

Grid Multi-Populational Genetic Algorithms applied to Glass Container Industry Problem

Flaviana Moreira de Souza Amorim

Federal Rural University of Semi-Árido

R. Francisco Mota, 572 - Pres. Costa e Silva, Mossoró - RN, 59625-900

flavianamsamorim@gmail.com

Magna Paulina de Souza Ferreira

Maximia Tecnologias

Av. Brig. Faria Lima, 2369 - andar 11 - conj 1102 - Jardim Paulistano, São Paulo - SP, 01452-922

magna.maximiza@gmail.com

Valdemar Abrao Pedro Anastacio Devesse

Santander

valdemar.abrao@alumni.usp.br

Márcio da Silva Arantes

Instituto SENAI de Inovação - Florianópolis

Av. Luiz Boiteux Piazza, 574 - Canasvieiras, Florianópolis - SC, 88054-700

marcio.arantes@sc.senai.br

Claudio Fabiano Motta Toledo

Instituto de Ciências Matemáticas e de Computação

Universidade de São Paulo

Avenida Trabalhador São-carlense, 400 - Centro

claudio@icmc.usp.br

ABSTRACT

In this paper we propose a mixed-integer linear programming (MILP) model and two genetic algorithms to tackle the Glass Container Industry Problem with a New Furnace. We start by providing an MILP formulation to outline the objectives and constraints associated with the problem. The genetic algorithms introduced in our study are designed to evolve the binary variables of the MILP model, and a linear version of the formulation helps find values for the continuous variables. Our methods were evaluated on a series of problem instances that were generated using data sourced from a glass container industry located in Portugal and Brazil. The results indicate that our model and genetic algorithms effectively provide viable solutions to the problem at hand.

KEYWORDS. Glass Container Industries; Mathematical Modeling; Metaheuristics, Matheuristic

Paper topics (indicate in order of PRIORITY the paper topic(s))

1. Introduction

Over the past two decades, the glass container industry has received increasing attention in the OR community, particularly in real case settings. The Glass Container Industry Problem (GCIP) is complex, as it deals with the decision to include a new furnace. The Glass Container Industry Problem is considered a class of the Lot Sizing problem. As in glass container manufacturing, production planning involves continuously operating furnaces and feeding moulding machines with melted glass. The challenge arises because setup changes interrupt the glass flow, leading to heat loss and requiring warm-up processes before production can resume. The decision-maker must address several tactical and operational-level choices and constraints. Decisions are relevant and must be made simultaneously when evaluating costs, including a new furnace.

This problem falls under capacitated lot sizing with sequence-dependent setups, where manufacturers must optimise production schedules while minimising costs associated with setup times, extraction losses, and machine usage.

Copil et al. [2017], Brahimi et al. [2017], Wörbelauer et al. [2019] provided a comprehensive review on lot-sizing problems, which is the broader area of the GCIP. Copil et al. [2017] presents the latest models, solution approaches, and the main advances combining lot sizing and scheduling. Brahimi et al. [2017], in turn, focuses on problem complexity and solution approaches for the single-item dynamic lot-sizing problem, and its extensions, assumptions on demand and resource constraints. Wörbelauer et al. [2019] evaluates simultaneous lot-sizing and scheduling problems, where they review papers addressing the limitation of resources such as raw material, setup tools, and setup operators.

The authors in Auer et al. [2017] report a case study using the Eco-Care-Matrix (ECM) to compare changes in the manufacturing system of a Glass Container Industry (GCI). This comparative study shows an improvement regarding environmental impact and incurred costs. Varsei and Polyakovskiy [2017] reports a supply chain redesign for a case study of the wine industry in Australia, with a general multi-objective mixed-integer programming model being proposed and customised next for the case study. Based on the proposed model, the redesign of the supply chain returned cost reductions of around 15%. Unlike Auer et al. [2017] and Varsei and Polyakovskiy [2017], we do not explicitly consider environmental impact and sustainability when redesigning the manufacturing system in this paper. Instead, we consider the ecological impact as proportional costs. However, similar to Auer et al. [2017] and Varsei and Polyakovskiy [2017], we evaluated the redesign decisions based on effects on short-term production planning, considering lot sizing and scheduling.

Some works deal with the CGI's short- and long-term production process, as reported by (Almada-Lobo et al. [2010]; Toledo et al. [2013, 2016]). However, these papers do not address the problem related to a new furnace installation.

