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

Flaviana Moreira S. Amorim
University of São Paulo - USP
São Carlos, São Paulo, Brazil
joflaviana@yahoo.com.br

Márcio da Silva Arantes
University of São Paulo - USP
São Carlos, São Paulo, Brazil
marcio@icmc.usp.br

Claudio Fabiano Motta Toledo
University of São Paulo - USP
São Carlos, São Paulo, Brazil
claudio@icmc.usp.br

Pierre Eric Frisch
Frisch Verrier Industry
Santa Isabel, São Paulo, Brazil
pierre_eric_frisch@hotmail.com

Jesimar da Silva Arantes
University of São Paulo - USP
São Carlos, São Paulo, Brazil
jesimar.arantes@usp.br

Bernardo Almada-Lobo
University of Porto
Porto, Portugal
almada.lobo@fe.up.pt

ABSTRACT

This paper applies three genetic algorithms, combined with mathematical programming technique, to solve a production planning problem in the Glass Containers Industry (GCI). The problem to be solved takes into account the scenario where one new furnace and the related machines must be added to the industrial plant. A mathematical formulation is introduced to define objectives and constraints for such problem. The results achieved indicate that the proposed model as well as the genetic algorithms are able to provide good quality solutions for such problem.

CCS CONCEPTS

•Mathematics of computing → Combinatorial optimization; •Theory of computation → Evolutionary algorithms; •Applied computing → Supply chain management;

KEYWORDS

Genetic Algorithm, Glass Container Industry, Mathematical Modeling, Production Planning, Meta-heuristics

ACM Reference format:
Flaviana Moreira S. Amorim, Márcio da Silva Arantes, Claudio Fabiano Motta Toledo, Pierre Eric Frisch, Jesimar da Silva Arantes, and Bernardo Almada-Lobo. 2017. Genetic Algorithms Approaches for the Production Planning in the Glass Container Industry. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076071

1 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 [1]. 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 with a New Furnace (GCIP-NF).

2 MATHEMATICAL MODEL

Parameters:

- m: Machines available ($m = 1, \ldots, M$).
- i: Products to be manufactured ($i = 1, \ldots, I$).
- a: Annual Time horizon ($a = 1, \ldots, A$).
- 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:

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

Subject to:

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

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

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

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

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

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

DOI: http://dx.doi.org/10.1145/3067695.3076071

$$\sum_i \sum_m Q_{ima} \leq KF \qquad \forall(a) \qquad (8)$$

$$KF, Q_{ima}, F_{ima} \geq 0 \qquad (9)$$

$$\overline{Y}_m \in \{0, 1\} \qquad (10)$$

3 METHODS

A total of four methods are applied to solve instances of the GCIP-NF: Branch-and-Cut (B&C) algorithm available in the 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) [3], respectively, as shown by Figure2. 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 other variables on GCIP–NF are optimally defined by solving the related linear programming model. Figure 1 illustrates individuals and operators.

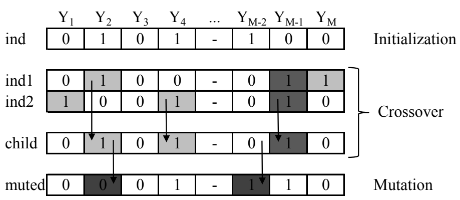

Figure 1: Individuals and genetic operators.

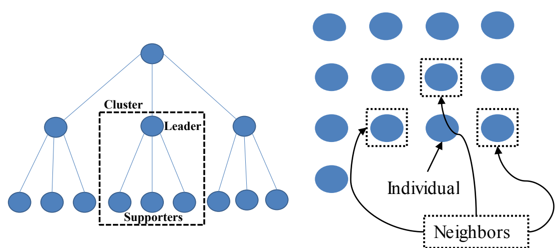

Figure 2: Tree and grid structures.

