Unequal individual genetic algorithm with intelligent diversification for the lot-scheduling problem in integrated mills using multiple-paper machines

Marcos Furlan[a,b], Bernardo Almada-Lobo[b], Maristela Santos[a], Reinaldo Morabito[c,*]

[a] *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Av. Trabalhador São-carlense 400, São Carlos 13560-970, Brazil*

[b] *INESC-TEC, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias s/n, Porto 4200-465, Portugal*

[c] *Departamento de Engenharia de Produção, Universidade Federal de São Carlos, Rod. Washington Luís KM 235, São Carlos 13565-905, Brazil*

Abstract

This paper addresses the lot-sizing and scheduling problem of pulp and paper mills involving multiple paper machines. The underlying multi-stage integrated production process considers the following critical units: continuous digester, intermediate stocks of pulp and liquor, multiple paper machines and a recovery line to treat by-products. This work presents a mixed integer programming (MIP) model to represent the problem, as well as a solution approach based on a customized genetic algorithm (GA) with an embedded residual linear programming model. Some GA tools are explored, including literature and new operators, a novel diversification process and other features. In particular, the diversification process uses a new allele frequency measure to change between diversification and intensification procedures. Computational results show the effectiveness of the method to solve relatively large instances of the single paper machine problem when compared to other single paper machine solution methods found in the literature. For multiple paper machine settings, in most runs the GA solutions are better than those obtained for the MIP model using an optimization software.

Keywords: Genetic algorithm; Diversification mechanism; Pulp and paper industry; Multi-stage lot-sizing and scheduling problem; Parallel machines.

1. Introduction

The pulp and paper industry is a high-capital productive sector of, mostly, large mills that can produce pulp, paper or both in an integrated fashion. This complex production system can be viewed as a multi-stage problem that combines some critical units in single or multiple production lines. These critical units may have limited resources which are expensive to enlarge in order to avoid production bottlenecks. For that reason, it is necessary to efficiently use those resources. Furthermore, there are different pulp and paper mill layouts and a wide range of items produced, ranging from standard paper to special pulps used to produce textiles, for example. Because of the complexities involved in these industrial settings, it is difficult to determine effective production plans for most companies in this sector, requiring problem description generalizations, elaborate mathematical models and novel solution approaches.

A pulp and paper mill basically consists of: the pulp digester, which produces virgin pulp using a thermochemical, chemical or mechanical process to extract the pulp from the wood; the paper machine, which produces paper of different characteristics to meet customer demands; the recovery plant, which recovers the production residual and produces steam, electric energy and some components used in pulp

*Corresponding author. Fone: +55 16 3351-9516

Email addresses: mafurlan@icmc.usp.br (Marcos Furlan), almada.lobo@fe.up.pt (Bernardo Almada-Lobo), mari@icmc.usp.br (Maristela Santos), morabito@ufscar.br (Reinaldo Morabito)

extraction in the case of chemical processes; and the intermediate stocks of pulp and liquors, which allows higher autonomy between consecutive production units in case of breaks and production exchanges.

The pulp and paper lot-sizing and scheduling problem aims at improving the use of the production resources in order to, for example, minimize variable production costs or maximize contribution margins and profits. Some production characteristics and restrictions should be considered and ensured in the developed production plans. The aim is to define the lot sizes and their production sequence during a pre-defined multi-period planning horizon, considering customer demands, resource limits and other characteristics such as production rates, setup costs and setup times in each period.

Various papers studying different planning problems associated with the pulp and paper industry can be found in the literature, such as using surplus steam for district healing [1, 2], lot-sizing problems associated with paper machines [3], production problems integrating lot-sizing and cutting decisions in the paper machines [4, 5] and integrated lot-sizing and scheduling problems [6, 7]. As the focus of the present study is on problems coupling lot-sizing and scheduling decisions, some related papers which have tackled these integrated problems are described in more detail below.

For instance, [6] studied the integrated problem in a Portuguese pulp and paper mill and proposed a MIP-based heuristic to solve it. The authors considered three sub-plants of the company: the pulp plant, the paper plant and the recovery plant. The solutions of the method provide detailed production plans for the entire mill. [7] studied the same integrated problem in [6], but considered two additional characteristics: production cycles in the production sequence of the paper machine and backlogging within these production cycles. A VNS approach was developed to solve the problem and its performance was compared to the previous MIP-based heuristic and an optimization solver. These papers addressed multi-stage mills with one resource per stage, that is, a single digester which continuously feeds a single paper machine.

This paper has two main purposes. The first is to extend previous mathematical formulations to cope with production processes which have multiple production lines, as found in other pulp and paper companies. This work focuses particularly on some Brazilian mills with multiple paper machines competing for raw material, such as virgin pulp. In these production processes, the digester feeds multiple paper machines and sends the by-product to a single recovery line. The pulp and by-product intermediate stock levels should be controlled during the production process. These mills pose additional production planning challenges which are not addressed in previous papers.

Synchronization between consecutive productive processes is necessary to enable intermediate inventory control. In [6], this control was performed in sub-period changes since the authors consider no variation in the production flow during each sub-period. Sub-periods are smaller parts of a period as, for example, production shifts or smaller periods of time. The model presented in Section 3 adopts the same assumption. However, a new challenge arises to solve that assumption due to the multiple paper machines. The quantity of sub-periods is increased to maintain the flexibility between lot sizes and machine schedules. This way, the number of sub-periods per day is at least equal to the number of shifts multiplied by the number of machines. The growth in the number of sub-periods associated with the minimum lot size requirements easily creates infeasibility situations, which means that it is difficult to ensure the minimum lot sizes as the quantity of sub-periods grows. To overcome this difficulty, the minimum lot-sizing constraints were rewritten, even allowing null sub-periods when the produced paper grammage does not change. In order to avoid speed changes in the digester during tiny or null sub-periods, a new set of constraints was included to define the minimum sub-period size and allow speed variation.

The second main purpose of this paper is to present a novel solution approach based on genetic algorithm (GA) to solve the pulp and paper lot-sizing and scheduling problem with multiple paper machines. GA became popular after the pioneering work by [8]. Several variants of the original GA have been proposed in different areas of the literature over the last decades. Basically, a GA consists of an initial population (set of solutions) and selection, crossover and mutation operators. A comprehensive survey of GA was presented by [9], where important characteristics of GA were discussed. [10] reviewed different papers applying GA to solve lot-sizing problems. The authors discussed GA and problem characteristics, and classified the literature papers according to these aspects. Some examples of recent studies applying GA to solve lot-sizing problems are found in [11–18].

The proposed GA contains some special structures to solve this problem. For example, the decode and genetic operators are based on rules of this specific problem, which can be adapted and extended to similar cases. Furthermore, as part of this work a diversification mechanism was developed to introduce some variability throughout the search. Sometimes the convergence process is premature in GA, depending on the selection pressure, type of individual replacement, shape of the solution space, among other factors. This issue is overcome by the alternative diversification procedure. Another feature presented in GA is the process of adjusting the number of sub-periods of each individual. Adjusting this number is a difficult task and quite often that quantity is overestimated. The idea here is to adjust this value through the generations, allowing the use of individuals with a different number of sub-periods in the same population. Furthermore, this work also investigates the benefits of using larger initial populations and some hot-start solutions (solutions constructed based on specific heuristics or rules).

The method presented here is a hybrid approach, which uses an LP solver to provide the best linear programming solution for each integrated scheduling pattern (individual). Methods combining meta-heuristics and exact methods in combinatorial optimization have already been used in the literature, yielding good results [12]. Recently, a genetic algorithm with mathematical programming techniques embedded in was successfully applied to a two-level lot sizing and scheduling problem arising from the soft drink industry [18]. These and other papers have demonstrated successfully the advantages of combining metaheuristics with mathematical programming to solve real-world problems. According to [19], this method fits into an integrative combination by incorporating an exact method (LP model) in a meta-heuristic (GA).

This paper is organized as follows: the pulp and paper lot-sizing and scheduling problem characteristics are described in detail in Section 2 and a mathematical model for the multiple paper machine case is presented in Section 3. The proposed GA is described in Section 4 and the problem data and computational results are presented in three parts in Section 5. Firstly, the best GA variant is chosen by combining each feature developed/used in this paper. Secondly, the best variant is compared to single machine approaches of the literature for solving single machine problem instances, and then to the MIP solver Cplex for solving multiple machine problem instances. Thirdly, additional computational tests are presented and analyzed considering different runtime limits for a subset of multiple paper machines instances. Finally, in Section 6 conclusions are drawn and some topics for future research are suggested.