Toledo et al. [2013] proposed a mathematical formulation for the multi-level capacitated lot sizing problem with backlogging, and a hybrid multi-population genetic algorithm (MPGA) is applied to solve it. The modelling improvements in Toledo et al. [2013] allowed solving small-to-medium and medium-to-large size instances using the exact method Branch and Cut from the CPLEX solver. The hybrid MPGA outperforms the deterministic approach, returning better solutions for large instances with reduced computational time.

In this work, we present a mixed-integer linear programming (MILP) formulation for GCIP-NF and two genetic algorithm approaches: a simple genetic algorithm and a multi-population genetic algorithm. Our main contribution is the definition of a multi-population genetic algorithm with individuals disposed in a grid structure. We've significantly decreased the computational time with the proposed approach, without compromising the solution quality.

The paper is organised as follows. Section 2 introduces its mathematical formulation of the GCIP-NF. The evolutionary methods are described in Section 3. Computational experiments are reported in Section 4 and our conclusions follow in Section 5.

2. Glass Container Industry Problem regarding a New Furnace

After ten years, a new furnace replaces an old one or is added anytime during an expansion plan of the existing industrial plant. The production process usually has moulding machines attached to furnaces that must satisfy the demand forecasting within a time horizon. The right choice of the furnace and the set of machines can guarantee better operating margins. Such an option involves a complex decision process to meet the demanded products and seek the best cost-benefit of the investment. The decision highlights the number of combinations for different configurations of machines and furnaces to solve the problem. Thus, when including a new furnace, the decision-maker must address several tactical and operational-level decisions and constraints. This setting is the Glass Container Industry Problem regarding a New Furnace (GCIP-NF), which has its mathematical formulation presented next.

Parameters:

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

Variables:

- KF : Melting capacity required for the furnace. (ton)
- Q_{ima} : Lot size of container i on machine m in the period a . (ton)
- F_{ima} : Fraction of time spent in production of container i by machine m at year a .
- $\overline{Y}_m$: 1 if the machine m is installed, 0 otherwise.

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

s.t.

$$\sum_{m=1}^M \bar{Y}_m \leq \bar{M} \quad (2)$$

$$F_{ima} \leq \bar{Y}_m * AC_{im} \quad \forall(i, m, a) \quad (3)$$

$$\sum_{i=1}^I F_{ima} \leq 1 \quad \forall(m, a) \quad (4)$$

$$Q_{ima} = F_{ima} * (R_i * W_i * NS_m * TG_m * \eta_m) \quad \forall(i, m, a) \quad (5)$$

$$\sum_{\tau=1}^a \sum_{m=1}^M Q_{im\tau} \geq \sum_{\tau=1}^a D_{i\tau} \quad \forall(i, a) \quad (6)$$

$$\sum_{i=1}^I \sum_{m=1}^M Q_{ima} \leq KF \quad \forall(a) \quad (7)$$

$$Q_{ima}, F_{ima} \geq 0 \quad \forall(i, m, a) \quad (8)$$

$$KF \geq 0 \quad (9)$$

$$\bar{Y}_m \in \{0, 1\} \quad (10)$$

The objective function (1) minimises furnace and machine configurations costs. The furnace cost increases as $CF * KF$ increases, giving a greater capacity. In contrast, the costs of each type of machine configuration can lead to a more expensive production process following $\sum_{m=1}^M C_m * \bar{Y}_m$. The constraints (2) define upper bounds for the number of machines attached to a furnace. Constraints (3) deal with the situations in Figure 2, where machine M_1 can only produce containers i_2 and i_3 , and machine M_2 can only produce i_1 . Thus, container i will be exclusively set for production if accepted by machine m (parameter $AC_{im} = 1$). Also, the machine m must be in the furnace with decision variable $Y_m = 1$. If a machine does not accept the container ($AC_{im} = 0$) or the machine is not assigned to the furnace ($Y_m = 0$), production elapsed time is null ($F_{ima} = 0$). As expected, the maximum production time for a container considering annual planning is one year. The constraint (4) describes the capacity equation, in which each machine is limited to 1 year of production. The equation (5) represents the production (Q_{imt}) in tons, considering the time dedicated to producing the products. In this case, we take into account the cavity rate (R_i), the weight (W_i), the number of sessions (NS_m), the type of gob (TG_m), and the machine efficiency (η_m). Multiplying these parameters results in the maximum amount of tons produced on the machine per year. Still, multiplying the result by variable F_{imt} returns the actual quantity produced in period t . The lot sizes must satisfy demands from constraints (6). All production added up to a year a needs to be greater than or equal to the demand added up to the same year. The furnace capacity will be met by constraint (7). The variable domains are stated by constraints (8) - (10).

