Batchless Layout Optimization Used in the Glass Tempering Process

Yuanli Wang
HP3 Software Inc.
Developing Department
2591 Wexford Bayne Road, Suite 208
Pittsburgh, PA, USA

Abstract: This paper introduces a batchless optimization algorithm used for glass tempering process. The input of the application is a set of arbitrary geometric configurations in two-dimension. The shape of the geometries may consist of lines or curves. There are two types of optimization in this application: *Sequencer Optimization*, which grantees that the input geometries (or glasses in this application) are selected in an efficient manner. and *Bed Optimization*, which lays the items in the bed of furnace with using the space as more as possible. The main features of this application include: 1) it optimizes the furnace layout in a batchless pattern. In another word, the input set of data could include an unlimited set of geometries. 2) the type of input data is not restricted to rectangle shapes, it can be extended for any arbitrary 2D shape optimization. 3) this application nests certain types of shape together to save space. Some sample outputs are presented at the end of the paper.

Key–Words: 2D Bin Packing, Cutting, 2D Layout Optimization
September 2007

1 Introduction

The purpose of this paper is to present a batchless optimization algorithm used in the glass industry, specifically developed for tempering process. After all the glass are cut into pieces and washed, they are sent through a tempering furnace. The furnace exposes the pieces to high and low temperature air (alternating) in order to quench the glass. This process is call *Tempering Process*. The purpose of this tempering process is to make the glass stronger and therefor harder to break.

The problem existed in the tempered line is that the furnace is not efficiently used if the glasses are not well placed on the furnace bed. The traditional scenario of producing tempered glasses is like this: one operator manually picks up glasses to be tempered from the racks (the places where the glasses are stored) and fills them on the furnace bed in a sequential order, meanwhile, the other operator stands at the end of the furnace and lays the glasses back to the racks after the glasses are tempered.

Instead of letting the operator manually and sequentially choose the glasses to be tempered, this present work introduces an optimization tool for glass tempering process through two types of optimization algorithm: 1) we first optimize the order of glasses being tempered in a more efficient manner, the glasses which have higher priorities will be chosen first. This is called *Sequencer Optimization* in this work. 2) we then optimize the layout of the furnace bed. This is called *Bed Optimization* in this work. Through the use of this powerful tool, the tempering furnace operators will spend significantly less time figuring out how to build a tempering load. This software maximizes the efficiency the yield for the usage of furnace, thus it saves the energy usage and reduces manpower.

This paper is organized as follows: some concepts used in this optimization algorithm are introduced in Section 2; in Section 3, we present the optimization strategies used in this algorithm. Some sample output generated by the software are illustrated in Section 4; and finally, Section 5 closes the paper.

2 Definitions

This section presents some concepts used in the application, *batchless*, *rotation*, *edge flip*, *bed size*, *spaceing between glass*, *priority*, and *nested shape*.

2.1 Batchless

A standard batch optimization requires that all the ordered items be partitioned into smaller, more usable quantities or batches. Once a batch is created, it is difficult to change and must be run in its entirety before beginning the next process. Any defective or broken pieces then have to be added into yet another batch run - these runs almost always suffer poor yields.

Batchless optimization eliminates the notion of a production batch. Our patented batchless optimization process incorporates many lean manufacturing principals. We no longer have to wait for the entire batch to complete to move onto the next process. We control biasing of the harp rack, A-frame or other storage medium to get the first rack completed while at the same time filling the other racks. This means there is now glass for the next process to begin working on. Since we are no longer limited to a fixed batch, the schedule of production pieces can be updated at any time and can be rearranged to reflect the ever changing production needs of the glass industry. Since each optimization occurs in real-time, any defective pieces can be added back in to the production run with full optimization yields. This eliminates the need for separate runs for defective pieces.