2. Problem definition

As illustrated in Figure 1, this paper considers the pulp and paper production process. This representation is based on the problem considered by [6], where the mill is divided into three different plants. The pulp plant consists of the digester, the associated recycled pulp mill and the tanks of pulp. The paper plant is composed of the parallel paper machines, the winders and the reels (intermediate stocks of big paper rolls called jumbos). The recovery plant includes the evaporator, the tanks of black liquor, the recovery boiler and the energy turbines to transform surplus steam into electric energy. The function of each unit is briefly presented below.

- Pulp plant:
  - The digester produces the virgin pulp and the black weak liquor as a by-product (see Figure 1). Two types of thermochemical digesters can be observed in the mills: continuous production and batch production. Continuous digesters are bigger and their production is controlled by changing the digester speed, the amount of chemicals introduced and the temperature. This study addresses a continuous digester and some industrial practices, as in [6].
  - The recycled pulp mill produces the recycled pulp which is used to produce different types (or families) of papers. The percentage of recycled pulp added can greatly vary between the types of papers. The papers produced in certain Brazilian mills do not contain recycled pulp in their compositions and there are no recycled mills in these plants.
  - The tanks of pulp store the virgin and recycled pulps. These stocks are buffers between the digester and paper machines and provide some flexibility to their operations along the planning

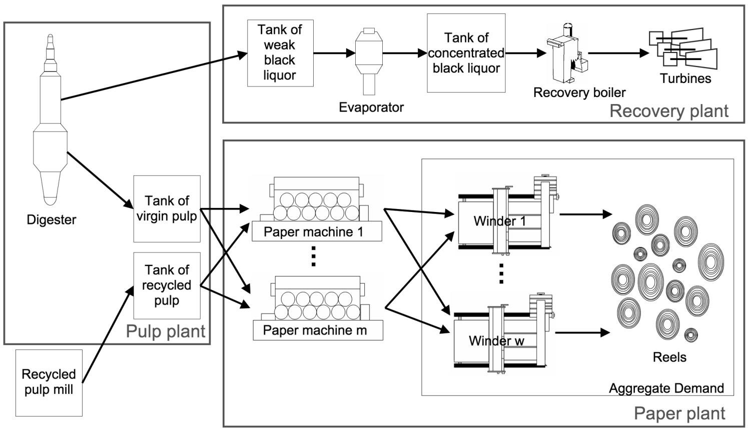


Figure 1: Illustration of an integrated pulp and paper mill with multiple paper machines.

horizon. There are maximum and minimum required inventory levels. The first relates to the tank capacities and the second is the minimum amount required to maintain a certain outflow of pulp.

- Paper plant:
  - The paper machines blend pulp and other components to produce different types of papers that meet customer demands. The components include (besides pulp): inks, water, corn starch, bleach and glue. This moisture is deposited in a canvas and then it is dried by pressure processes and steam. Each paper machine produces only one type of paper at a time and the changes between paper grammages generate production losses in terms of machine time and raw material. These losses are sequence dependent and are bigger between products of different paper families, if compared to products of the same family.
  - The winders and reels roll the paper produced in the paper machines in jumbos, which are then cut into smaller reels and/or rectangular pieces of paper according to customer demand. This study considers customer demand aggregated in tons for each paper grammage, as in [6].
- Recovery plant:
  - The evaporator removes water from the weak black liquor. The process commonly uses temperature and/or pressure. The evaporator has a limited production capacity.
  - The tanks of black liquor store the weak and concentrated black liquors. These stocks are buffers between the digester, the evaporator and the recovery boiler. Similarly to the tanks of pulp, they also have minimum and maximum required inventory levels.
  - The recovery boiler burns the concentrated black liquor to produce steam, which pushes the turbines, and inorganic residue. This inorganic residue goes to a causticizing process in order to be reused in the digester. The steam is used in internal processes, such as wood cooking in the digester and paper drying in the paper machines. The turbines transform surplus steam into electric energy.

3. Mathematical formulation

The MIP model for the pulp and paper lot-sizing and scheduling problem with multiple paper machines is presented according to the plant division in Figure 1. The model considers the following assumptions:

1. The digester speed requires a minimum time after changes to stabilize its production.

2. The intermediate item flows are constant during a sub-period and may change only between sub-periods making it possible to maintain the stock level feasibility during the continuous time only by controlling the sub-period changes;
3. Each paper machine can produce any grammage, that is, any type of paper.
4. A reasonable number of sub-periods is necessary to ensure flexibility between paper machine schedules and lot sizes. A minimum value is assumed to be equal to the number of paper machines multiplied by the number of shifts.
5. The product demands (in reels and/or ordered items) are aggregated by grammage.
6. To anticipate the downstream average waste of the cutting process, the aggregate demand is increased by an average trim loss.

Three major changes are made to the single paper machine lot-sizing and scheduling model of [6] in order to extend it to deal with the multiple paper machine case: the minimum lot-size constraints are changed according to [20]; minimum sub-period size constraints are included to avoid speed changes in fictitious sub-periods (sub-periods too small); and the paper machine changeover constraints are exchanged by tighter constraints, as referred by [21] and performed by [7] in the same context. Furthermore, the constraints have been eliminated that synchronize the changes in the digester and the changes in the paper machine. The direct reuse of lost raw material has also been excluded as recycled pulp, because these losses must be dealt with before they are reused.

The following sections present short explanations of the model decisions and constraints involved in each plant that composes the mill presented in Figure 1. Tables 1, 2 and 3 show the model parameters and variables used to formulate each plant. The complete extended MIP model is presented in detail in Section 3.5.

3.1. Pulp plant

The digester produces the raw material used in the paper machines, as well as the by-product treated in the recovery line. The planning is divided into T periods ($t = 1 \dots T$) of fixed length (days) and each period is divided into $|S_t|$ sub-period ($s = 1 \dots S$) of variable length. Its outputs in a given sub-period s are related to the size of this time bucket, the digester work speed and the conversion factor α or ρ (defined in Table 1). These speeds are discrete due to a smooth shift control observed in practice [6], and also to maintain the linearity, since the digester speed and the bucket size are model variables. The discrete steps (minimum allowed changes) are defined by parameter Φ (in RPM).For example, using a digester that has a minimum speed of 10 RPM and $\Phi = 0.25$, the set of possible speeds is $\{10, 10.25, 10.5, ...\}$. A maximum variation (Δ) in the speed between consecutive sub-periods is defined in order to maintain the digester stability. A minimum sub-period size is also required to allow for velocity changes. Parameter n_{min} in Table 1 defines the minimum sub-period size to allow speed changes and it is used in constraints (9). Table 1 summarizes the data and the variables used to model the pulp plant. The constraints (2) - (11) are related to the pulp plant.

3.2. Paper plant

The main decision to be made in each paper machine is the grammage which will be produced at each sub-period s. The percentage of the moisture, the production rate and the setup times and loss are defined according to the grammage produced. The model assumes that each paper machine works without interruption, and therefore idle times are not allowed. Trim loss estimations are considered in the aggregated demand, since the cutting of reels is not considered in the model. Data and problem variables that control the paper plant operations are presented in Table 2 and constraints (12) - (24).

3.3. Recovery plant

The production plan in this plant aims at defining the amount produced in each unit. Four main characteristics are considered here: the recovery boiler production limits of steam and burning quantity, the evaporator processing limit and the intermediate stock levels. Table 3 summarizes the data and problem variables associated with the recovery plant. Constraints (25) - (33) represent this plant.

Table 1: Data and decision variables associated with the pulp plant.