3. Methods

We propose three methods to tackle the GCIP-NF instances: the Branch and Cut algorithm available in the commercial solver CPLEX, a simple genetic algorithm (GA), and a Multi-Population Genetic Algorithm (MPGA) with individuals in a grid structure. Algorithms 1 and 2 describe the simple genetic algorithm (GA) and the multi-population genetic algorithm (MPGA), respectively.

Algorithm 1: Simple Genetic Algorithm

```

1 begin
2   forall ind on population do
3     initialize(ind);
4     evaluate(ind);
5   repeat
6     select(parents);
7     child ← crossover (parents);
8     λ ← a random value ∈ [0.0 , 1.0];
9     if λ ≤ mutation rate then
10      mutation(child);
11     evaluate(child);
12     add(child on population);
13   until reach(time limit);

```

Algorithm 2: Multi-Population Genetic Algorithms

```

1 begin
2   repeat
3     forall pop on populations do
4       forall ind on pop do
5         initialize(ind);
6         evaluate(ind);
7       organize(pop);
8       repeat
9         for j = 1 to the number of crossovers do
10          select(parents);
11          child ← crossover (parents);
12          λ ← a random value ∈ [0.0 , 1.0];
13          if λ ≤ mutation rate then
14            mutation(child);
15          evaluate(child);
16          add(child on pop);
17        organize(pop);
18      until converge(pop);
19    forall pop on populations do
20      migrate(pop);
21  until reach(time limit);

```

The MPGA applies a grid structure for individuals, inspired by the concept of the evolutionary diffusion algorithm from Eiben and Smith [2015]. A pair of adjacent neighbour individuals is more likely to cross than two individuals separated by long distances, but with a diffusion effect over the whole population. The grid structure works with a more straightforward organisation; when a child is better than a parent, it replaces the worst parent by taking its place in the grid, not requiring a reorganisation.

Each individual has four neighbours in the upper, lower, left and right positions. One of the advantages of this type of structure is the linear computational complexity, $O(1)$, for selecting

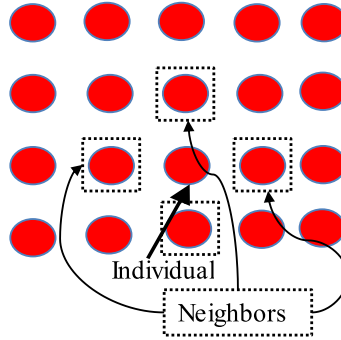


Figure 1: Example of a grid structure.

and adding new individuals. In this case, the relevant information encoded within the child will first spread within its neighbourhood, following a slow diffusion through the population across the generations. Other reproduction selection methods, such as the roulette wheel and the tournament, are more complex in order of $O(n)$ and $O(t)$, respectively, where t is the size of the tournament Eiben and Smith [2015].

3.1. Operators

In the genetic algorithm, we encode the machine installation variables, i.e, the binary variables of the MILP model, as an individual, as depicted in Figure 2. The sequence of the binary variable is randomly initialised, without surpassing the maximum number of machines as described by constraints (2) in Section 2. For instance, the individual chooses machine types Y_2 , Y_4 and Y_{M-2} to be attached to the furnace.

	Y_1	Y_2	Y_3	Y_4	...	Y_{M-3}	Y_{M-2}	Y_{M-1}	Y_M	
<i>ind</i>	0	1	0	1	-	0	1	0	0	<i>Initialization</i>
<i>ind1</i>	0	1	0	0	-	0	0	1	1	} <i>Crossover</i>
<i>ind2</i>	1	0	0	1	-	0	0	1	0	
<i>child</i>	0	1	0	1	-	0	0	1	0	
<i>muted</i>	0	0	0	1	-	0	1	1	0	<i>Mutation</i>

Figure 2: The genetic algorithm operators.