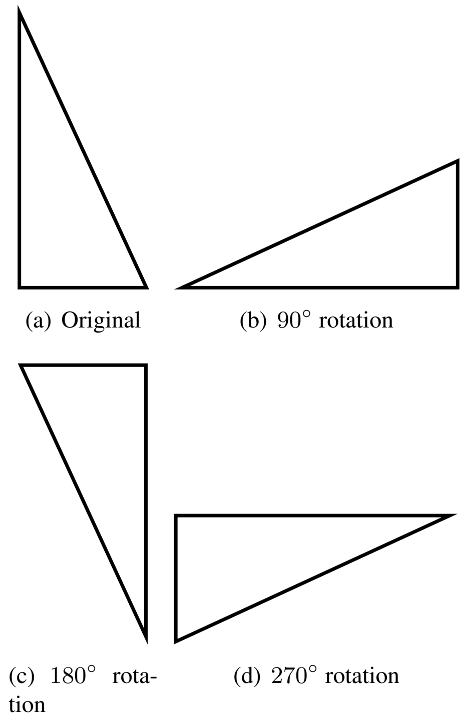
Our batchless process always ensures that a continuous flow of work is permeating through the facility making maximum use of the equipment and personnel, saving both time and money.

2.2 Rotation

This algorithm supplies *rotation* operation for each geometric configuration (see Figure 1). The rotation is performed along the counter-clockwise direction. Figure 1(a) is the original geometric definition. Figures 1(b) to 1(d) illustrate the definition of 90° , 180° and 270° rotation respectively. Not all the glasses are permitted for orientation. For example, if the glasses have roller wave requirement, then rotation operation is forbidden for this particular item, and only the original geometric configuration can be used.

2.3 Edge Flip

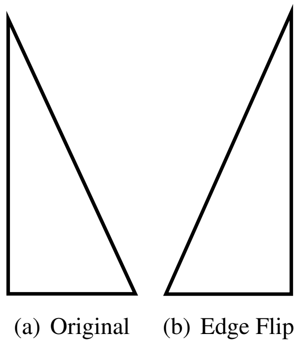
Figure 2 defines the *Edge Flip* operation. Figure 2(a) is the original configuration, Figure 2(b) is the geometry after edge flip operation. This type of operation is not always allowed, for example, if the glass is coated glass, then edge flip is not allowed during bed optimization.



(a) Original (b) 90° rotation

(c) 180° rotation (d) 270° rotation

Figure 1: Rotation definition



(a) Original (b) Edge Flip

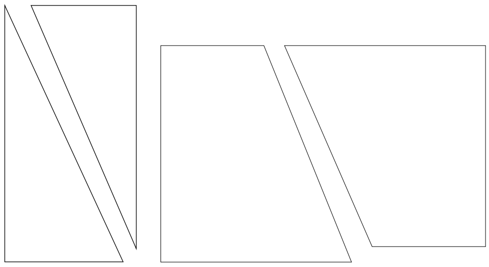
Figure 2: Edge Flip definition

2.4 Bed Size

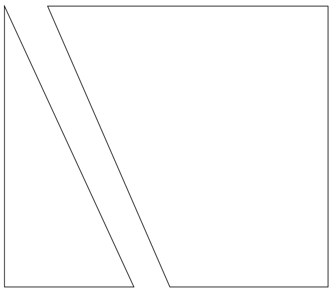
The effective sizes of the load bed is input by the operator, and varied from different furnaces.

2.5 Spacing Between Glasses

The space to be left between pieces of glass are specified based upon the thickness of the glasses being loaded. Also, the profile of the furnace can influence the space between items as well. Adequate spacing insures that items do not accidentally collide as they are conveyed over the rollers.



(a) Nested Shape 1 (b) Nested Shape 2



(c) Nested Shape 3

Figure 3: Nested Shapes

2.6 Priority

Each glass has its own priority. The glass with higher priority is always desired to be processed as soon as possible.

2.7 Nested Shape

There are three types of shape nesting operations defined in this application, which are shown in Figure 3. Figure 3(a) and 3(b) define the same type of shape nesting operation. Figure 3(c) defines different types of shape nesting operation. The dimensions of the shapes are not necessarily the same.