4 RESULTS

The mixed integer linear programming model proposed and genetic algorithms are coded using the toolbox Professional Optimization Framework (ProOF) [2] integrated with IBM ILOG CPLEX 12.6. The computational tests are performed on a computer with an Intel Xeon E5-2680v2 de 2.8 GHz and 128 GB RAM, and operating system Linux. The methods t–GA, g–GA and GA are set with crossover and mutation rates of 3.0 and 0.7, respectively. The crossover rate means a total of $(crossoverrate) * (populationsize)$ new individuals evaluated at each generation. The population size is 39, where t-GA and g-GA evolve 3 populations with 13 individuals. The methods are evaluated from a total of 200 instances, 50 in each set of small instances SFM and SHT as well as large instances LFM and LHT. The parameters applied to generate SFM instances are $M \in \{100, 200, 300, 400, 500\}$ with $T = 8$ and $M = 300$ with $T \in \{4, 6, 8, 10, 12\}$ for SHT. The parameters for LFM subset are $M \in \{1000, 2000, 3000, 4000, 5000\}$ with $T = 8$, while $M = 3000$ with $T \in \{4, 6, 8, 10, 12\}$ for LHT. The time limit is 3600 seconds to run each method only once. The performance of the exact method is evaluated based on the Upper Bound (UB) deviation from the Lower Bound (LB) achieved by B&C algorithm of CPLEX solver. Figure 3 shows that the exact method has the lowest average GAP among all methods.

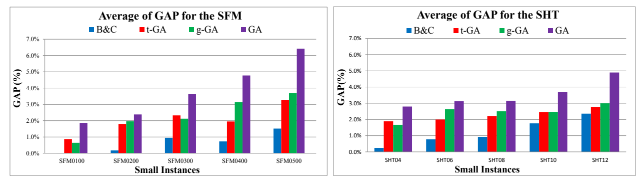

Figure 3: Average results obtained through GAP.

B&C from CPLEX solver does not return optimal solutions for large instances, finding only 26 feasible solutions within the time limit. This situation occurs with all instances in subsets LFM3000, LFM5000 and LHT12. On the other hand, the meta–heuristics return feasible solutions for all large instances. Figure 4 compares the gap values, but only for the subset of instances solved by all methods. The exact method has the lowest average gap only for LFM4000 and LHT08 in Figure 4. The genetic algorithms present better gap values on average. The t-GA is better for LFM2000, LHT04 and LHT06, while g-GA behaves better in LFM1000 and LHT10. The Kruskal–Wallis test was applied since the data set evaluated does not follow a normal distribution by Anderson–Darling test. The results show no significant difference for all set of small instances as well as for all subset of large instances solved by all methods.

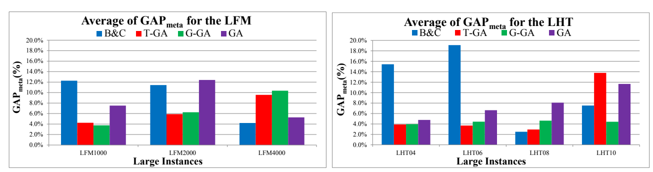

Figure 4: Average results obtained through GAP.

5 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] Bernardo Almada-Lobo, Diego Klabjan, Maria Antónia Carravilla, and José Fernando Oliveira. 2010. Multiple machine continuous setup lotsizing with sequence-dependent setups. *Comp. Opt. and Appl.* 47, 3 (2010), 529–552.

[2] M. S. Arantes. 2014. *Ambiante para desenvolvimento de métodos aplicados a problemas de otimização.* Masters thesis – Universidade de São Paulo (USP/ICMC). https://github.com/marcio-da-silva-arantes/ProOF/wiki

[3] C. F. M. Toledo, M. Arantes, M. Y. B. Hossomi, and B. Almada-Lobo. 2016. Mathematical programming-based approaches for multi–facility glass container production planning. *Computers Operations Research* 74, 2016 (2016), 92–107.