Data	
T	Set of periods in the planning horizon (expressed in days);
S	Set of sub-periods in the planning horizon;
M	Set of paper machines;
S_t	Set of sub-periods belonging to period t;
V	Set of discrete digester speeds;
Φ	Speed variation step in the discrete set (in rotation per minute - RPM);
Δ	Maximum speed variation (expressed in number of discrete levels);
$v^{dig}_{max}(v^{dig}_{min})$	Maximum (minimum) speed of the digester (in RPM);
sp_v	Speed (in RPM) of the digester at each discrete value v;
cap_t	Capacity of period t (in hours);
n_{min}	Minimum size of a micro-period to allow for a change at the digester speed (length in hours);
α	Production rate of pulp in the digester (tons per RPM);
ρ	Production rate of weak black liquor in the digester (tons per RPM);
$Y^{dig}_{v,0}$	Initial speed definition; ($Y^{dig}_{v,0} = 1$, if the digester starts the planning horizon with speed v; $Y^{dig}_{v,0} = 0$, otherwise).
$xr^{recy}_{max}(xr^{recy}_{min})$	Maximum (minimum) production rate of the recycled plant (in tons per hour);
$I^{virg}_0(I^{recy}_0)$	Initial Inventory of virgin pulp (recycled pulp) (in tons);
$I^{virg}_{max}(I^{virg}_{min})$	Maximum (minimum) stock level of virgin pulp (in tons);
$I^{recy}_{max}(I^{recy}_{min})$	Maximum (minimum) stock level of recycled pulp (in tons);

Decision variables	
$Y^{dig}_{vs} \in \{0,1\}$	Speed of the digester in sub-period s; ($Y^{dig}_{vs} = 1$, if the digester runs at speed v in sub-period s; $Y^{dig}_{vs} = 0$, otherwise);
$Nh_{vs} \geq 0$	Amount of time that the digester's speed is set to v in sub-period s;
$N_s \geq 0$	Length of sub-period s (expressed in hours);
$X^{dig}_s \geq 0$	Amount of pulp produced in sub-period s;
$X^{liquor}_s \geq 0$	Amount of weak black liquor produced in sub-period s.
$X^{recy}_s \geq 0$	Amount of recycled pulp prepared in sub-period s.
$I^{virg}_s \geq 0$ ($I^{recy}_s \geq 0$)	Inventory of virgin pulp (recycled pulp) at the end of sub-period s (in tons);
$O^{virg}_{sm} \geq 0$ ($O^{recy}_{sm} \geq 0$)	Output of virgin pulp (recycled pulp) in sub-period s to supply the paper machine m (in tons);

3.4. Objective function

There are some difficulties associated with the accurate measurement of costs and/or profits in production planning. Inventory costs, for example, can be estimated by the price of the items stored or by the value associated with the inventory space. The profit driven by the production is also difficult to define accurately. The values of raw materials, energy consumption, and other cost parameters of the process need to be known. For example, it is difficult to calculate the profit of the electric energy production, since part of the steam and electric energy are consumed in the production process and its amount is not well known.

3.5. Mathematical model

The complete extended mathematical model is presented below.

$$min \sum_{j\in K}\sum_{t\in T} h^+_{jt} \cdot IG^+_{jt} + \sum_{j\in K}\sum_{t\in T} h^-_{jt} \cdot IG^-_{jt} + \sum_{j\in K}\sum_{k\in K}\sum_{s\in S}\sum_{m\in M} sc_{kjm} \cdot Z_{kjsm} - \sum_{s\in S} \gamma \cdot O^{steam}_s \qquad (1)$$

Table 2: Data and decision variables associated with the paper plant.

Data	
K	Set of grammages;
sl_{kjm}	Paper loss (in tons) in a changeover from grammage k to grammage j in machine m;
st_{kjm}	Time loss (in hours) in a changeover from grammage k to grammage j in machine m;
$b^{virg}_{jm}(b^{recy}_{jm})$	Percentage of virgin pulp (recycled pulp) used to produce grammage j in machine m;
b_{jm}	Processing time of grammage j in machine m (hours/ton);
G_{jm}	Large number to each grammage j and machine m (in tons);
m_{jm}	Minimum lot-size of grammage j in machine m (in tons);
D_{jt}	Demand for grammage j in period t (in tons);
Y_{j0m}	Initial configuration of machine m: ($Y_{j0m} = 1$, if machine m is prepared to produce grammage j at the beginning of the planning horizon; $Y_{j0m} = 0$, otherwise).

Decision variables	
$Z_{kjsm} \in \{0,1\}$	Changeovers in each machine: ($Z_{kjs} = 1$, if a changeover from grammage k to j occurs in sub-period s and machine m; $Z_{kjs} = 0$, otherwise);
$Y_{jsm} \in \{0,1\}$	Setup configuration of each machine: ($Y_{js} = 1$, if machine m is set up to grammage j in sub-period s; $Y_{js} = 0$, otherwise);
$X_{jsm} \geq 0$	Amount of grammage j produced in sub-period s and machine m (in tons);
$IG^{+}_{jt} \geq 0$	Inventory of grammage j in period t (in tons);
$IG^{-}_{jt} \geq 0$	Backlogged amount of grammage j in period t (in tons).

Table 3: Data and decision variables associated with the recovery plant.

Data	
I^{liquor}_0	Initial inventory of weak black liquor;
$I^{liquor}_{max}(I^{liquor}_{min})$	Maximum (minimum) weak black liquor stock level (in m^3);
C^{evap}	Maximum capacity of the evaporator (in m^3 per hour);
β	Evaporation factor to transform weak black liquor in concentrated black liquor;
$I^{c.liq}_0$	Initial inventory of concentrated black weak liquor (in m^3);
$I^{c.liq}_{max}(I^{c.liq}_{min})$	Maximum (minimum) stock levels of concentrated black liquor (in m^3);
C^{steam}_{burn}	Maximum capacity of recovery boiler to produce steam (in m^3 per hour);
σ	Conversion factor from concentrated black liquor to steam;
$C^{r.boiler}_{burn}$	Maximum burning capacity of recovery boiler (in tons per hour);

Decision variables	
$O^{liquor}_s \geq 0$	Amount of weak black liquor evaporated in sub-period s:
$I^{liquor}_s \geq 0$	Inventory of weak black liquor in sub-period s:
$X^{c.liq}_s \geq 0$	Quantity of concentrated black liquor produced in sub-period s:
$I^{c.liq}_s \geq 0$	Inventory of concentrated black liquor in sub-period s;
$O^{c.liq}_s \geq 0$	Output of concentrated black liquor in sub-period s;
$O^{steam}_s \geq 0$	Amount of steam produced in sub-period s.

$$s.t.: \sum_{v=1}^{V} Y^{dig}_{vs} = 1, \; s \in S, \tag{2}$$

$$Y_{vs}^{dig} \leq \sum_{k=(v-\Delta\cdot\Phi)}^{v+\Delta\cdot\Phi} Y_{k,s-1}^{dig},\ v \in V,\ s \in S, \tag{3}$$

$$Nh_{vs} \leq cap_t \cdot Y_{vs}^{dig},\ v \in V,\ t \in T,\ s \in S_t, \tag{4}$$

$$N_s = \sum_{v \in V} Nh_{vs},\ s \in S, \tag{5}$$

$$\sum_{s \in S_t} N_s = cap_t,\ t \in T, \tag{6}$$

$$X_s^{dig} = \alpha \cdot \sum_{v \in V} sp_v \cdot Nh_{vs},\ s \in S, \tag{7}$$

$$X_s^{liquor} = \rho \cdot \sum_{v \in V} sp_v \cdot Nh_{vs},\ s \in S, \tag{8}$$

$$N_s \geq n_{min} \cdot (Y_{vs}^{dig} - Y_{v,s-1}^{dig}),\ v \in V,\ s \in S, \tag{9}$$

$$X_s^{recy} \geq xr_{min}^{recy} \cdot N_s,\ s \in S, \tag{10}$$

$$X_s^{recy} \leq xr_{max}^{recy} \cdot N_s,\ s \in S, \tag{11}$$

$$X_s^{dig} + I_{s-1}^{virg} = \sum_{m \in M} O_{sm}^{virg} + I_s^{virg},\ s \in S, \tag{12}$$

$$X_s^{recy} + I_{s-1}^{recy} = \sum_{m \in M} O_{sm}^{recy} + I_s^{recy},\ s \in S, \tag{13}$$

$$I_{min}^{virg} \leq I_s^{virg} \leq I_{max}^{virg},\ s \in S, \tag{14}$$

$$I_{min}^{recy} \leq I_s^{recy} \leq I_{max}^{recy},\ s \in S, \tag{15}$$

$$\sum_{j \in K} b_{jm}^{virg} \cdot (X_{jsm} + \sum_{k \in K} sl_{kjm} \cdot Z_{kjsm}) = O_{sm}^{virg},\ s \in S,\ m \in M, \tag{16}$$

$$\sum_{j \in K} b_{jm}^{recy} \cdot (X_{jsm} + \sum_{k \in K} sl_{kjm} \cdot Z_{kjsm}) = O_{sm}^{recy},\ s \in S,\ m \in M, \tag{17}$$