3 Optimization Strategies

The input of the software is a set of pieces of glass (see Figure 4), the shape of the glass could be regular (rectangles or squares only) and irregular (any other types of shapes) shapes. The output is the optimized 2-dimensional layout with best yield (lowest waste), meanwhile, the space between glass pieces is generated during optimization (see Figure 5). The procedures of the application is as follows:

1. Load input

   The information for the glass pieces to be tempered is loaded from the database into the system. The input information include: the location of the glass in the input rack and output rack, the

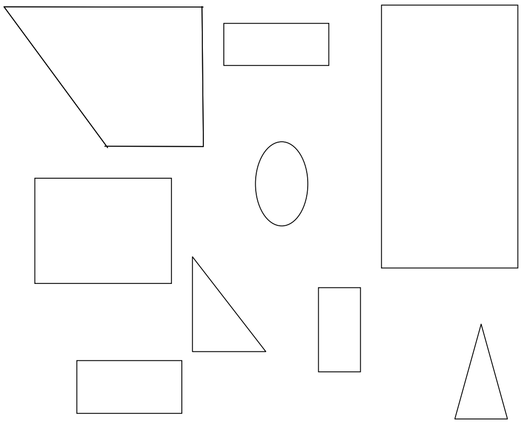


Figure 4: Input of the application

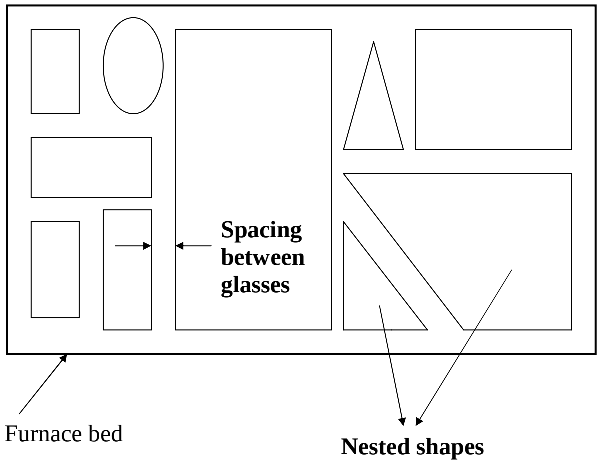


Figure 5: Output of the application

the dimensions, the type, the thickness, the priority, and some other relative information.

2. Glass sequencer selection

   Sequencer selection algorithm chooses the pieces to be processed in an efficient manner. This is the sequencer optimization algorithm, which is introduced in Section 3.1.

3. Initialization

   This step sorts the input data and stores them into the proper data structure.

4. Bed optimization

   This algorithm generates the layout with high yield to maximize the usage of the furnace bed, which is introduced in Section 3.2.

5. Layout displaying

   The resulting layout generated by the bed optimizer is graphically displayed on a large TV monitor at the load end to assist the loading personnel. For each cycle of the furnace, the television shows the position of each glass piece stored in the rack and the position on the bed. A second monitor placed at the Off-Load end of the furnace displays the same layout with new rack assignments.

This application involves two types of optimization algorithm, *sequencer optimization* and *bed optimization*.

3.1 Sequencer Optimization

Sequence optimization is the algorithm to efficiently choose the pieces which have high priority levels but also not violate the best yield principle.

It is very importance to always generate a layout for the furnace with the best yield, since it will greatly improve the tempering process time. However, there are another two facts need to be considered and these may effect the selections: 1) the processing temperature and time in the furnace are determined by the glass type, and glass thickness, each layout for the furnace must always pick the same type of glass pieces, that means, if the temperature and processing time have been set by the operators, we can only choose the pieces which have the same glass type and thickness from the input container, 2) All the pieces of glass to be tempered are stored in the racks, and difference rack has different priority levels, thus when we select prices, we always want to choose from the highest priority rack if it is possible.

