

# A Hyper-Heuristic Approach to solve the Multi-Objective Container Loading Problem

Y. González, C. León, and G. Miranda

Universidad de La Laguna, 28271 Tenerife, Spain  
ygonzale@ull.es | cleon@ull.es | gmiranda@ull.es

## 1 Introduction

The *Container Loading Problem* (CLP) belongs to an area of active research and has numerous applications in the real world, particularly in container transportation and distribution industries. When solving the CLP, normally, the goal is to distribute a set of rectangular pieces (boxes) in one large rectangular object (container) so as to maximize the total volume of packed boxes. However, a rather common aspect in the scope of this problem is the weight limit of the containers, since they normally can't exceed a certain weight for their transportation.

In our case of study, the goal is to load the items (boxes) that would provide the highest total volume and weight to the container. These two objectives are conflicting because the volume of the box is usually not proportional to its weight. This way, the problem can be stated as a multi-objective optimization problem, trying to optimize the pieces layout inside the container so that the volume is maximized at the same time that the weight, without exceeding the weight limits. The formulation of the here addressed problem is as follows. We have a container with known width  $W$ , length  $L$ , height  $H$ , maximum weight  $P_{max}$ , and a set of  $N$  rectangular boxes. These boxes belong to one of the sets of boxes types  $\mathcal{D} = \{T_1 \dots T_m\}$ . Associated with each  $T_i$  type exists a weight  $p_i < P_{max}$  and a volume  $v_i$ . The aim is to find:

$$\max \sum_{i=1}^m x_i v_i \quad \text{and} \quad \max \sum_{i=1}^m x_i p_i$$

In the literature, although there are some isolated works that use exact algorithms to deal with the CLP, most studies focus on providing solutions using heuristics and metaheuristics, because, computationally, the CLP is a NP-hard problem [1]. Many approaches deal with single-objective formulations of the CLP, the works dealing with a multi-objective formulation of the problem are almost non-existent [2].

## 2 Multi-objective Approach

The multi-objective evolutionary algorithms (MOEA) have shown promise for solving other problems in the area of cutting and packing. To apply MOEAs to the CLP, have been designed: a solutions representation scheme, a set of evolutionary operators and two methods for evaluation of the candidate solutions.

### 2.1 Representation of candidate solutions

For the representation of candidate solutions, we have defined a sequence of genes  $G_1, \dots, G_s$  where each gene consists of three elements  $(t_i, n_i, r_i)$ , with  $t_i \in [t_1, t_m]$ ,  $n_i \in [1, b_i]$  and  $r_i \in [0, o_i]$ . So, it is a sequence formed by piece type ( $t_i$ ), number of pieces of that type ( $n_i$ ) and rotation for those pieces ( $r_i$ ). This will determine the order and orientation in which the pieces will be put inside the container. Each piece has allowed two, four or six possible orientations. A valid chromosome must contain all the pieces of each different type.

### 2.2 Evaluation of the objectives

In order to decide where exactly to locate each item, we propose two evaluation heuristics based on a fill-by-levels strategy in which the pieces are stored into the container: the *Single-Level Filling Heuristic* (SLFH) implemented in [3] and *Multiple-Level Filling Heuristic* (MLFH) implemented in [4]. Both heuristics are based on the creation and management of piece levels or layers within the container. Such levels or layers identify empty spaces inside the container, and thus, they represent areas where to locate items. MLFH works as follows:

- The first piece in the chromosome is introduced at the bottom left corner of the container. This piece will determine the dimensions of the spaces to be generated: one in front of the box, other above, and other beside the placed item.
- The following piece in the chromosome is selected and placed into the container according to the next guidelines:
  - First, we try to fill the space created in front of the box. If there is no available space for our current piece in the front-level, we check for available space in the above-level. If even then, the current piece doesn't fit, we try to locate it into the level beside the box.
  - When the piece fits into a level, the box is placed at the bottom left corner of the level. In such a case, it must be checked if the box fits into the level without leaving empty space. If so, the given level is now completed and thus, we can avoid it from our list of open or available levels to locate items. If not, it means that there is still some remaining free space in front, above and/or beside the already placed box, and so it's necessary to create a set of new - in front, above, and/or beside - levels.
  - At any moment, when checking for a given type of level, e.g., above level, first we must consider the most recently created above-level.
- When the current analysed box doesn't fit in any of the available levels, the procedure finishes, and no more items are loaded into the container. At this moment, it's possible to compute the value of the objectives (total volume and weight) by adding the volume and weight of all the loaded pieces.

The difference between SLFH and MLFH is that SLFH never uses the next layer if there are other empty spaces that can allocate items. However, in MLFH all layers are available to accommodate pieces, i.e., if a given piece doesn't fit into the current empty spaces, it is located into the next and unused layer, thus creating a new set of empty spaces.

### 2.3 Operators

We have adapted the one point crossover operator to the CLP. Taking into account an individual represented by the chromosome  $C1 = (G1_1, \dots, G1_{s_1})$  and another individual represented by the chromosome  $C2 = (G2_1, \dots, G2_{s_2})$ . The first part of each individual, is modified to respect the total number of pieces of each type present in the chromosome. That is, we remove any extra piece types if there are too many pieces of this type, or we add if missing

For the mutation we have introduced three different types of movements on the chromosome, each of which applied under the probability of mutation:

- *Add one gene*: a type of piece is randomly generated  $t_x \in [t_1, t_m]$ . Then, all genes with this type of piece are searched, and we select one with more than one associated piece. A number of pieces  $n_x < n_y$  is chosen from the selected gene  $G_y = (t_y, n_y, r_y)$  with  $t_y = t_x$ , so that the pieces are distributed between that gene  $(t_y, n_y - n_x, r_y)$  and the new one  $(t_x, n_x, r_x)$ . The orientation is chosen from those allowed for that type of piece  $r_x \in [0, o_x)$ . Finally, we choose the position of the chromosome in which to insert the new gene, moving the rest to the right.
- *Remove a gene*: a position within the chromosome is randomly selected. If the piece type  $t_x$  of the selected gene  $G_x = (t_x, n_x, r_x)$  appears more times in that chromosome, then a gene  $G_y = (t_y, n_y, r_y)$  is randomly selected from among the same type, i.e.  $t_y = t_x$ , and the number of pieces will be increased with the number of pieces of the first gene to be removed ( $n_y = n_y + n_x$ ). As it is possible that both genes do not have the same type of orientation of pieces ( $r_x \neq r_y$ ), one of them is randomly selected. Finally,  $G_x$  is eliminated by compaction to the left.
- *Change a gene*: a random position of the chromosome is selected and the type of orientation is randomly changed within the possible orientations for the piece type.

### References

1. Scheithauer, Guntram, "Algorithms for the Container Loading Problem". Operations Research Proceedings 1991, Springer-Verlag, pp.445–452, 1992
2. Andreas Bortfeldt & Gerhard Wäscher, "Container Loading Problems - A State-of-the-Art Review". FEMM Working Papers 120007, Otto-von-Guericke University Magdeburg, Faculty of Economics and Management.
3. de Armas, J.; Gonzalez, Y.; Miranda, G.; Leon, C., "Parallelization of the multi-objective container loading problem". Evolutionary Computation (CEC), 2012 IEEE Congress, pp.1,8, 10-15 June 2012
4. González, Y.; Miranda, G.; León, C., "A Multi-level Filling Heuristic for the Multi-objective Container Loading Problem". Springer International Publishing, pp. 11-20, 2013