$$\sum_{j \in K} (b_{jm} \cdot X_{jsm} + \sum_{k \in K} st_{kjm} \cdot Z_{kjsm}) = N_s,\ s \in S,\ m \in M, \tag{18}$$

$$\sum_{s \in S_t} \sum_{m \in M} X_{jsm} + IG_{j,t-1}^{+} - IG_{j,t-1}^{-} = D_{jt} + IG_{jt}^{+} - IG_{jt}^{-},\ j \in K,\ t \in T, \tag{19}$$

$$X_{jsm} \leq G_{jm} \cdot Y_{jsm},\ j \in K,\ s \in S,\ m \in M, \tag{20}$$

$$m_{jm} \cdot (Y_{jsm} - Y_{j,s-1,m}) \leq X_{jsm},\ j \in K,\ s \in S,\ m \in M, \tag{21}$$

$$\sum_{j \in K} Y_{jsm} \leq 1,\ s \in S,\ m \in M, \tag{22}$$

$$\sum_{j \in K} Z_{kjsm} = Y_{k,s-1,m},\ k \in K,\ s \in S,\ m \in M, \tag{23}$$

$$\sum_{k \in K} Z_{kjsm} = Y_{jsm},\ j \in K,\ s \in S,\ m \in M, \tag{24}$$

$$X_s^{liquor} + I_{s-1}^{liquor} = O_s^{liquor} + I_s^{liquor},\ s \in S, \tag{25}$$

$$I_{min}^{liquor} \leq I_s^{liquor} \leq I_{max}^{liquor},\ s \in S, \tag{26}$$

$$0 \leq O_s^{liquor} \leq C_{evap} \cdot N_s,\ s \in S, \tag{27}$$

$$X_s^{c.liq} = \beta \cdot O_s^{liquor},\ s \in S, \tag{28}$$

$$X_s^{c.liq} + I_{s-1}^{c.liq} = O_s^{c.liq} + I_s^{c.liq},\ s \in S, \tag{29}$$

$$I_{min}^{c.liq} \leq I_s^{c.liq} \leq I_{max}^{c.liq}, \; s \in S, \tag{30}$$

$$O_s^{c.liq} \leq C_{burn}^{r.boiler} \cdot N_s, \; s \in S, \tag{31}$$

$$O_s^{steam} = \sigma \cdot O_s^{c.liq}, \; s \in S, \tag{32}$$

$$O_s^{steam} \leq C_{steam}^{r.boiler} \cdot N_s, \; s \in S. \tag{33}$$

The objective function (1) consists of minimizing production costs while maximizing the production of steam, in which h_{jt}^{+} refers to the inventory cost of one ton of grammage j at period t, h_{jt}^{-} is the backlogging cost of one ton of grammage j at period t and sc_{kjm} is the cost to change from grammage k to grammage j in machine m. γ represents an incentive to produce electric energy.

Constraints (2) define the speed of the digester in each sub-period. Constraints (3) set the smooth variation in the digester speeds. To maintain the stability of the digester processing, it is important that the throughput is not changed too quickly. Therefore, the absolute speed variation is limited to $\Delta \cdot \Phi$. The sub-period size is defined by constraints (4) and (5). Note that only one speed is chosen per micro-period. Considering this information, the production of virgin pulp and weak black liquor can easily be determined by means of (7) and (8). Idle time is not allowed in production planning and this feature is imposed by constraints (6). Constraints (9) allow changes in the digester speed if and only if the sub-period is of a minimum required length (n_{min}). Constraints (10) and (11) define minimum and maximum production of the recycled pulp plant at each sub-period s, respectively. Constraints (12) and (13) ensure the inventory balance of virgin pulp and recycled pulp, respectively. The minimum and maximum levels of the tank stocks are ensured by (14) and (15).

Constraints (16) and (17) evaluate the output of virgin and recycled pulp in each machine m at each sub-period s. Constraints (18) ensure that production and setup times are equal to the sub-period size, which means that paper machines are not allowed idle times. Furthermore, these constraints synchronize the paper machines with the other resources. The grammage inventory balance is given by (19). Constraints (20) ensure that a paper machine only makes a grammage if it is properly set up and constraints (21) enforce a minimum lot size to each grammage j when there is a changeover. In other words, the minimum lot-sizes are required only for the first sub-period of a production campaign. The term "production campaign" means a continuous sequence of sub-periods where the same grammage is produced. According to (22) a maximum of one grammage can be scheduled per machine at each point in time. Finally, the flow of the changeovers are caught by constraints (23) and (24). As referred by [21], this type of constraints establishes a tighter correspondence between setup and changeover variables than those originally used by [20]. The tighter constraints are not present in [6]; however, they are also used by [7]. It is mainly associated with methods which solve linear relaxed sub-problems, for example, MIP-based heuristics and a branch-and-cut method.

The inventory balance is given by (25) and the stock bounds are validated by constraints (26). The evaporator capacity is met in (27). Constraints (28) define the amount of concentrated black liquor produced in each sub-period s. Equations (29) provide the inventory balance of concentrated black liquor. The respective inventory bounds are defined by (30). Constraints (31) and (33) set the burning capacity of the recovery boiler and the upper bound on the steam produced per hour, which is determined in (32).

4. Genetic Algorithm

This section describes a genetic algorithm based method to solve the pulp and paper lot-sizing and scheduling problem with multiple paper machines. It is a hybrid solution method composed of an exact method embedded in the GA. The representation of the solution simultaneously addressed the two scheduling problems (paper machine grammage and digester speed schedules) and an LP solver is used to obtain the linear optimal solution for the residual problem, according to each scheduling pattern evaluated (individual). Furthermore, novel operators to introduce diversity in the solution pool and a diversity measurement based on alleles frequency are presented. Improved initial solutions and population size variation are also shown and further investigated.

The GA proposed here uses the 2-point crossover and six different mutation operators in which five are of general purpose and one is based on the knowledge of the problem. A novel diversification process is also proposed, based on diversity thresholds and a diversity measurement. The process of adjusting the number of sub-periods is performed by allowing individuals of different sizes (with regards to the number of sub-periods) in the same population. The use of a larger initial population and some hot-start solutions are also investigated. The pseudo-code is presented in Algorithm 1.

Algorithm 1: The hybrid genetic algorithm pseudo-code.

```
Initial population(pop) (Sect. 4.2);
Greedy Sequencing Heuristic(pop) (Sect. 4.3);
Parameters Adjustment(mutprob,φ,divFlag) (Sect. 4.5.1);
Evaluation(pop,φ) (Sect. 4.6);
Selection(pop) (Sect. 4.9);
while time limit is not reached do
    Crossover(pop) (Sect. 4.7);
    Parameters Adjustment(mutprob,φ,divFlag);
    Mutation(pop.offspring, mutprob) (Sect. 4.8);
    Evaluation(pop,φ);
    Selection(pop);
    Diversification(pop,divFlag) (Sect. 4.5);
end
Return the best solution;
```

4.1. Individual

The underlying problem includes one pure scheduling problem associated with the digester and M lot sizing and scheduling problems associated with the M paper machines. The first problem defines the digester speeds over the planning horizon. The other M problems determines the grammage production sequence in each paper machine. In both cases, the idea is to make decisions on the speed used/grammage produced at each sub-period of the digester/paper machine.

In Figure 2 each line or vector is called chromosome, each gray position (light and dark gray) is a gene and each possible value is called allele. In the Digester (Variation chromosome), the possible alleles refer to the range of speed variations. The alleles from PM1 and PM2 chromosomes vary within the grammage index values, in which line PM1 shows the production sequence for paper machine 1 and PM2 presents the sequence to paper machine 2.

Periods		0	1				2				3			
Sub-periods		0	1	2	3	4	5	6	7	8	9	10	11	12
Digester	Variation	*	2	3	4	2	1	0	2	3	1	1	3	4
	Speeds	12.0	12.0	12.5	13.5	*	13.0	12.0	12.0	*	11.5	11.0	11.5	*
PM 1 (grammages)		*	1	2	5	2	4	3	0	2	3	4	5	2
PM 2 (grammages)		*	1	1	3	3	5	4	3	1	3	2	6	4

Figure 2: Individual representation with the digester speed variation and the paper machine scheduling.