In this sequencer selection algorithm, the key technique applied to solve the confliction of the "best yield" and "priority" is to use the *balance coefficient*, which balances the weight of influences between the "best yield" and "priority". Its value varies from "0" to "1". "0" represents that we will only consider to generate the best yield when choosing pieces from the input container, so the pieces of tempered glass may be chosen from the low-priority racks; "1" represents that we will first choose the available pieces from the rack which has the highest priority until we run out of all the prices in the first rack, and then we start from the second priority rack, and etc. User can change the value of ""balance coefficient" to finish the high priority racks as early as possible but also generate a layout with best yield.

3.2 Bed Optimization

In this bed optimization problem, the input is an unlimited set of two-dimensional geometries, regular shapes or irregular shapes, each one has width and height. The requirement of the application is to allocate, without overlapping, all the items to the minimum number of bins, the size of bins is the size of the furnace bed. The edges of the items should parallel to the furnace bed. For each item, if user does not have restrictions for rotation and edge flip operations being predefined, then by default, it is assumed that each piece can be performed with 90°, 180° and 270° rotations, and edge flip operation.

In the present work, the bed optimization is viewed as an variant of the *2 Dimension Bin Packing* problem. Bin Packing problems have been proven to be NP-hard and as such many researchers have concentrated on finding approximation algorithms [1]. There are two types of solutions for solving this problem: *Strip Packing* and *free Packing*.

The *strip packing* solution is also called *level algorithms* or *two phase algorithms*. This solution first starts by placing all the items into a single strip from left to right, in rows forming levels, and then, all the strips are used to construct a packing into finite bins. The first level is the bottom of the strip. There are three classical strategies (First-Fit Decreasing Height (FFDH), Next-Fit Decreasing Height (NFDH) and Best-Fit Decreasing Height (BFDH)) derived from the one-dimensional bin packing case. In each case, the items are initially sorted by non-decreasing height and packed in the corresponding sequence. [5] presented a Hybrid First-Fit (HFF) algorithm. He first packed the strips by the FFDH algorithm, and then he applied the First-Fit Decreasing (FFD) algorithm to pack an item to the first bin that it fits or start a new bin otherwise. Similar to the work of [5], [6] and [4] presented the Hybrid Next-Fit (HNF) and the Hybrid Best-Fit (HBF) ideas respectively. [2] and [3] introduced a Floor-Ceiling (FC) algorithm which is derived from the Best-Fit Decreasing algorithm.

Free packing directly pack the items into the finite bins. Lodi et al. [7] proposed an alternate directions (AD) algorithm which first opens L bins (L being a lower bound on the optimal number of bins required). AD first packs to the bottom of these L bins a subset of items using BFD. The remaining items are packed, one bin at a time, into bands, alternatively from left to right and from right to left. When no item can be packed in either direction in the current bin, the next existing bin or a new empty bin becomes the current one. Loh[8] presented a similar idea which is called Weight Annealing-Based Algorithm to pack the items into bin.

This work first constructs bounding box for irregular shapes, and then nests the predefined types of shapes together, finally, we use the strip packing algorithm to optimize the glass items according the bed size.

4 Result

Figures 6 to 12 are the sample outputs produced by the application.

Figure 6 and 7 display the output when input data combines irregular shapes (which are not rectangles); Figure 8 and 9 illustrate the layout which only combine the rectangle shapes; Figure 10, 11 and 12 show that the predefined types of shapes are correctly nested, the dimensions are not necessarily the same, as shown in Figure 12. Also, in Figure 12, the types of the two nested shapes are not the same. Figure 12 presents the example output for edge flip operation. The shape $A1$ is flipped along the right edge, and its output is the shape $A2$ in this figure. In Figure 10, the two nested triangles are rotated with 90° to generate a high yield layout.

5 Conclusion

The present work provides a way of maximizing bed coverage, so more material finished in less time in the glass tempering processing line. The cost of producing tempered glass is decreased by reducing energy usage and manpower.

Future research directions include 1) investigate the possibilities of generating better optimization result by using the free cutting pattern [8]; 2) support more different shape combinations to further save the space and improving the yield of the optimization result; 3) improve the equation of space calculation between glass pieces.

