OPTIGLASS

Application of Artificial Intelligence-based techniques for optimizing the continuous Glass Cutting Problem

Description

In automated float glass manufacturing, a continuous ribbon of glass is cut according to customer orders. This process involves many tasks that must be optimized to improve profitability. This industry is considered an energy intensive industry, so the main objective is to maximize the production of glass with the given energy consumption. Thus, the profitability of applying intelligent techniques to optimize problems associated to this process is twofold: the company remains more competitive and also it is more sustainable.

Float glass manufacturing is a continuous process whereby a ribbon of molten glass is produced in a furnace and then cooled on a bath of molten tin to ensure flatness. The stream of glass is pulled along the top of the molten tin by haul-off conveyors at the end of the float area which transport the glass into the annealing lehr. Then the sheet of glass is cut according to customer requirements, and it is offloaded for storage and distribution.

In this project, we focused on designing and transfer Artificial Intelligence-based technology to solve interrelated scheduling problems involved in a glass manufacturing company. These problems can be summarized in:

> System 1.- The glass cutting problem. The main goal of this problem is to generate the sequence of sheets of the input customer orders to minimize the layout cullet.
>
> System 2.- The customer order selection problem. This problem aims to select, from the batch of all customer orders, the more appropriate ones to be inserted into the system. The selected orders will be served and they will be the input of the glass cutting algorithm.

To this end, a simulation tool has been developed to solve the glass cutting problem and to analyze the best customer order selection problem.

Benefits

The tool developed in this project has reduced the cullet of the company and increased the competitiveness of the company in the market. The company ensures that they save around 150,000€/year.

TRL

The Technology readiness levels (TRL) of the developed tool is:

Level 5 – Component and/or Breadboard Validated in Simulated or Realspace Environment

Simulator: System 1: http://gps.webs.upv.es/vidrio/simulador1/index.html

The information reported by the inspection scanners to detect flaws in the glass is stored and analyzed. Given this information, our tool carried out a preprocessing step to obtain feasible sections of glass that can satisfy the customer requirement according to quality constraints. Thus, the developed algorithm determines the best glass sheet that minimizes the cullet. This algorithm must search for all sheet possibilities to decide which sheet must be selected to be cut. Figure 1 shows the flow of the developed algorithm to minimize the cullet in the glass cutting problem. It is necessary that the algorithm finishes in a given time due to the continuous glass manufacturing process cannot stop. Thus, the algorithm includes a heuristic search technique to obtain anytime optimized solutions in order to be valid for all thickness ranges. It must be taken into account that as the glass thickness is smaller, the speed of the float glass is greater. The search algorithm analyzes all possible combinations of sheets and selects the best one in the given time.

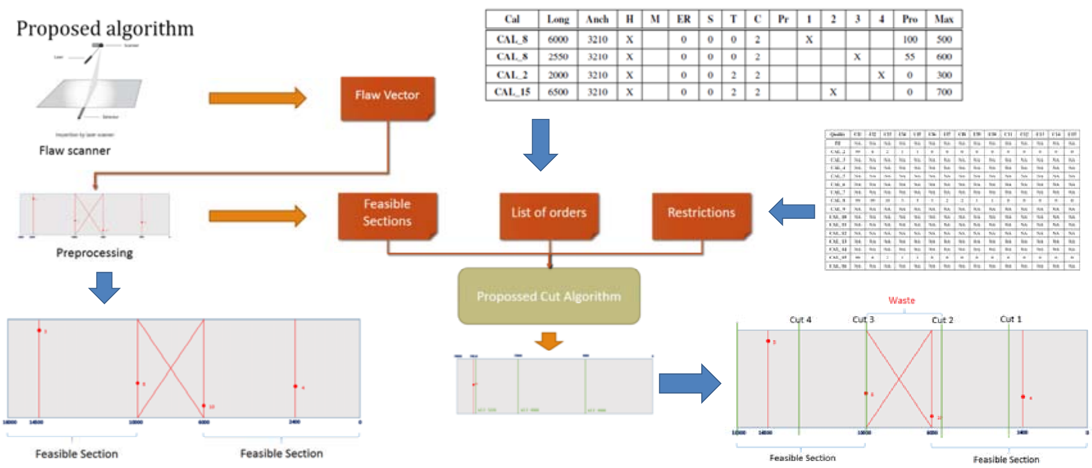

Figure 1: The proposed scheduling algorithm to optimize the glass cutting problem.

