Genetic Algorithms Approaches for the Production Planning in the Glass Container Industry

F.M.S. Amorim[1], M.S. Arantes[1], C.F.M. Toledo[1], P.E. Frisch[3], J.S. Arantes[1] and B. Almada-Lobo[2]

[1] *University of São Paulo;* [2] *University of Porto*
[3] *Frisch Verrier Industry*

Introduction

The production process in a Glass Container Industry (GCI) is usually composed by two main stages. In the first stage, the components constituting the glass as grit, kelp, limestone, oxides and glass recyclables are melted by furnaces. The final products (containers) are produced by molding machine in the second stage. The present work is motivated by the installation of a new furnace in a single production plant and evaluates decisions related to the configurations of machines. A machine is connected to a single furnace from which the glass paste is received. Moreover, a furnace can feed multiple machines connected to it. The configurations of machines need to be defined following the demand forecast within a time horizon. This problem will be named GCI Problem with a New Furnace (GCIP-NF).

Mathematical Model

Parameters:

- m: Machines available ($m = 1, ..., M$).
- i: Products to be manufactured ($i = 1, ..., I$).
- a: Annual Time horizon ($a = 1, ..., A$).
- NS_m: Number of sections by machine m.
- TG_m: Type of gob by machine m.
- AC_{im}: 1 if product i is accepted in the machine m.
- C_m: Cost to install machine m. ($)
- D_{ia}: Demand expected of product i in period a. (ton)
- W_i: Weight of product i. (ton)
- R_i: Efficiency of the cavity for product i. (bottles/min)
- $\overline{M}$: Maximum machines supported by the new furnace.
- CF: Cost to install fuse capacity on furnace. ($/ton)
- η_m: Efficiency of machine m. (%)

Variables:

- KF: Melting capacity required for the furnace. (ton)
- Q_{ima}: Lot size of product i on machine m in the period a. (ton)
- F_{ima}: Time spent on period a in which machine m was dedicated to produce product i. (years)
- $\overline{Y}_m$: 1 if the machine m is installed, 0 otherwise.

Formulation:

$$Min \quad f(KF, \overline{Y}_1, ..., \overline{Y}_M) = CF * KF + \sum_{m=1}^{M} C_m.\overline{Y}_m \tag{1}$$

Subject to:

$$\sum_{m=1}^{M} \overline{Y}_m \leq \overline{M} \tag{2}$$

$$F_{ima} \leq \overline{Y}_m \forall (i, m, a) \tag{3}$$

$$F_{ima} \leq AC_{im} \forall (i, m, a) \tag{4}$$

$$\sum_{i} F_{ima} = \overline{Y}_m \forall (m, a) \tag{5}$$

$$Q_{ima} = F_{ima}.(R_i.W_i.NS_m.TG_m.\eta_m) \forall (i, m, a) \tag{6}$$

$$\sum_{\tau=1}^{a} \sum_{m} Q_{im\tau} \geq \sum_{\tau=1}^{a} D_{i\tau} \forall (i, a) \tag{7}$$

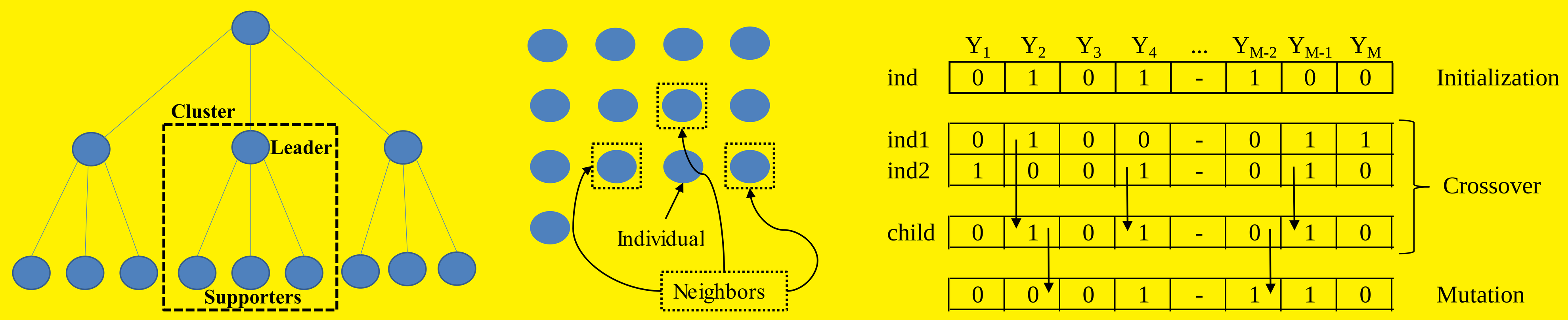
$$\sum_{i} \sum_{m} Q_{ima} \leq KF \forall (a) \tag{8}$$

$$KF, Q_{ima}, F_{ima} \geq 0 \tag{9}$$

$$\overline{Y}_m \in \{0, 1\} \tag{10}$$

Methods

A total of four methods are applied to solve the GCIP-NF: Branch-and-Cut (B&C) algorithm, from commercial solver CPLEX, a simple genetic algorithm (GA) and two multi–populations genetic algorithms with a tree structure (t–GA) and a grid structure (g–GA) [1] as shown by Figures . All genetic algorithms encode the binary variables of GCIP-NF model as individual and the objective function (1) is set as fitness function. Thus, the continuous variables on GCIP–NF are optimally defined by solving the related linear programming model. Figure (C) illustrates individuals and genetic operators.



(A) Tree structure. (B) Grid structure. (C) Individuals and genetic operators.

