **2019**, *23*, 12621–12640. [CrossRef]
43. Salcedo-Sanz, S.; García-Herrera, R.; Camacho-Gómez, C.; Alexre, E.; Carro-Calvo, L.; Jaume-Santero, F. Near-optimal selection of representative measuring points for robust temperature field reconstruction with the CRO-SL and analogue methods. *Glob. Planet. Chang.* **2019**, *178*, 15–34. [CrossRef]
44. Tsai, C.W.; Chang, W.Y.; Wang, Y.C.; Chen, H. A high-performance parallel coral reef optimization for data clustering. *Soft Comput.* **2019**, *23*, 9327–9340. [CrossRef]
45. Camacho-Gomez, C.; Sanchez-Montero, R.; Martínez-Villanueva, D.; López-Espxix, P.-L.; Salcedo-Sanz, S. Design of a Multi-Band Microstrip Textile Patch Antenna for LTE and 5G Services with the CRO-SL Ensemble. *Appl. Sci.* **2020**, *10*, 1168. [CrossRef]
46. Azgomi, H.; Sohrabi, M.K. A novel coral reefs optimization algorithm for materialized view selection in data warehouse environments. *Appl. Intell.* **2019**, *49*, 3965–3989. [CrossRef]
47. Martín, A.; Vargas, V.M.; Gutiérrez, P.A.; Camacho, D.; Hervás-Martxixnez, C. Optimising Convolutional Neural Networks using a Hybrid Statistically-driven Coral Reef Optimisation algorithm. *Appl. Soft Comput. J.* **2020**, *90*, 106144. [CrossRef]
48. García-Hernández, L.; Salas-Morera, L.; Garcxixa-Hernxaxndez, J.; Salcedo-Sanz, S.; de Oliveira, J.V. Applying the coral reefs optimization algorithm for solving unequal area facility layout problems. *Expert Syst. Appl.* **2019**, *138*, 112819. [CrossRef]
49. García-Hernández, L.; Salas-Morera, L.; Carmona-Muñoz, C.; Garcia-Hernez, J.; Salcedo-Sanz, S. A novel Island Model based on Coral Reefs Optimization algorithm for solving the unequal area facility layout problem. *Eng. Appl. Artif. Intell.* **2020**, *89*, 103445. [CrossRef]

50. García-Hernández, L.; Garcxixa-Hernxaxndez, J.A.; Salas-Morera, L.; Carmona-Muñoz, C.; Alghamdi, N.S.; de Oliveira, J.V.; Salcedo-Sanz, S. Addressing Unequal Area Facility Layout Problems with the Coral Reef Optimization algorithm with Substrate Layers. *Eng. Appl. Artif. Intell.* **2020**, *93*, 103697. [CrossRef]
51. Tate, D.M.; Smith, A.E. Unequal area facility layout using genetic search. *IIE Trans.* **1995**, *27*, 465–472. [CrossRef]
52. Gomez, A.; Fernez, Q.; laFuenteGarcia, D.D.; Garcia, P. Using genetic algorithms to resolve layout problems in facilities where there are aisles. *Int. J. Prod. Econ.* **2003**, *84*, 271–282. [CrossRef]
53. Dunn, J.C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *J. Cybern.* **1973**, *3*, 32–57. [CrossRef]
54. García-Hernández, L.; Pierreval, H.; Salas-Morera, L.; Arauzo-Azofra, A. Handling qualitative aspects in unequal area facility layout problem: An interactive genetic algorithm. *Appl. Soft Comput.* **2013**, *13*, 1718–1727. [CrossRef]
55. Salas-Morera, L.; Cubero-Atienza, A.; Ayuso-Munoz, R. Computer-aided plant layout | Distribucion en planta asistida por ordenador. *Inf. Tecnol.* **1996**, *7*, 39–46.
56. García-Hernández, L.; Arauzo-Azofra, A.; Salas-Morera, L.; Pierreval, H.; Corchado, E. Facility layout design using a multi-objective interactive genetic algorithm to support the DM. *Expert Syst.* **2015**, *32*, 94–107. [CrossRef]