In the project, the proposed algorithm (UPV) was compared with the existing algorithm (Bottero Algorithm) to analyze the behavior of both techniques and the achieved improvement. Figure 2 shows the behavior of both techniques with different order sets.

	Order Set 1			Order Set 2			Order Set 3		
	Propossed	Greedy	Improvement	Propossed	Greedy	Improvement	Propossed	Greedy	Improvement
dataset 1	100	122	18,03	63	80	21,25	28	54	48,15
dataset 2	77	103	25,24	54	85	36,47	38	58	34,48
dataset 3	72	87	17,24	44	53	16,98	22	35	37,14
dataset 4	98	113	13,27	45	76	40,79	31	49	36,73
dataset 5	81	117	30,77	52	78	33,33	30	58	48,28
		Improvement Average	21%		Improvement Average	30%		Improvement Average	41%

Figure 2: Evaluation and comparison of the proposed algorithm and the greedy existing one.

It can be observed that the improvement is significant in all the analyzed sets. However the improvement depends on the topology and distribution of flaws and also the customer order selection. As the customer orders are more heterogeneous, less cullet will be generated. Thus, our tool is able to select form a given set of customer orders which four will be managed together to minimize cullet.

In the provided link it can be seem a demo of our system 1 with the following features:

- Simulation Files: in the system, the user can select the quality table and the set of orders. This information is static in the demo.
- Simulation Parameters: in the system and demo, the user can select the simulation speed, the glass speed, the minimum waste to cut the sheets and the simulation mode.
- Flaw Generator: in the system, the user can select the flaw generation ratio and the distance between the scanners and the cutting zone (in millimetres)
- Algorithm Selection: both algorithms are selected to be compared.

Once you begin simulation, the user can see two boxes, the left one corresponds to Bottero system and the right one corresponds to our system.

In this screen the user can see the glass produced, the glass wasted and the efficiency ratio of both systems.

The efficiency ratio depends on the flaw distribution.

Simulator: System 2. http://gps.webs.upv.es/vidrio/simulador2/index.html

The system 2 (see Figure 3) is able to select the set of orders to be inserted into the system to minimize the cullet. Initially, the user selects the set of orders to be included, and the system 2 is able to recommend the next orders to include in the cutting process, according to the properties of the current orders and the available stackers. This system is able to simulate the execution of all orders or simulate the execution step by step.