The sub-problem resulting from the fixed scheduling is solved by an LP solver. To ensure that all individuals respect the smooth shift on the speed of the digester (constraint (3)), the related vector defines the variation of the speeds between consecutive sub-periods instead of the absolute speed values (Variation in Figure 2). This variation is limited by parameter Δ. For example, $\Phi = 0.5$ RPM and $\Delta = 2$ means that the speed can vary at most by 1 RPM ($\Phi \cdot \Delta$) between consecutive sub-periods (Figure 2). To avoid negative values in the chromosome, the range of values is relocated. In Figure 2, a value zero in the chromosome means a speed reduction of 1 RPM, a value 1 means a speed reduction of 0.5 RPM, a value 2 means that the speed is kept unchanged, a value 3 means a speed increase of 0.5 RPM, and, finally a value 4 means a speed increment of 1 RPM. Furthermore, when a calculated speed exceeds the maximum or the minimum velocity allowed, it is replaced by the respective limit. Smooth changeovers are also desirable in the paper machine; however, positive setup costs and times reinforce this smoothing.

Other characteristics that can be observed in Figure 2 are the variation in the number of sub-periods. The light gray cells define the active genes that affect the solution scheduling, dark gray cells define the passive genes that are overlooked in the decoding process, and the white cells are implicit values (sub-periods or speeds). The key idea is that disabled genes could transmit passive information that can be useful in future generations. This information maintains some diversity by holding alleles to next generations, without penalizing the solution in terms of changeover costs. The control of the number of active genes at each period is defined by a "dominant gene" that ranges from a minimum number of sub-periods to the original number of sub-periods. In the example of Figure 2, this gene value is 3.

4.2. Initial population

The initial population is commonly generated randomly in the literature. Here, a random grammage is chosen to be produced for each individual and each sub-period, regardless of the other sub-periods. The vector of speed variation is also randomly generated within the allowed range of speed variation. This way of generating the initial population produces a good variability and guarantees a desirable initial diversity. This paper uses part or all of the solutions produced in this way.

The initial population could also be generated by other procedures, such as a simple problem-specific heuristic or more standard heuristics. The population can be partially or totally constructed by those "improved" individuals. It depends on the diversity of the solutions, the computational complexity and degree of improvement of the last generation solutions comparatively to the previous ones. These are called hot-start solutions as they can provide a better start to the GA. Section 4.2.1 presents a general heuristic to construct initial solutions based on the linear relaxation of the entire mathematical model.

4.2.1. Non-random initial solutions (hot-start solutions)

Hot-start solutions have been used in different research to improve the GA by accelerating the initial convergence process. This type of solution can be produced by using constructive heuristics (such as MIP based-heuristics and rounding-based methods) and improvements can be applied before the solution is inserted in the population. By starting with better solutions, the method is supposed to have more time to find even better solutions. Examples of papers that rely on hot-starts are [22] and [23].

The hot-solutions are constructed based on the linear relaxation of the model (1)-(33). The main idea here is to first solve the complete linear relaxation of the problem. The values of the setup decisions (variables Y_{jsm}) of the linear solution are then used to define the probabilities of each grammage being chosen for each micro-period. As an example, in case $Y^*_{121} = 0.3$ the probability of choosing grammage 1 at sub-period 2 in machine 1 is equal to 30%. In the following step, a roulette wheel method is performed to define each allele according to the calculated probabilities. To add more diversity, 10% of the probabilities are equally divided by all alleles. After freezing the setup-related binary variables, the digester's speed scheduling is defined randomly. The digester's binary variables are not determined by fixing the linear relaxation results because they do not necessarily respect the smooth constraints. Finally, the remaining linear variable values are obtained by solving the LP sub-problem.

The feasibility of these solutions is not guaranteed nor mandatory, as infeasible individuals are addressed (see Section 4.6.1).

4.3. Greedy sequencing heuristic

This heuristic rearranges the production sequences between sub-periods in the same period, reducing the changeover costs without increasing inventory and backlog costs. The method starts from the beginning of the planning horizon onwards by removing all the alleles for all paper machines and then re-inserting them sequentially ordered by increasing changeover costs, according to the last grammage allocated. This process is performed iteratively for each period in a forward procedure. Consider 2 paper machines, ready to produce 4 different grammages per period and initially set to produce grammage 1. Consider also an individual with chromosomes [1,4,2] and [2,3,3] for machines 1 and 2, respectively. For the sake of simplicity, consider changeover costs proportionally to the numerical difference between the grammage indices, i.e., changeover from grammages 1 to 2 costs 1, from grammages 1 to 3 costs 2 and so on. The procedure starts by removing all the alleles and creating the auxiliary set of values {1,4,2,2,3,3}. At each iteration, the allele with the smaller changeover cost (considering the last grammage used) is removed from this set and it is moved to the gene which is being defined. For example, in the first iteration, grammage 1 is removed from the auxiliary set and is allocated to the beginning of the production sequence of machine 1 (the changeover cost is zero, since machine 1 is initially set to produce grammage 1). In the second iteration, the process chooses one grammage 2 from the reduced set {4,2,2,3,3}. This procedure continues until completing paper machine 1, followed by the schedule of the other paper machines in the sequence. In the example provided, this procedure returns the production sequences [1,2,2] and [3,3,4], and the total changeover is reduced from 6 to 4 in this example.

4.4. Frequency spread measure

The frequency spread measure (FSM) is a metric based on the frequency of each allele in each gene of the entire population. The alleles of a gene correspond to all possible values this gene can have. The allele frequency of gene s in an individual i is defined as: $\frac{(freq(s,g(s,i))-1)}{Popsize-1}$, in which $g(s,i)$ defines the allele scheduled in the individual i in gene s, $freq(s,g(s,i))$ is the amount of times that this allele appears at this position (gene) s and $Popsize$ is the population size.

Based on the frequency of each chromosome in an individual, is possible to define its "Individual Diversity Incapacity" (IDI) coefficient as the average of the frequency value of the genes in this individual. The IDI measures the percentage of alleles that an individual shares with the population, ranging from 0 to 1, where 0 means an individual composed only of alleles different from the entire population and 1 means that the population has the same alleles in all genes, which means that all individuals are equal. A population with individuals with high IDI is weakly spread (bigger value), thus confirming the need to start the diversification process.

The FSM is calculated as the average of all individuals' IDI and is used to establish when the diversification process should be initiated and concluded. This paper determines two different frequency tables and IDI, one for the digester problem (associated with the Y_{vs}^{dig} variables) called IDI^{dig} and the other for the set of paper machine scheduling (associated with the Y_{jsm} variables) called IDI^{pm}. The final IDI of an individual is the maximum between IDI^{dig} and IDI^{pm}, that is, $IDI = \max(IDI^{dig}, IDI^{pm})$.

For example, consider individuals with genes defined in Table 4. In the first gene, allele A1 appears four times, while the two other alleles (A2,A3) appear once. The respective frequency table is shown in Table 5.

Table 4: Population of 6 individuals, 3 genes and 3 different alleles.

Individuals	Genes		
	G1	G2	G3
I1	A1	A2	A3
I2	A1	A1	A3
I3	A2	A3	A1
I4	A3	A1	A2
I5	A1	A3	A3
I6	A1	A2	A2

Table 5: Frequency table of population from Table 4.

Allele	Genes		
	G1	G2	G3
A1	4	2	1
A2	1	2	2
A3	1	2	3

Individual I1 has the sequence of alleles [A1, A2, A3] with frequencies [4, 2, 3], respectively (Table 5). The IDI of I1 is $average\{\frac{4-1}{6-1}; \frac{2-1}{6-1}; \frac{3-1}{6-1}\} = 0.4$. Moreover, I3 contains alleles [A2, A3, A1] with frequencies [1, 2, 1], and its IDI is $average\{\frac{1-1}{6-1}; \frac{2-1}{6-1}; \frac{1-1}{6-1}\} \approx 0.07$. Therefore, I3 has more potential to add diversity $(IDI(I1) > IDI(I3))$.

The frequency spread measure (FSM) of the population, defined by the average IDI, is equal to:

$$FSM = \frac{0.4(I1) + 0.4(I2) + 0.07(I3) + 0.13(I4) + 0.4(I5) + 0.33(I6)}{6} \approx 0.29$$

4.5. Diversification process

The premature convergence is an important drawback of population based methods. This issue may occur due to an excessive selection pressure, elitism or even problem characteristics. [24] and [25] propose different diversification approaches.

