# A Hyper-heuristic for Solving One and Two-dimensional Bin Packing Problems

Eunice López-Camacho, Hugo Terashima-Marín ITESM-Intelligent Systems Av. E. Garza Sada 2501 Monterrey, N.L., 64849 Mexico {eunice.lopez, terashima}@itesm.mx

# ABSTRACT

The idea behind hyper-heuristics is to discover rules that relate different problem states with the best single heuristic to apply. This investigation works towards extending the problem domain in which a given hyper-heuristic can be applied and implements a framework to generate hyperheuristics for a wider range of bin packing problems. We present a GA-based method that produces general hyperheuristics that solve a variety of instances of one- and twodimensional bin packing problem without further parameter tuning. The two-dimensional problem instances considered deal with rectangles, convex and non-convex polygons.

**Categories and Subject Descriptors:** I.2 [Computing Methodologies]: Artificial Intelligence — Problem Solving, Control Methods and Search.

General Terms: Algorithms.

**Keywords:** Evolutionary Computation, Hyper-heuristics, Optimization, Bin Packing Problem, 2D irregular Bin Packing Problem.

## **1. INTRODUCTION**

The one-dimensional (1D) and two-dimensional (2D) bin packing problem (BPP) are particular cases of the cutting and packing problem, which consists of finding an arrangement of items or pieces inside identical objects such that the number of objects required to contain all pieces is minimum. Heuristic approaches for the 2D bin packing problem present at least two phases: first, the **selection** of the next piece to be placed and the corresponding object to place it; and second, the actual **placement** of the selected piece in a position inside the object according to a given criteria.

A hyper-heuristic is a re-usable method that chooses between a range of heuristic approaches to robustly tackle a wider range of problems. This paper proposes a hyperheuristic generation model that is useful for solving 1D and 2D BPP. The proposed model was applied to one-dimensional bin packing problems [3], and later adapted separately to the 2D regular [5] and 2D irregular (convex) packing problems [6]. Now, we integrate these problems into one framework and extend the hyper-heuristic solution model to instances including non-convex polygons increasing the level of geometrical complexity and computational burden.

Copyright is held by the author/owner(s). GECCO'11, July 12–16, 2011, Dublin, Ireland. ACM 978-1-4503-0690-4/11/07. Peter Ross School of Computing Napier University Edinburgh EH10 5DT UK P.Ross@napier.ac.uk

#### Table 1: Representation of the instance state.

Feature	Description	
1	Number of pieces.	
2	Mean area of remaining pieces.	
3	Variance of the area of remaining instance pieces.	
4	Mean of the rectangularity of remaining pieces.	
5	Variance of the rectangularity of remaining pieces.	
6	Mean of the height of the remaining pieces.	
7	Variance of the width of the remaining pieces.	
8	Fraction of remaining pieces in the instance	
	whose area is above $1/2$ of the object area.	
9	Mean of the degree of concavity of the remaining pieces.	
10	Fraction of the instance total items remaining.	

# 2. COMBINING HEURISTICS WITH A GE-NETIC ALGORITHM (GA)

Ross et al. [3] and Terashima et al. [5, 6] present a GAbased method that produces general hyper-heuristics that solve the 1D and the 2D BBP respectively. Each individual in the GA population is a possible hyper-heuristic, that is, a rule that relates each possible instance state (condition) with a single heuristic to be applied (action).

Set of Heuristics. The following six selection heuristic were employed: First Fit Decreasing (FFD), Filler + FFD, Best Fit Decreasing, Djang and Finch with initial fullness of 1/4, with initial fullness of 1/3 and with initial fullness of 1/2. The 1D and 2D cases of the BPP share the same selection heuristics. For the 2D BPP, additionally a placement heuristic has to be applied to find a solution: Constructive Approach with Maximum Adjacency [6].

**Representation of the instance state.** Each instance to be solved by the hyper-heuristic is characterized by a numerical vector that summarizes some of its relevant features. According to this numerical vector, the hyper-heuristic decides which single heuristic to apply every time. We employed the methodology proposed in [2] with all our testbed instances and found different features that are related to heuristic performance (see Table 1).

All items in 1D instances have only one dimension (height), so, their width has a variance of zero. For 1D instances, area is computed assuming all items and bins have a fixed width. Rectangularity is a quantity that represents the proportion between the area of a piece and the area of a horizontal rectangle containing it. The degree of concavity is defined in [7]. For a concave polygon, the degree of concavity is > 1. It is possible to know what kind of instance we have at hand when looking at the features 4, 7 and 9 of the tenvalue numerical representation of an instance (see Table 1). For example, a 2D regular instance has the following values: variance of width  $\neq 0$ , mean of rectangularity = 1 and mean of degree of concavity = 1.

**Chromosomes.** Each chromosome is composed of a series of *blocks*. Each block includes several numbers. All numbers in a block, except the last one, represent an instance state which is the numerical vector mentioned above. The last number identifies a single heuristic (from the 6 selection heuristics). An individual solves a problem instance as following: given an instance and having computed its numerical representation, find the closest block in the chromosome and apply the related single heuristic. This will place one or several items or pieces and will produce a new problem-state representation. The process is repeated until all pieces are placed. The task is to find a chromosome (hyper-heuristic) that is capable of obtaining good solutions for a wider variety of problems.