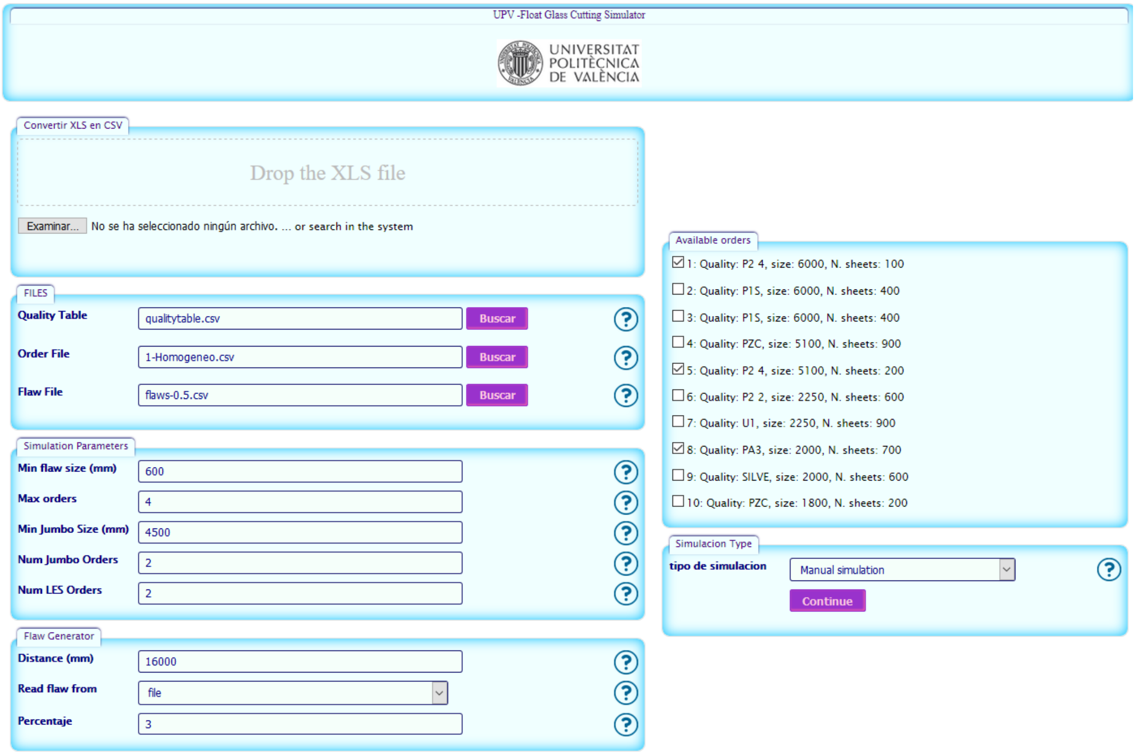

Figure 3: Tool for simulating the orders in insert into the system.

For instance, if the manual simulation is selected, the user selects the orders (1,5,8 for instance), and the system is executed returning the information given in Figure 4.

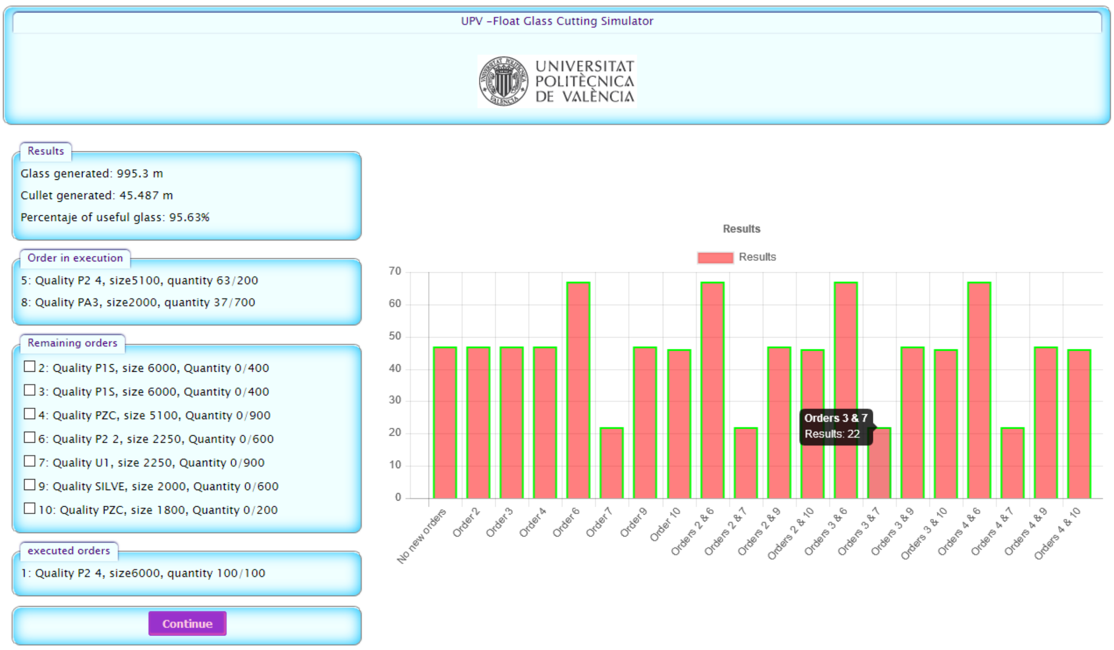

Figure 4: First step of manual simulation.