As part of this work, a novel diversification mechanism was designed based on the alleles frequency and two thresholds ($\overline{dt}$ and $\underline{dt}$). The $\overline{dt}$ denotes the upper bound on the alleles frequency variation in the population accepted until the diversity process starts. Moreover, $\underline{dt}$ determines the stoppage of the diversity process, restarting the normal search procedure (intensification). The divFlag indicates whether the diversification is active (true) or not (false). Algorithm 2 presents the pseudo-code of this diversification.

Algorithm 2: Diversification(pop,divFlag)

averDist=FSM(pop) (Sect. 4.4);
if *averDist*$> \overline{dt}$ *and divFlag=false* **then**
    `/* Starting the diversify process */`
    divFlag=true;
end
if *averDist*$< \underline{dt}$ *and divFlag=true* **then**
    `/* Ending the diversify process */`
    divFlag=false;
end

4.5.1. Parameter adjustment

The mutation rate and the φ (diversification intensity) factor are increased during the diversification process. This increases the relevance of diversity by improving the acceptance of different solutions in the population. Parameter φ was defined as the trade-off between diversity and quality of the solution. Then the following equations were built to estimate the φ value according to how much the decision maker is willing to lose in the objective function in order to gain diversity:

$$(1 + \varepsilon)^{\varphi} = 1 + \psi, \tag{34}$$

which can be easily transformed into

$$\varphi = \frac{ln(1 + \psi)}{ln(1 + \varepsilon)}, \tag{35}$$

where ψ is the percentage of loss ($\psi \geq 0$) and ε is the diversity(IDI) difference between the solutions. Exponential functions were adopted to penalize similar individuals because the penalties increase faster in very similar solutions as opposed to others. Furthermore, it is possible to turn off the diversification process just by setting φ to zero. For example, if a 1% of loss in the solution's quality is accepted to maintain a solution that adds 10% more diversity, φ is calculated as:

$$\varphi = \frac{ln(1+0.01)}{ln(1+0.1)} \approx 0.104,$$

When the process stops, the φ parameter is gradually reduced to ensure that diversity fades smoothly.

The mutation probabilities were calculated to be proportional to the number of genes using the following expression:

$$mut^p = \left(1 - p^{\frac{1}{GS}}\right), \tag{36}$$

where GS denotes the chromosome size ($|S|$) and p refers to the probability that an individual has of not mutating. Equation (36) is used to define mut^{Max} and mut^{Min}, given predefined Min and Max mutation probabilities.

Algorithm 3 shows the complete process of adjusting parameters. During the diversification process, the mutation is increased by steps and φ is changed to its upper bound. The opposite occurs after the diversification process stops, that is, the mutprob is changed to its lower bound and φ is reduced by steps. $divIt$ is an input parameter used to define the number of steps. φ^{Min} and φ^{Max} can be calculated by equation (35) and used during the intensification and diversification search processes, respectively.

Algorithm 3: ParametersAdjustment(mutprob,φ,divFlag)

```
/* mutprob is the current mutation probability, initially set to mut^Min */
if divFlag then
    mutprob=min(mut^Max,mutprob+⌈(mut^Max − mut^Min)/divIt⌉);
    φ = φ^Max;
else
    mutprob=mut^Min;
    φ = max(φ^Min, φ − (φ^Max − φ^Min)/divIt);
end
```

4.5.2. Alternative diversification process

This section presents an alternative to the process presented in Section 4.5.1. The idea is to verify, with computational tests, the need to separate the diversification and intensification processes. Here, the entire diversification procedure is performed by a dynamic parameter adjustment. This means that, the mutation probability and the parameter φ are adjusted in each generation according to the population distances (population IDI). Algorithm 4 shows the pseudo-code of the alternative diversification process.

Algorithm 4: AlternativeParametersAdjust(pop,mutprob,φ)

$$mutprob = mut^{Min} + \left\lceil min\left(1, max\left(0, \frac{FSM(pop) - \underline{dt}}{\overline{dt} - \underline{dt}}\right)\right)\right\rceil \cdot \left(mut^{Max} - mut^{Min}\right);$$

$$\varphi = \varphi^{Min} + min\left(1, max\left(0, \frac{FSM(pop) - \underline{dt}}{\overline{dt} - \underline{dt}}\right)\right) \cdot \left(\varphi^{Max} - \varphi^{Min}\right);$$

4.6. Decoding and fitness evaluation

The decoding of each individual is related to both paper machines and digester scheduling problems. The paper machine schedules are directly defined by the integer values of the chromosome (grammage indices).

For example, in Figure 2 paper machine 1 (PM1) is assigned to produce the grammage sequence $\{1, 2, 5\}$ in the first period. The representation of the digester speed is indirect. The chromosome defines the variation of the speed throughout the planning horizon. The speeds for each sub-period are defined according to the previous velocity and the respective limits, in a forward procedure.

The resulting sub-problem is a linear programming problem defined from the mathematical model (1)-(33) with the integer variables fixed. Some constraints can be removed from the mathematical model, namely constraints (2), (3), (22), (23) and (24), given that their conditions are guaranteed by the individual codification. The remaining model is subsequently modified fixing the following variables: Y_{vs}^{dig},Y_{jsm} and Z_{kjsm}. The linear sub-problem is solved by an LP solver.

The fitness function, presented in (37), contains three components:

1. the evaluation of the objective function for the individual p, called $f(p)$ and defined by (1), standardized by the average objective function $(\overline{f})$ of the entire population;
2. the relative unfitness given by the individual infeasibility $u(p)$ that is normalized comparatively to the individual with the highest infeasibility from the population;
3. the diversity measure is calculated using φ and the IDI of the individual.

$$fitness(p) = \frac{f(p)}{\overline{f}} \cdot \left(1 + \frac{u(p)}{\max_{q\in Pop} u(q)}\right) \cdot (1 + IDI(p))^{\varphi} \tag{37}$$

4.6.1. Infeasibility

The main objective of keeping infeasible solutions is to maintain population variability. According to [26], infeasible solutions can be treated in different ways, for example: rejecting infeasible solutions, repairing infeasible solutions or introducing a penalty function (used here).

The summation of infeasibility is calculated by relaxing the constraints that control the inventory of virgin pulp (12), the inventory of recycled pulp (13), the inventory of weak black liquor (25) and the minimum lot-size requirements (21). The relaxed model is presented below.

$$\begin{aligned} min\ u(p) = {} & \vartheta \cdot \sum_{s\in S}(I_s^{virg(+)} + I_s^{virg(-)}) + \iota \cdot \sum_{s\in S}(I_s^{recy(+)} + I_s^{recy(-)}) \\ & + \delta \cdot \sum_{s\in S}(I_s^{liquor(+)} + I_s^{liquor(-)}) + \eta \sum_{s\in S} X_{jsm}^{slack} \qquad (38) \\ s.to: {} & f(p) = (1), \\ & (2) - (11), \\ & (14) - (33), \\ & X_s^{dig} + I_{s-1}^{virg} + I_s^{virg(+)} = \sum_{m\in M} O_{sm}^{virg} + I_s^{virg} + I_s^{virg(-)},\ s \in S, \qquad (39) \\ & X_s^{recy} + I_{s-1}^{recy} + I_s^{recy(+)} = \sum_{m\in M} O_{sm}^{recy} + I_s^{recy} + I_s^{recy(-)},\ s \in S, \qquad (40) \\ & X_s^{liquor} + I_{s-1}^{liquor} + I_s^{liquor(+)} = O_s^{liquor} + I_s^{liquor} + I_s^{liquor(-)},\ s \in S, \qquad (41) \\ & m_{jm} \cdot (Y_{jsm} - Y_{j,s-1,m}) \leq X_{jsm} + X_{jsm}^{slack},\ j \in K,\ s \in S,\ m \in M. \qquad (42) \end{aligned}$$

where variables $I_s^{(+)}$ and $I_s^{(-)}$ increase and decrease the inventory levels I_s, respectively, to maintain them within their respective bounds. X_{jsm}^{slack} represents the amount of production still required to reach minimum lot-size when a changeover takes place. Each time an individual is deemed infeasible, the relaxed model (38) - (42) is solved to meet the infeasibility level.

4.7. Crossover

The crossover operator is responsible for mixing the genes of individuals, producing offspring with expected better individuals. Various papers discuss crossover operator design and the building blocks theory [9, 27, 28]. In this paper, the 2-point operator was chosen to perform the solution's crossover in the GA and this options is supported by [29]. The authors show that 2-point operators treat the chromosome as a ring and is less disruptive than uniform and other n-point crossover operators.