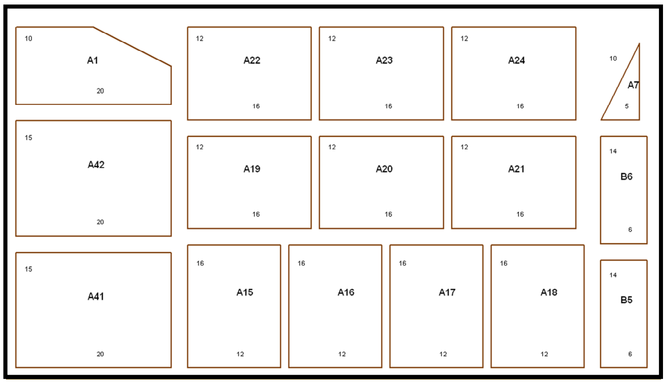


Figure 6: **example output 1**

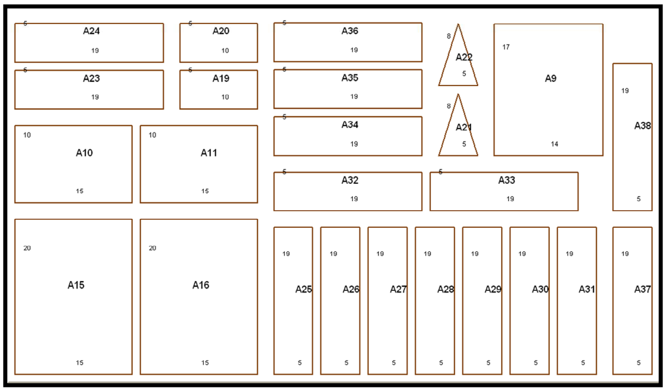


Figure 7: **example output 2**

References:

[1] E.G. Co1man Jr., M.R. Garey, D.S. Johnson, and R.E. Tarjan, Performance bounds for level-oriented two-dimensional packing algorithms, *SIAM J. Comput.*, ,1980, 9, pp.801Ű826.

[2] A. Lodi, S. Martello, and D. Vigo, Neighborhood search algorithm for the guillotine non-oriented two-dimensional bin packing problem, *Meta-Heuristics: Advances and Trends in Local Search Paradigms for optimization, Kluwer Academic Publishers, Boston*, pp.125-139. 1998.

[3] A. Lodi and S. Martello and D. Vigo. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, (11) pp.345-357, 1999

[4] J.O. Berkey and P.Y. Wong. Two dimensional finite bin packing algorithms. *Journal of Operational Research Society*, 2 pp.423-429, 1987