When crossing over two individuals, the cross-over operator first checks the machine installation entries from *ind1* and *ind2*. The new individual (*child*) will keep the common machines from *ind1* and *ind2*. Afterwards, the operator randomly selects the machines available in *ind1* and *ind2*, as the upper bound $\bar{M}$ is achieved. Figure 2 depicts the union and the intersection of entries from the individuals *ind1* and *ind2*. From the union has Y_1 , Y_2 , Y_4 , Y_{M-1} and Y_M entries, whereas the intersection has only Y_{M-1} . The *child* will keep Y_{M-1} and a randomly selected subset from the union, which are Y_2 and Y_4 in this example. The mutation operator randomly selects two entries and changes the values for entries Y_2 and Y_{M-2} .

3.2. Fitness function

The linear programming (LP) model provides the fitness of each individual. The LP model arises by fixing the binary variables of the MILP with binary values from an individual's representation. First, the costs of machines are previous estimated ($Cost_M = \sum_{m=1}^M C_m * \bar{Y}_m$) from the

values of Y_m given by each individual. Next, the LP model is solved where Y_m becomes an input parameter, and the optimum value of the objective function will be the individual's fitness.

$$\text{Min} \quad f(KF) = CF * KF + Cost_M \quad (11)$$

Subject to:

Constraints(3)-(7)

$$KF, Q_{ima}, F_{ima} \geq 0 \quad (12)$$

4. Computational experiments

We coded all the methods described above in Java on ProOF platform developed by Arantes [2014] and performed experiments on an Intel Xeon E5-2680v2 computer at 2.8 GHz with 10 cores and 128 GB of memory. We also used libraries from the IBM ILOG Cplex 12.8 solver for the mathematical model. For every solution approach, we established 3600 seconds of execution time as the stopping criterion. The population has 39 individuals for GA and 13 individuals belonging to each of the three populations in the MPGA.

Table 1: Values of operator parameters

Operator of Parameters				
Methods	Number of Populations	Number of Individuals	Crossover Rate	Mutation Rate
GA	1	39	5.0	0.7
MPGA _g	3	3x13	5.0	0.7

4.1. Problem Instances

The instances used to address the problem are based on data from the Glass Container Industry, which operates in Portugal and Brazil. We generate the instances following a uniform distribution within an interval $U[a, b]$ or a subset of integer values. The only parameter with fixed values is the maximum number of machines attached to a furnace with $\overline{M} = 6$. The $m = 1...M$ represents the total of machine options available in a given market that are being evaluated by the plant's team, and it can be of very high value. Other parameters used are presented next:

- Number of sections by machine: $NS_m \in \{6, 8, 10, 12, 16, 20\}$
- Type of gob by machine: $TG_m \in \{1, 2, 3\}$
- Product accepted by machine: $AC_{im} \in \{0, 1\}$
- Cost to install machine: $C_m \in U[9.0, 11.0] * 100000$ (\$)
- Cost of product: $C_i \in U[0.65, 2.60]$ (\$)
- Weight of product: $W_i \in U[0.067, 1.233]/1000$
- Efficiency of the cavity for product: $R_i \in U[1.0, 5.0] * 60 * 24 * 365/W_i$
- Maximum number of machines: $\overline{M} = 6$
- Cost to install furnace: $CF \in U[500.0, 600.0]$

- Efficiency of machine: $\eta_m \in U[87.8\%, 92.5\%]$

We categorised the instances as Machine-Based (MB) and Horizon-Based (HB) to evaluate the performance of the methods when increasing the two principal parameters of the problem: number of available machines (M) and time horizon (T). In the set of MB instances, we increase the different types of machines available for a fixed time horizon, whereas we increase the time horizon for HB instances from a fixed number of machines. Table 2 shows the main parameters that define the Small-Sized Machine Based (SMB) and Horizon Based (SHB) sets of instances.

Table 2: Set of instances

Instance	M	T	Instance	M	T
SMB100	100	8	SHB04	300	4
SMB200	200		SHB06		6
SMB300	300		SHB08		8
SMB400	400		SHB10		10
SMB500	500		SHB12		12