Regarding the solutions with a different number of sub-periods, the same combination procedure of two parents was considered with the original number of sub-periods (maximum number of sub-periods). This means that, the number of genes is constant, and only the amount of active genes vary. For example, matching parents with 3 and 4 active genes generate one child with 3 active genes and another one with 4 active genes.

The individual consists of two scheduling problems and contains $M+1$ chromosomes to encode a solution, where M is the amount of paper machines. The crossover points are chosen independently for each vector, which means that the scheduling crossover has the same source (parents) but the combination points are different for each other.

4.8. Mutation

Seven different mutation operators are proposed. The first five are of general purpose, one is related to the variation in the number of sub-periods and the remaining one incorporates knowledge of the problem, inducing the growth of the continuous production campaign. All mutations have different probabilities of being realized and they are used sequentially, from more to less disruptive. This means that all changes can be done in the same individual, depending on the application frequency of each operator and the mutation rate. Table 6 shows the application frequency and the disruption level of each mutation used. All mutation operators are presented below.

1. Simple mutation
   An allele different form the current value is randomly selected when a gene is also randomly selected for mutation. For example, consider that the gene selected for mutation has allele 3 and this gene's alleles range from 1 to 8. The simple mutation can generate a value in the set: $[1, 2] \bigcup [4, 8]$.
2. Swap mutation
   The swap mutation exchanges the alleles of two genes which have been randomly selected for mutation. After the first gene was selected for mutation, the second is chosen in the group of genes with equal characteristics. This means that, genes with a paper machine chromosome do not swap alleles with genes with a digester variation chromosome. When the two alleles are equal, a simple mutation is applied to the first selected gene.
3. Insert mutation
   Inserting the mutation disturbs the solution more and is applied less often: the mutation is applied only once per chromosome, which means that each individual mutates at most $M + 1$ times. If a chromosome is selected for mutation, two positions are chosen and a value v is randomly generated. The value v is inserted into the first position and the subsequent genes are postponed by one sub-period until the second position is reached. The last value, that is, the value from the second position in the initial solution, is removed. Figure 3 illustrates an example of insert mutation where the first position is sub-period 3, the second position is sub-period 7 and the inserted value is 0. The following values are pushed until sub-period 7.
4. Ejection chain mutation
   The ejection chain mutation rotates a vector of values R times to the right, in which R is a random value between zero and the chromosome size. This process also drastically changes the individual, and therefore, it is applied less often than other mutations. Figure 4 illustrates an example with $R = 1$, $R = 2$ and $R = 3$.
5. Sequence mutation
   This mutation operator modifies the sequence of a complete period. The new sequence is produced

Periods		1			2			3		
Sub-periods		1	2	3	4	5	6	7	8	9
Before	Digester	2	3	4	1	0	2	1	1	3
	PM1	1	2	5	5	3	0	3	4	5
After	Digester	2	3	0	4	1	0	2	1	3
	PM1	1	2	5	5	3	0	3	4	5

Figure 3: Insert mutation example.

Periods		1			2			3		
Sub-periods		1	2	3	4	5	6	7	8	9
Before	Digester	2	3	4	1	0	2	1	1	3
	PM1	1	2	5	5	3	0	3	4	5
R=1	Digester	2	3	4	1	0	2	1	1	3
	PM1	5	1	2	5	5	3	0	3	4
R=2	Digester	2	3	4	1	0	2	1	1	3
	PM1	4	5	1	2	5	5	3	0	3
R=3	Digester	2	3	4	1	0	2	1	1	3
	PM1	3	4	5	1	2	5	5	3	0

Figure 4: An example of ejection chain mutation.

randomly and is kept even when the solution is worse. Consider that a period chosen for mutation has the production sequence $\{5, 3, 2\}$. The sequence mutation can produce any permutation between 2,3 and 5, for example, $\{3, 2, 5\}$ or $\{2, 3, 5\}$. This mutation is performed every time a period is chosen for mutation.

6. Sub-period mutation
The sub-period mutation increases the number of active genes, thus changing the number of sub-periods considered by an individual at each period. Every time an individual is chosen for mutation, its size increases because it is expected that the selection operator put an emphasis mostly on individuals with less sub-periods (smaller setup costs). However, the size of the individual decreases when the number of sub-periods is equal to its upper bound. For example, if the individual shown in Figure 2 is selected for mutation, its value increases from 3 to 4, which means that 4 alleles should be considered per period.

7. Campaign mutation
The campaign mutation considers the production sequence to change the value of a gene. This paper considers that a production campaign is a set of consecutive sub-periods in which the same item is produced without changes. This mutation was developed to improve the production rates by reducing the setup loss. When a gene is selected for mutation, six conditions need to be met as shown below:

   - Case 1: If the chosen gene s is the first gene of a chromosome ($s = 1$), their allele receives the same value found in the second gene, that is, $x[1] \leftarrow x[2]$.
   - Case 2: If the chosen gene s corresponds to the last gene of a chromosome ($s = |S|$), the following operation, $x[s] \leftarrow x[s-1]$, is executed.
   - Case 3: If the chosen gene s is at the end of a production campaign, which means that this gene's allele is equal to the previous ($x[s] = x[s-1]$) and is different from the subsequent allele ($x[s] \neq x[s+1]$), then the gene is changed to start the next production campaign ($x[s] \leftarrow x[s+1]$).
   - Case 4: If the chosen gene s is placed at the beginning of the production campaign ($x[s] = x[s+1]$ and $x[s] \neq x[s-1]$), its allele is changed to incorporate it in the previous production campaign ($x[s] \leftarrow x[s-1]$).
   - Case 5: If the chosen gene s breaks a production campaign, that is, if the previous and subsequent alleles are equal to each other ($x[s-1] = x[s+1]$) and the allele of gene s is different from both ($x[s] \neq x[s+1]$), then its allele is changed to join the campaign ($x[s] \leftarrow x[s-1]$).

- Case 6: If the allele of the chosen gene s is different from the previous ($x[s] \neq x[s-1]$) and the subsequent ($x[s] \neq x[s+1]$) values and they are different from each other ($x[s-1] \neq x[s+1]$), it will have a 50% probability of taking the value of the previous ($x[s] \leftarrow x[s-1]$) and a 50% probability of taking the subsequent allele ($x[s] \leftarrow x[s+1]$).

Mutations summary

Table 6 presents a summary for the mutations used, considering three characteristics: the frequency with which a mutation is applied, the disruption caused to the mutated individual and whether it uses knowledge of the problem. The choice was to balance the application frequency with the disruption, that this, the more disruptive mutations are applied less frequently than others which are less disruptive. The last mutation incorporates some knowledge of the problem; however, the sub-period mutation could be seen also as having knowledge, since it changes the scheduling size.

Table 6: Mutations summary.

Name	Application frequency	Disruption caused	Problem knowledge
Simple	High	Low	No
Swap	High	Low	No
Insert	Low	High	No
Ejection chain	Low	High	No
Sequence	Medium	Medium	No
Sub-periods	Low	High	Yes
Campaign	High	Low	Yes

4.9. Selection

The crossover and survival selections are performed by 2-tournament operators. The k-tournament consists of randomly choosing k individuals, comparing and obtaining the best solution from the k [30].

The survival and crossover selection varies according to the number of the individuals selected, that is, the amount of tournaments done. In the survival selection, this process is performed until the pre-defined number of survivors is reached. In the crossover selection, two parents are selected by this approach to produce two new solutions. The offspring is produced to fulfil the population that was reduced in the survival selection.

There are two relevant differences between both selection processes. The survival selection copies the best individual without tournament and the number of times that an individual can be chosen varies. One individual can produce children more than one time in the crossover selection. However, in the survival selection, when an individual is selected to survive, it is removed from the set of possible choices so that there are no multiple copies of the same individual.

5. Computational experiments

5.1. Data

The problem instances were generated using a modified version of the generator proposed by [7]. These instances were built based on the data obtained from a Portuguese mill in [6], as well as on data and information obtained from Brazilian mills. Variations were included in the demand profiles and initial stocks of intermediate goods. Demand profiles of the generated instances ranged between 65% and 85% of the total resource capacities within a uniform distribution. The initial stocks also vary according to a uniform distribution, ranging from 80% to 120% of the original values.