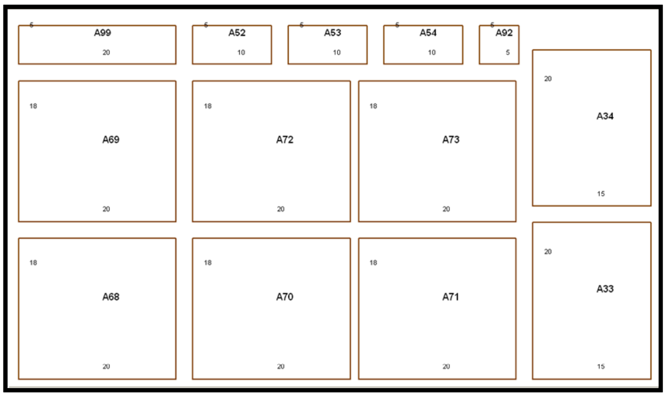


Figure 8: **example output 3**

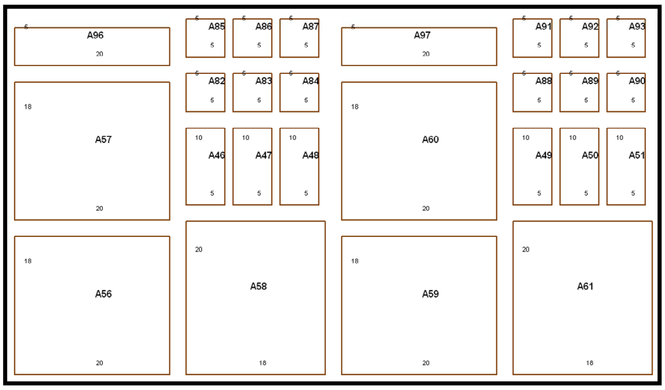


Figure 9: **example output 4**

[5] F.K.R. Chung, M.R. Garey, and D.S. Johnson. On packing two-dimensional bins. *SIAM J. Algebraic Discrete Methods*, 3 pp.66-76, 1982.

[6] J.B. Frenk and G.G. Galambos. Hybrid next-fit algorithm for the two-dimensional rectangle bin-packing problem. *Computing*, 39 pp.201-217, 1987

[7] A. Lodi and S. Martello and D. Vigo. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11:345–357, 1999

[8] K.H. Loh, Weight Annealing Heuristics for Solving Bin Packing and other Combinatorial Optimization Problems: Concepts, Algorithms and Computational Results, *PhD thesis*, 2006, University of MaryLand

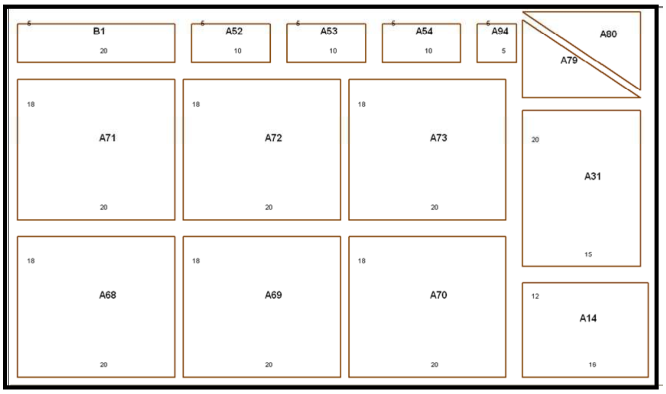


Figure 10: **example output 5**

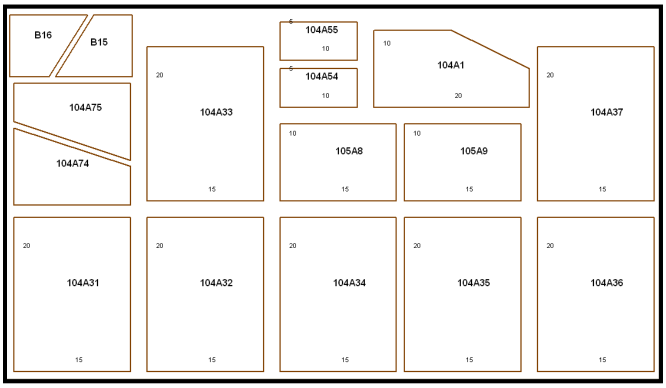


Figure 11: **example output 6**

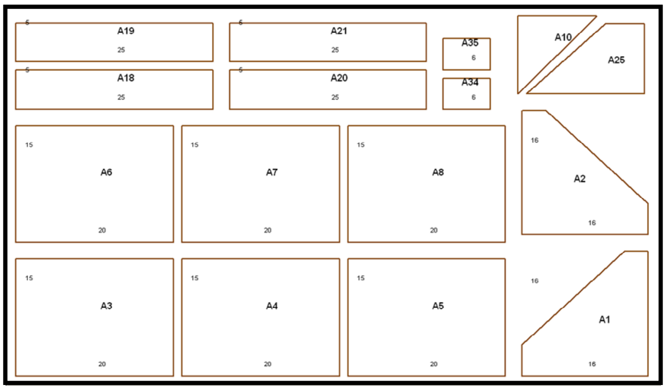


Figure 12: **example output 7**