4.2. Results

We applied, as comparison criterion, the GAP, in equation 13, which measures the deviation from the upper bound (UB) achieved by each method from the lower bound (LB) returned by the exact method B&C. As presented in Table 3, the subset SMB100 had optimal solutions for every instance, and near-optimal (with gaps $< 2.5\%$) for the remaining subsets. The deterministic approach outperforms the simple and multi-population genetic algorithms approaches in every instance set, primarily due to the size of the instances. Regarding computational time, the Multi-Population Genetic Algorithm outperforms the simple Genetic Algorithm and the B&C in nine out of ten instances. In contrast, the simple genetic algorithm surpasses the B&C in the same instances. The MPGA approach provides solutions in significantly less computational time compared to the deterministic approach, despite yielding slightly suboptimal results within a deviation of less than 5%. This efficiency stems from the MPGA leveraging approximation techniques that circumvent the exhaustive search required in deterministic methods. Unlike exact algorithms, which systematically explore all possible solutions to ensure global optimality, the structure of the MPGA we use employs a local search in the crossover process that expedites convergence toward near-optimal solutions.

$$GAP(\%) = 100 * \frac{Method^{UB} - B\&C^{LB}}{Method^{UB}} \quad (13)$$

To enhance our analysis, we applied the Kruskal-Wallis (KW) and Simes-Hochberg tests based on the GAP and Time values, as detailed in Table 4. The KW test is a non-parametric method that compares two or more populations. It evaluates the null hypothesis (H_0), which states that all samples have equal distribution functions, against the alternative hypothesis (H_1), which posits that at least two samples have different distribution functions. The hypotheses are presented in Equation (14).

The p-value is a statistical metric used to assess the validity of the null hypothesis, indicating whether there is a correlation between groups. The null hypothesis is rejected when the p-value is below the significance level, typically set at 0.05 (5%). In this case, since the p-value falls below this threshold, we reject the null hypothesis.

The non-parametric test is applied since the GAP values do not satisfy the Anderson-Darling normality test. After the KW test, the Simes-Hochberg test was used for multiple pairwise comparisons.

Table 3: Average GAP (%) and time for small instances.

	B&C		sGA		MPGA	
	GAP	Time (s)	GAP	Time (s)	GAP	Time (s)
SMB100	0,00%	92,36	1,86%	846,23	0,57%	151,48
SMB200	0,18%	1806,89	2,38%	1493,91	1,31%	379,40
SMB300	0,95%	1929,47	3,65%	1810,22	1,99%	653,29
SMB400	0,73%	1855,47	4,77%	1719,97	2,37%	527,96
SMB500	1,52%	3274,99	6,42%	1261,01	2,91%	400,84
SHB04	0,25%	1450,35	2,80%	1468,52	1,48%	738,53
SHB06	0,78%	1842,92	3,12%	835,41	2,16%	1330,77
SHB08	0,93%	2620,10	3,15%	681,85	1,72%	639,79
SHB10	1,76%	2303,48	3,70%	711,15	2,57%	347,09
SHB12	2,36%	2498,36	4,90%	777,44	2,63%	192,57

Table 4: Statistical analysis for gap and time

Instance Groups	B&C		GA		MPGA	
	KW	P-value	GAP	Time	GAP	Time
SMB100	23.10	0.000	B	BC	A	AB
SMB200	38.64	0.000	C	A	A	A
SMB300	29.90	0.000	C	A	A	AB
SMB400	37.44	0.000	E	A	A	A
SMB500	39.20	0.000	B	A	A	BC
SHB04	39.50	0.000	C	AB	A	AB
SHB06	44.88	0.000	C	A	A	AB
SHB08	27.37	0.000	C	A	A	B
SHB10	24.94	0.000	B	A	A	AB
SHB12	29.68	0.000	BC	A	A	BC

$$KW = \begin{cases} H_0 : B\&C = GA = MPGA \\ H_1 : B\&C, GA, MPGA; \text{ are not all equal} \end{cases} \quad (14)$$

The p-values from the Kruskal-Wallis (KW) test indicate a significant difference, as they are all below 0.05 (5%) in Table 4. The pairwise comparison reveals that the exact method B&C achieves the lowest rank values among the groups, demonstrating the lowest rank in 9 out of 10 subsets. The KW test again reveals a significant difference in computational efficiency to reach optimal solutions for each instance, with all p-values remaining under 0.05 (5%). However, the exact method does not yield the lowest rank values among the groups. In contrast, both genetic algorithms typically converge more rapidly to their optimal solutions, which aligns with the expected behavior of evolutionary algorithms, particularly when utilizing prior knowledge of the problem.

