The Impact of the Bin Packing Problem Structure in Hyper-heuristic Performance

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

ABSTRACT

We use a knowledge discovery approach to get insights over the features of the bin packing problem and its relationship in the performance of an evolutionary-based model of hyper-heuristics. The evolutionary model produces rules that combine the application of up to six different low-level heuristics during the solution of a given problem instance. Using the Principal Component Analysis (PCA) method, we visualize in two dimensions all instances characterized by a larger number of features. By over imposing features and hyper-heuristic performance over the 2D graphs, it is possible to draw conclusions about the relation between the bin packing problem structure and the hyper-heuristics performance.

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

General Terms: Algorithms.

Keywords: Bin Packing, 2D irregular Bin Packing Problem, Optimization, Heuristics, Hyper-heuristics, Principal Component Analysis.

1. INTRODUCTION

The bin packing problem (BPP) consists of finding an arrangement of pieces inside identical objects, minimizing the number of objects required. A better understanding of the problem features will bring new elements to shed light on the design of new heuristics and hyper-heuristics methods. Some efforts to predict heuristic performance based on problem characteristics have been done [6].

PCA reduces the dimensionality of the data while retaining most of the variation in the data set. PCA builds uncorrelated variables called principal components, which are linear combinations of the original variables. The two components with largest variance are usually chosen as new axis for plotting all observations, making it possible to visually assess similarities and differences between observations [5].

2. THE HYPER-HEURISTIC STRUCTURE

A method that produces general hyper-heuristics for one-dimensional (1D) and two-dimensional (2D) BPP instances is proposed in [2]. A hyper-heuristic is a rule that selects a single heuristic to be applied (action) based on each possible instance state (condition). Once a hyper-heuristic is developed, it is able to solve any 1D or 2D instance without further parameter tuning. The proposed method is based on a genetic algorithm that evolves combinations of condition-action rules (called hyper-heuristics). The structure or characterization of 1D and 2D problem instances is summarized by several features in a numerical vector.

For some of the instances, hyper-heuristics achieve better results than the best of the single heuristics showing that combination of single heuristics may outperform any of the single heuristics considered separately. For most instances, hyper-heuristics get the same result than the best single heuristic. This is beneficial as well, as the choice of best heuristic varies from instance to instance [2].

3. EXPERIMENTAL SETUP

Our experimental testbed is comprised by a total of 1417 instances [2]. The 397 1D problem instances were drawn from the literature, the 540 2D instances containing only convex polygonal pieces and the 480 2D instances containing some non-convex polygons were randomly generated. There is a variety of instance feature values; for example, average size of the pieces goes from 1/30 to 1/3 of the object size. Six selection heuristic approaches were employed: First Fit Decreasing, Filler, Best Fit Decreasing, Djang and Finch with initial fullness of 1/4, 1/3 and 1/2. The heuristic Constructive Approach with Maximum Adjacency was employed as the placement heuristic for the 2D instances.

A critical part of the proposed analysis is the identification of suitable features of the problem instances that reflect the structure of the problem and the characteristics of the instances that might explain algorithm performance. Many features can be derived from a given instance of the BPP (especially the 2D irregular BPP), making this problem in particular, complex and difficult to understand thoroughly. The methodology proposed in [1] finds a subset of features related with single heuristics performance, given a larger set of problem features. With this methodology, we reduced a set of 23 computed features to only nine: 1. Number of pieces. 2. Mean the area of the pieces. 3. Variance of the area. 4. Mean of the rectangularity of the pieces. 5. Variance of the rectangularity. 6. Mean of the height of the pieces. 7. Variance of the width. 8. Percentage of pieces whose area is above 1/2 of the object area. 9. Mean of degree of concavity of the pieces (defined in [7]). For 1D instances, area is proportional to height and width variance is zero. Rectangularity is a quantity that represents the proportion between the area of a piece and the area of a horizontal rectangle containing it. These nine features were chosen for the numerical representation of the evolutionary

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

process and also for the PCA analysis. The basic question is about what features values have those instances that are the more suitable to be solved better by the hyper-heuristics.