Results

The mathematical model and GAs are coded using the toolbox ProOF [2] integrated with IBM ILOG CPLEX 12.6. The computational tests are performed on an Intel Xeon E5-2680v2 computer with 2.8 GHz and 128 GB RAM. The methods t–GA, g–GA and GA are set with crossover and mutation rates of 5.0 and 0.7, respectively. A total of 200 instances are created by increasing parameters M and T. The instances are separated into set of instances named as small (SFM and SHT) and large (LFM and LHT) sets.

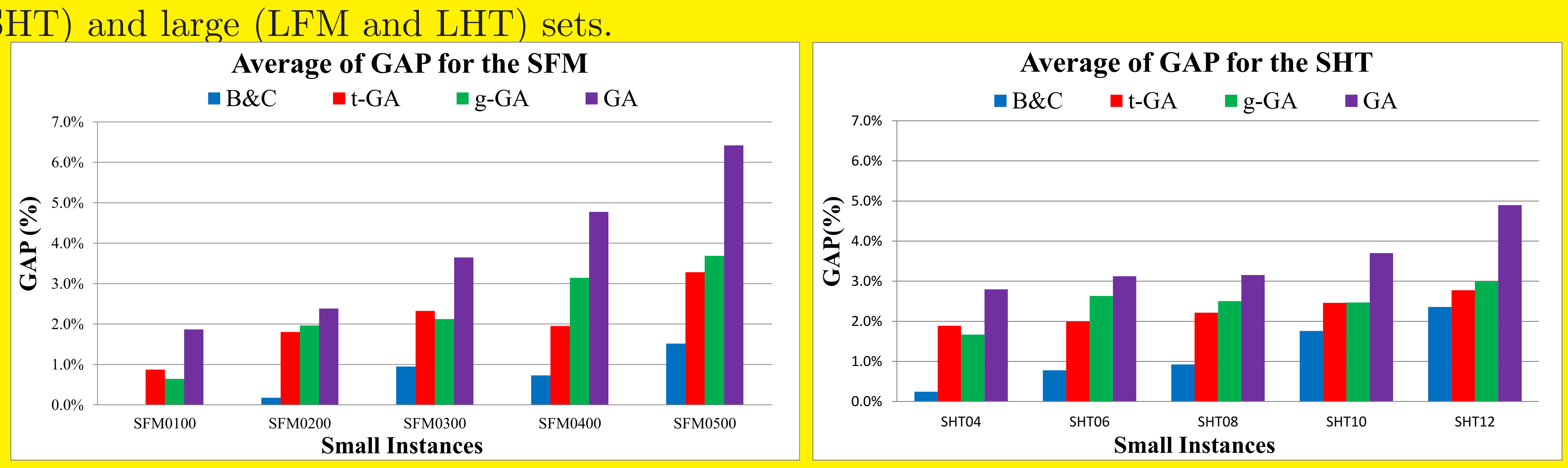


Figure 1: Average GAP results for small instances.

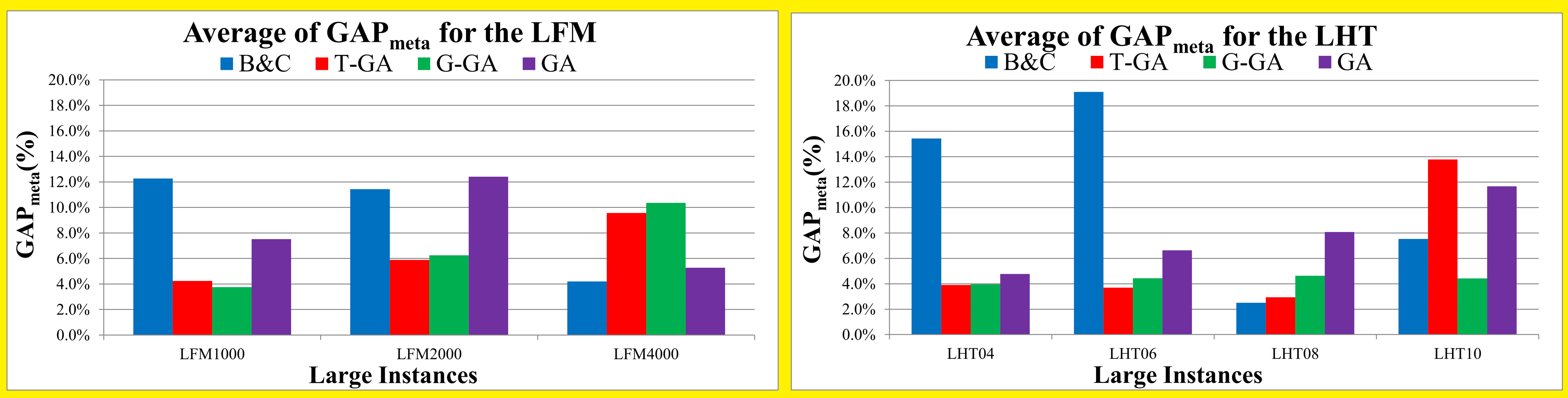


Figure 2: Average results obtained through GAP of large instances.

Conclusion

The exact method solves small instances reaching many optimal solutions. However, it is not able to find optimal or even feasible solutions for many large instances, while t–GA and g–GA returned feasible solutions for all large instances. For the same subset of large instances solved by all methods, there is no significant statistical difference between t–GA and g–GA.

References

[1] C. F. M. Toledo, M. Arantes, M. Y. B. Hossomi, and B. Almada-Lobo, "Mathematical programming-based approaches for multi–facility glass container production planning," *Computers Operations Research*, vol. 74, no. 2016, pp. 92–107, 2016.

[2] M. S. Arantes, *Ambiente para desenvolvimento de métodos aplicados a problemas de otimização.* PhD thesis – Universidade de São Paulo (USP), 2014.