### 3. EXPERIMENTS AND RESULTS

Our experimental testbed is comprised by a total of 1417 instances. The 397 1D problem instances were drawn from the literature [1,4,8]. We have 540 2D instances containing only convex polygonal pieces that were randomly generated in [6] including 30 rectangular instances. The 480 new 2D instances containing some non-convex polygons were randomly produced for this investigation. There is a variety of instance feature values; for example, there are instances whose pieces have an average size of 1/30 of the object, while other instances have huge pieces (averaging almost 2/3 of object each piece).

Two experiments were conducted sorting the available instances in two balanced training and testing sets and other two experiments swapped training and testing sets. Four experiments overall. Experiments were conducted with population of size 30, crossover probability of 1.0, mutation probability of 0.10, for 80 generations. For each of the four experiments five GA processes were run. For each complete run the two individuals with highest fitness were employed to solve the whole testing set and the best was selected as the hyper-heuristic of the run. Overall, 20 hyper-heuristics were employed to measure the effectiveness of the model.

Each hyper-heuristic generated was used to solve the testing set of instances of the experiment where it comes from. There exists correspondence between the category of problem instances and the single heuristics more often employed (test of contingency table with the  $\chi^2$  statistic, p-value < 0.001). This is what we expected because different categories of instances have different numerical representations; so, the hyper-heuristics suggest different single heuristics to apply. For 1D instances, the Filler heuristic was employed 29.1% of the times, while this heuristic was chosen only 7.5% of the times when solving 2D convex instances.

In Table 2, we report the average number of extra objects delivered by the best-hyperheuristic per experiment, compared against the number of objects employed by the best single heuristic for each instance. For experiments 1, 3 and 4, the best hyper-heuristic for 1D instances deliver less objects than the correspondent best single heuristic (numbers are negative for 1D instances). On average, the best hyperheuristic of the 5 runs solved 1D instances employing 0.028 objects less than the best of the 6 single heuristics. For Experiment 1, extra number of objects delivered by the best hyper-heuristic is statistically different for 1D and for 2D instances (non-parametric Mann-Whitney U statistical test for means, p-value = 0.001). For the rest of the experiments, the difference is not significant between 1D and 2D BPP.

Table 2: Average number of extra objects delivered by the best-hyperheuristic, compared against results of the best single heuristic for each instance.

Experiment	1D	Convex 2D	Non Convex 2D
1	-0.090	0.007	-0.008
2	0	0	0
3	-0.005	0.011	0.008
4	-0.015	-0.007	0.021
Average	-0.028	0.003	0.005

# 4. CONCLUSIONS

In this research we applied a hyper-heuristic approach to solve efficiently a wide range of 1D and 2D bin packing problem instances with minimum parameter tuning and good results. Among the 2D instances, there are rectangles, convex and non-convex polygons. For some of the instances, the hyper-heuristics achives better results than the best of the single heuristics.

#### 5. ACKNOWLEDGMENTS

This research was supported in part by ITESM under the Research Chair CAT-144 and the CONACYT Project under grant 99695.

#### 6. **REFERENCES**

- E. Falkenauer. A Hybrid Grouping Genetic Algorithm for Bin Packing, 1996.
- [2] E. López-Camacho, H. Terashima-Marín, and P. Ross. Defining a problem-state representation with data mining within a hyper-heuristic model which solves 2D irregular bin packing problems. In Á. F. Kuri-Morales and G. R. Simari, editors, Advances in Artificial Intelligence IBERAMIA, volume 6433 of Lecture Notes in Computer Science, pages 204–213. Springer, 2010.
- [3] P. Ross, J. G. Marín-Blázquez, S. Schulenburg, and E. Hart. Learning a procedure that can solve hard bin-packing problems: A new GA-based approach to hyper-heuristics. In *Conference* on Genetic and Evolutionary Computation. Lecture Notes in Computer Science, volume 2724, pages 1295–1306. Springer-Verlag, 2003.
- [4] A. Scholl, R. Klein, and C. Jürgens. Bison: a fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Comput. Oper. Res.*, 24:627–645, July 1997.
- [5] H. Terashima-Marín, C. J. Farías-Zárate, P. Ross, and M. Valenzuela-Rendón. A GA-based method to produce generalized hyper-heuristics for the 2D-regular cutting stock problem. In Conference on Genetic and Evolutionary Computation. Lecture Notes in Computer Science, pages 591–598, New York, NY, USA, 2006. ACM Press.
- [6] H. Terashima-Marín, P. Ross, C. J. Farías-Zárate, E. López-Camacho, and M. Valenzuela-Rendón. Generalized hyper-heuristics for solving 2D regular and irregular packing problems. Ann. Oper. Res., 179:369–392, 2010.
- [7] W. X. Wang. Binary image segmentation of aggregates based on polygonal approximation and classification of concavities. *Pattern Recognition*, 31(10):1503-1524, 1998.
- [8] G. Wäscher and T. Gau. Heuristics for the integer one-dimensional cutting stock problem: A computational study. OR Spectrum, 18:131–144, 1996. 10.1007/BF01539705.