Once an order is finished (order 1), the system returns the information about the state of the orders in execution (order 5 and 8) and simulates the amount of cullet that will be generated if a new order or set of orders are added into the system. Thus, the user select a new order/s and continue for the next step.

If the automatic simulation is selected, the system returns the preferences of execution of all orders and the cullet generated by our algorithm and the Bottero algorithm (Figure 5).

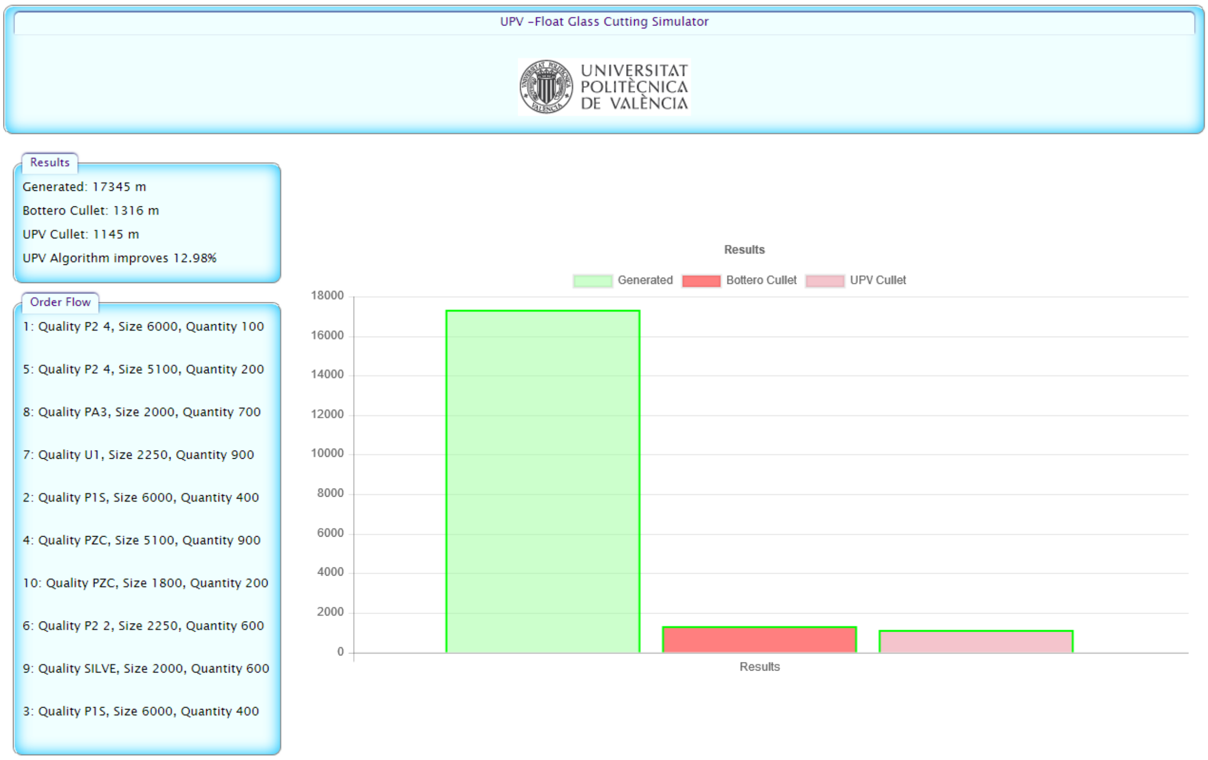

Figure 4: Results obtained with automatic simulation.