Algorithm for Producing Random Instances that Include Non-convex Pieces

Eunice López-Camacho

Tecnológico de Monterrey,
Av. E. Garza Sada 2501,
Monterrey, NL, 64849 México.
lopezeunice@gmail.com

1 Introduction

This document explains how the problem instance set Terashima2 was generated. The Terashima2 dataset is available at the site
`http://paginas.fe.up.pt/~esicup/tiki-index.php`.

In the Terashima2.zip file package, the _ReadmeT2.txt file explains how data is organized in each instance file. Four published articles have used this dataset up to now [1–4].

2 The algorithm

Our algorithm for producing instances that include non-convex pieces takes as an input a problem instance with convex pieces. We have also designed an algorithm to generate random problem instances with convex pieces (Terashima et al. [5], Section 4.3), which can be run to produce the convex instances this algorithm needs.

Then, the algorithm randomly selects some convex polygons and splits each one into two pieces: one convex and one non-convex polygon. Both, the convex and the non-convex subitems are maintained in the new instance replacing the original convex shape. Besides the problem instance as the main input, the following parameters or restrictions are useful to control the irregularity of the new instances:

- Number of pieces selected to split.
- Minimum length of any edge of the new pieces.
- Maximum internal angle of the new non-convex polygons (this will determine the concaveness).
- Minimum internal angle of any new piece.
- Maximum ratio between the largest and smallest edge of any new piece.

The last two restrictions will affect the irregularity and rectangularity factor. Our implementation requires integers as the pieces coordinates because our solver is programmed this way. Nevertheless, edge lengths are not required to be

integers. This algorithm was designed to produce the 480 non-convex instances of our experimental testbed.

The outline of the algorithm is as follows:

1. Randomly choose a convex piece to split and select two points $Q \neq R$ with integer coordinates somewhere on the boundary of the shape.
   This is done by selecting two different edges from the original piece and choosing a point inside each one of them. To ensure that a point on the edge E with vertices (x_1, y_1) and (x_2, y_2) has integer coordinates the following process is done: First, choose an integer value x in the range x_1 to x_2, then find the coordinate y on edge E with horizontal coordinate x. If the correspondent y value is not integer, try again choosing a new x. When the algorithm has failed 100 times to find a random point with integer coordinates on edge E, it selects one out of the two vertices of the edge. Finding a point with integer coordinates on an edge may be difficult or impossible for some sloping edges. In such cases, a vertex is selected.
2. Randomly choose a point P inside the piece.
   Here is the process to choose a point inside a shape (Fig. 1):
   (a) Select a point on a random edge.
   (b) From the selected point draw a ray that crosses the shape in either vertical or horizontal direction. If both directions are possible, select randomly.
   (c) Select a point on the ray such that it is inside the shape.

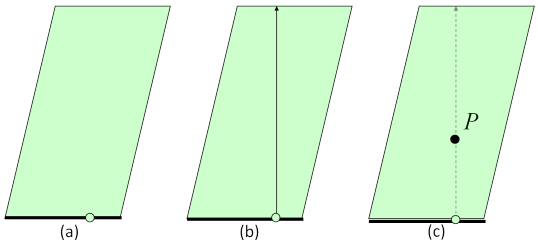

Fig. 1. Procedure for selecting a point inside a shape.

   The cut QPR separates the original shape into a convex shape and a non-convex shape (Fig. 2).
3. Check that these two new shapes satisfy the desired restrictions. If this is the case, replace the original selected piece by the two new pieces. If not, start the algorithm again.

Finally, the algorithm randomizes the order of all the pieces to prevent that the two parts of a split piece end in consecutive positions. Fig. 3 shows how a 30-piece instance whose optimum is 3 objects has been transformed into a 35-piece instance with 5 nonconvex pieces using the developed algorithm.

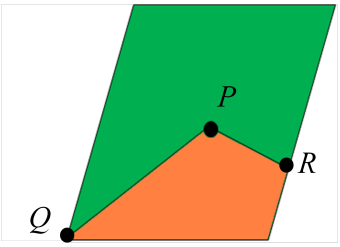

Fig. 2. One convex and one non-convex polygon created by the developed algorithm.

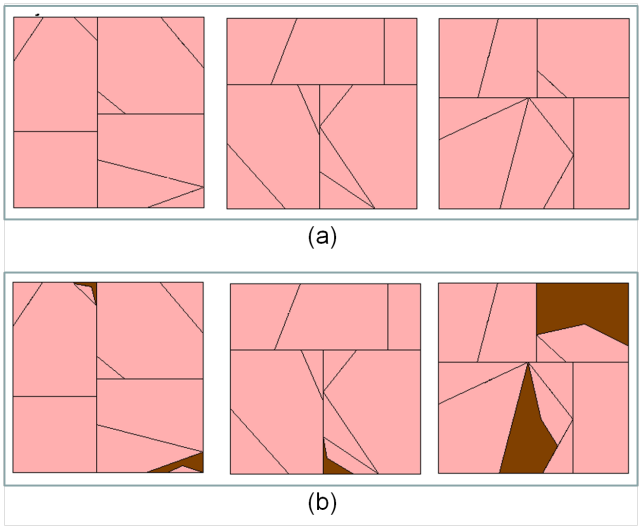

Fig. 3. (a) A convex problem instance. (b) 5 pieces were randomly selected to build 5 non-convex polygons.

When this algorithm is applied to an instance that already has non-convex shapes it is possible to produce U-shaped polygons or even shapes with an internal empty space reached by a smaller entrance (resembling letter G). Shapes with holes are not produced by the proposed algorithm. Types *NConv U*, *NConv W* and *NConv X* from the Terashima2 testbed were produced by introducing convex instances twice into the presented generator algorithm.

Acknowledgments

This research was supported in part by Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) under the Research Chair CAT-144 and the Consejo Nacional de Ciencia y Tecnología (CONACYT) Project under grant 99695.

References

1. López-Camacho, E., Terashima-Marín, H., Ochoa, G., Conant-Pablos, S.E.: Understanding the structure of bin packing problems through principal component analysis. International Journal of Production Economics. Special Section on Cutting and Packing. (2013)
2. López-Camacho, E., Terashima-Marín, H., Ross, P.: A hyper-heuristic for solving one and two-dimensional bin packing problems. In: 13th annual conference companion on Genetic and evolutionary computation. GECCO '11, New York, NY, USA, ACM (2011) 257–258
3. López-Camacho, E., Terashima-Marín, H., Conant-Pablos, S.E.: The impact of the bin packing problem structure in hyper-heuristic performance. In Soule, T., Moore, J.H., eds.: GECCO (Companion), ACM (2012) 1545–1546
4. López-Camacho, E., Terashima-Marín, H.: Evolving feature selection for characterizing and solving the 1D and 2D bin packing problem. In: IEEE Congress on Evolutionary Computation. (2013) 2094–2101
5. Terashima-Marín, H., Ross, P., Farías-Zárate, C.J., López-Camacho, E., Valenzuela-Rendón, M.: Generalized hyper-heuristics for solving 2D regular and irregular packing problems. Annals of Operations Research **179** (2010) 369–392