To sum up, B&C method consistently achieves the smallest or second smallest GAP values for most instances, with particularly notable performance for SMB100 (0.00%) and SHB04 (0.25%). However, this advantage in solution quality is offset by significantly higher computational times, often exceeding 1800 seconds and reaching up to 3274.99 seconds for SMB500. The sGA method, while generally slower than MPGA, exhibits larger GAP values, especially for larger instances such as SMB500 (6.42%) and SMB400 (4.77%). Nevertheless, it is substantially faster than

B&C in several instances, indicating its capacity to find reasonably good solutions in shorter times. MPGA demonstrates a balanced performance, achieving relatively low GAP values (e.g., 0.57% for SMB100 and 1.31% for SMB200) with significantly reduced computational times compared to B&C and sGA. This suggests that MPGA effectively trades a slight increase in GAP for a large improvement in computational efficiency. The results reveal a clear trade-off between solution quality and computational time. While B&C ensures the smallest GAP values, MPGA provides a more favorable balance between accuracy and efficiency, making it an attractive choice for scenarios where computational resources or time are limited.

5. Concluding Remarks

In this work, we proposed an MILP and applied two genetic algorithms (sGA and MPGA with an individual disposed in a grid structure) approaches to solve instances of GCIP-NF. The instances are generated based on data provided by a CGI, where increasing values of parameters such as the number of machines and time horizon are used when defining scenarios. The MILP was solved using the branch and cut algorithm available in the CPLEX solver, which provided optimal and near-optimal solutions in small instances, while the multi-population genetic algorithms find near-optimal solutions within a reasonable computational time. All methods can reach the best feasible solution, where B&C reaches such solutions spending less than 2500 sec for 55 out of 100 instances. The evolutionary techniques are a satisfactory approach; however, the exact method B&C seems to present the better trade-off between solution quality (feasible and optimal) and execution time. From a practical standpoint, the grid structure allows MPGA to reach high-quality solutions (low GAP values) in significantly less time than the exact B&C method and the standard sGA. While B&C achieves slightly better or similar GAP values, its exhaustive combinatorial search leads to prohibitive computational costs. In contrast, MPGA balances intensification (through focused search within promising grid cells) and diversification (through exploration across the grid), producing competitive solutions at a fraction of the time, as confirmed by the statistical results in Table 4. Future works include developing a MILP formulation for multiple furnaces and a stochastic model for demand forecast. Other proposed solutions combine meta-heuristics such as Variable Neighbourhood Search or Greedy Randomized Adaptive Search Procedure with mathematical programming techniques for the GCI-NF.

References

- Almada-Lobo, B., Klabjan, D., Carravilla, M. A., and Oliveira, J. F. (2010). Multiple machine continuous setup lot sizing with sequence dependent setups. *Computational Optimization and Applications*, 47(3):529–552.
- Arantes, M. S. (2014). Ambiente para desenvolvimento de métodos aplicados a problemas de otimização. Master's thesis, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, S.P. 116 pg.
- Auer, J., Bey, N., and Schäfer, J.-M. (2017). Combined life cycle assessment and life cycle costing in the eco-care-matrix: A case study on the performance of a modernized manufacturing system for glass containers. *Journal of Cleaner Production*, 141:99–109.
- Brahimi, N., Absi, N., Dauzère-Pérès, S., and Nordli, A. (2017). Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*, 263(3):838–863.

- Copil, K., Wörbelauer, M., Meyr, H., and Tempelmeier, H. (2017). Simultaneous lotsizing and scheduling problems: a classification and review of models. *OR Spectrum*, 39(1):1–64. ISSN 1436-6304.
- Eiben, A. E. and Smith, J. E. (2015). *Introduction to evolutionary computing*. springer.
- Toledo, C. F. M., Arantes, M., Hossomi, M. Y. B., and Almada-Lobo, B. (2016). Mathematical programming-based approaches for multi-facility glass container production planning. *Computers Operations Research*, 74:92–107.
- Toledo, C. F. M., Oliveira, R. R. R., and França, P. M. (2013). A hybrid multi-population genetic algorithm applied to solve the multi-level capacitated lot sizing problem with backlogging. *Computers Operations Research*, 40(4):910–919.
- Varsei, M. and Polyakovskiy, S. (2017). Sustainable supply chain network design: A case of the wine industry in australia. *Omega*, 66:236–247.
- Wörbelauer, M., Meyr, H., and Almada-Lobo, B. (2019). Simultaneous lotsizing and scheduling considering secondary resources: a general model, literature review and classification. *Or Spectrum*, 41:1–43.