4. RESULTS

We performed the PCA considering the 1417 instances and the 9 previously selected variables (using the *R* programming language [4]). To ensure magnitude consistency, we standardized every variable (average of 0 and standard deviation of 1). The first two principal components explain 65% of the total variance, jointly. Close points in Figure 1 represent similar instances according to the 9 variables. The largest values of principal component 1 (PC1) refer to instances with high variability of pieces sizes and pieces width. A lower value of PC2 is related to instances with large items and large variability of items sizes.

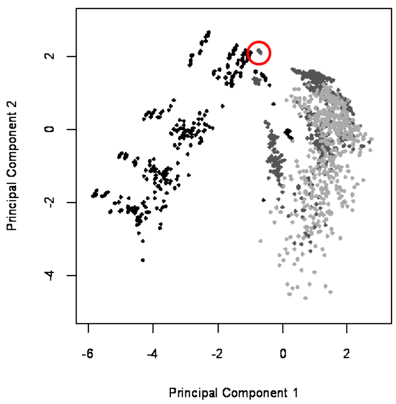

Figure 1: The 1417 instances plotted along PC1 and PC2. In black, 1D instances. In dark gray, instances of the 2D BPP (convex). Inside the circle, the 30 rectangular instances considered. In light gray, instances of the 2D irregular BPP (non-convex).

Figure 2 marks with letters *b* and *w* those instances whose best hyper-heuristic obtained a different result (better or worse) compared against the result of the best of the six heuristics. These cases are concentrated in a few sections. The best and worst cases are mixed in these sections, which means that this particular analysis is able to show which instances are likely to be solved *different* by the hyper-heuristic compared with the best single heuristic, but does not distinguish between solving cases with fewer or more objects. We want to know which characteristics have those sections of the graph to know which features of the BPP are able to explain hyper-heuristic performance. 1D instances with small items with similar sizes characterize the zone in the rectangle of Figure 2. The *b*'s and *w*'s outside the rectangle are for 2D instances. Most of them have PC2 near 0 (inside the circle).

5. CONCLUSIONS

We have introduced the popular PCA technique into the hyper-heuristic arena. We found that PCA can help us to characterize the BPP and relate some feature combinations with hyper-heuristic performance. The BPP has a complex structure. There are not simple rules about the relation between features and algorithm performance. It may be necessary to consider feature combinations in order to have insight into hyper-heuristic performance. This contrasts with other combinatorial optimization problems. For example, in the constraint satisfaction problem, a couple of well selected features (density and tightness) are enough to predict which of two heuristics will be the best [3].

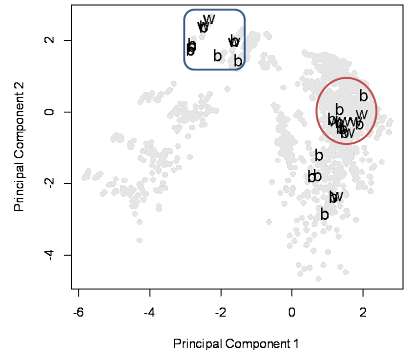

Figure 2: Letter b: instances solved with fewer objects by the best hyper-heuristic than any of the 6 heuristics. Letter w: cases where none hyper-heuristic could reach the best single heuristic result. Gray: the best hyper-heuristic got the same number of objects than the best heuristic for each case.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

[1] 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.

[2] E. López-Camacho, H. Terashima-Marín, and P. Ross. A hyper-heuristic for solving one and two-dimensional bin packing problems. In *13th annual conference companion on Genetic and evolutionary computation*, GECCO '11, pages 257–258, New York, NY, USA, 2011. ACM.

[3] J. C. Ortiz-Bayliss, E. Ozcan, A. J. Parkes, and H. Terashima-Marín. Mapping the performance of heuristics for constraint satisfaction. In *IEEE Congress on Evolutionary Computation'10*, pages 1–8, 2010.

[4] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[5] M. Ringner. What is principal component analysis? *Nat Biotechnol*, 26(3):303–304, Mar. 2008.

[6] K. A. Smith-Miles, R. J. James, J. W. Giffin, and Y. Tu. In T. Stützle, editor, *Learning and Intelligent Optimization*, chapter A Knowledge Discovery Approach to Understanding Relationships between Scheduling Problem Structure and Heuristic Performance, pages 89–103. Springer-Verlag, Berlin, Heidelberg, 2009.

[7] W. X. Wang. Binary image segmentation of aggregates based on polygonal approximation and classification of concavities. *Pattern Recognition*, 31(10):1503–1524, 1998.