The price of each grammage was calculated as the average sales price per ton. Instances with one and two paper machines were generated, as these are the most typical situations in practice. For the case with two paper machines, the demand was appropriately increased to maintain the total resource load. Furthermore, characteristic of the production cycles considered by [7] was removed since the problem addressed here does not have this characteristic. The paper prices were used to calculate the objective function costs. The setup (sc_{jkm}), inventory (h^+_{jt}) and backlog (h^-_{jt}) costs were calculated as: $sc_{kjm} = 0.25 \cdot sl_{kjm} \cdot price_j$, $h^+_{jt} = 0.25 \cdot \frac{price_j}{365}$ and $h^-_{jt} = 10 \cdot h^+_{jt}$, where sl_{kjm} is the setup paper loss (in tons) and $price_j$ is the average price per ton of grammage j.

Table 7 shows the sizes of the generated instances. Note that in instances with two paper machines, the number of sub-periods doubled comparatively to the ones with one paper machine. This was done to ensure problem feasibility. The recovery plant and digester capacities were increased to maintain the proportion of the resources. This was performed by multiplying the original values by the number of paper machines. For the sake of simplicity, two identical paper machines were considered and it was assumed that each machine can produce all paper grammages.

Table 7: Instance characteristics.

# of machines $\mid M \mid$	# of periods $\mid T \mid$	# of grammages $\mid K \mid$	# of sub-periods $\mid S \mid$	# of instances
1	7	{8,16}	{3,4}	10
1	15	{8,16}	{3,4}	10
1	30	{16}	{3,4}	10
2	7	{8,16}	{6,8}	10
2	15	{8,16}	{6,8}	10
2	30	{16}	{6,8}	10

5.2. Results

The solution approaches were built in C++ using ILOG OPL libraries and the LP sub-problems of Section 4 were solved using the Cplex 12.2 solver. The MIP problem in Section 3 was also solved using the Cplex in order to compare the results to the proposed GA. The performance of the VNS in [7] was also verified for the single paper machine problem instances. All computational experiments were performed on a PC with a 2.8GHz Core i5 2300 processor and 4 GB of RAM memory. The Cplex runtime limit was set to 3600 seconds. The GA runtime limit was set according to the Cplex time required. In cases that Cplex solved the problem in less than 600 seconds (small instances), the time limit used in the GA was set to 600 seconds; otherwise, the time limit was set to the same time required by Cplex. This procedure was adopted because the GA performed just a few generations in less than 600 seconds. Under these same processing capability and computational times, the comparisons were expected to be reasonably fair. The same time limit used for our GA was set as the time limit for the VNS.

Different GA parameters have to be adjusted, such as the population size or selection pressure. Performing this adjustment is a difficult task, since different sets of parameters present better results in different sets of tests. To equalize this variation, there was an attempt to choose more reliable parameter values after performing some preliminary tests. The following values were used here: population size = 100, selection pressure = 10%, minimum mutation probability = $1 - 0.8^{\frac{1}{|S|}}$, maximum mutation probability = $1 - 0.5^{\frac{1}{|S|}}$, crossover probability = 99% and $divIt$ = 50 generations.

The results are presented in three parts: in the first, the results of some variants of the GA are compared, taking into consideration diversification approaches, different bounds for the variation in the individual's size, hot-start solutions and bigger populations in the first generation. The combination of these features were compared to the use of performance charts (see [31]). In these charts, two or more methods can be compared on a finite set of instances. The gaps between the solution of each method and the best solution of all methods are calculated for each instance. A chart is then printed based on the amount of solutions with gaps smaller than τ. This is performed for every method and the amount of solutions is expressed in a percentage of the entire set of test cases. For example, in Figure 5(a), the variant NSF contains 60% of the solutions (for the problem instances) with gaps smaller than or equal to 10% of the best solution obtained and more than 70% of the solutions with gaps smaller than or equal to 40%. Furthermore, the NSR variant contains more than 85% and almost 95% of the solutions better than or equal to these same gaps, respectively. In the second part of the experiments, the best variant of the GA is compared to the VNS approach in [7] on single machine problem instances and to the Cplex 12.2 solutions on both single and multiple machine problem instances. In the third part, tests under different runtime limits, namely 1800 seconds (half hour) and 10800 seconds (three hours), were performed on a subset of multiple paper machines problem instances and the results were statistically analyzed.

These results were also compared in order to verify whether the differences in the solution quality of the methods have statistical significance within a certain degree of confidence. These analyses were made based on the relative gaps between the solution values and the best lower bound obtained by the CPLEX within a time limit. Instances were grouped by the amount of paper machines. For the meta-heuristics GA and VNS, an average value of five runs was used as a reference value for each instance. Confidence levels of 95 % were used and only the instances solved by both methods were considered, that is, paired analyzes were performed. The p-values were defined as follows: $p\text{-}value_{X,Y}^{Z,U,T}$, where X defines the compared methods, Y indicates the applied statistical method, Z and U refer to the set and the amount of problem instances, and T defines the time limit in seconds. The possible values for Z are: W for the Wilcoxon test, A for the Anderson-Darling test and T for the t-test. For example, $p\text{-}value_{GA\text{-}CPLEX,W}^{M1,80,3600}$ means that this p-value contains the comparison between GA and CPLEX solutions using the Wilcoxon test with 80 single paper machine instances and time limit of 3600 seconds.

Firstly, the Anderson-Darling test was applied to verify if the distributions of the relative gaps could be assumed as normal distributions. Then, pairwise tests were performed to compare the methods' gaps using either the t-test or the Wilcoxon test, according to the Anderson-Darling test results.

5.2.1. GA variants

Firstly, the lower bound of the individual's size (number of sub-periods) was adjusted and the best diversification process was chosen. Three options of each feature were tested amounting to 9 combinations. Taking the best variant in consideration, the same process was followed for the amount of hot-solutions and the initial population size with four possibilities for each characteristic in a total of 16 combinations. The tests were conducted by solving a subset of 80 instances, from different test sets, with 8 and 16 days. These instances were solved three times each, totaling 240 runs per variant. Table 8 shows a summary of the options and the acronym used to present each variant.

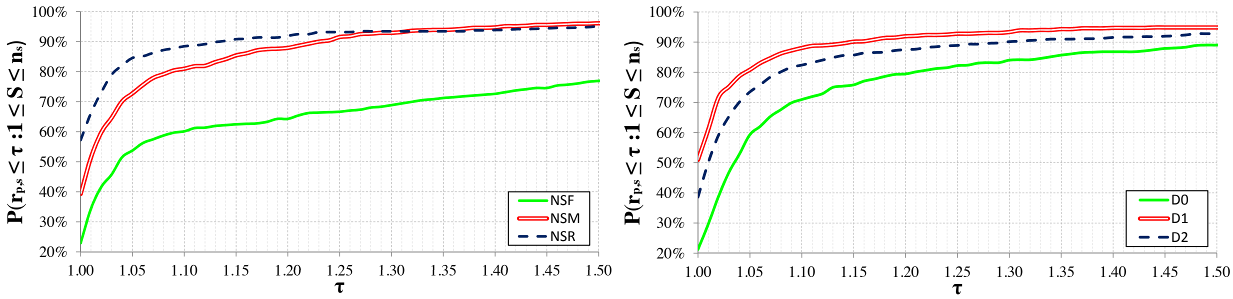
Figure 5(a) shows that the NSR and NSM variants outperform NSF, with a small advantage to the first. These results indicate that flexibility in the number of sub-periods leads to improvements in the quality of the final solution. Furthermore, the acceptance of individuals with different sizes brings/maintains some diversity in the solution pool. Figure 5(b) shows that the main diversification process (D1) is more effective, probably because it introduces more diversity. D2 is also better than the GA without a diversification process (D0), which demonstrates the need for a diversification process to avoid premature convergence.

Figure 6 shows the performance of all combinations of these two first features (number of sub-periods and diversification). The D1NSR combination provides the best results, and for that reason it was selected for the second sub-part of the tests. Moreover, it was possible to verify that the simplest GA, without diversification and variation in the number of sub-periods (D0NSF), produces the worst results, showing that both the diversification process and individuals with different sizes improve the GA performance. These results support the assumption that the simultaneous use of both techniques introduces more benefits than each technique used separately.

Besides the performance charts (Figure 6), the statistical relevance of some proposed operators was verified, such as: the diversification procedure, the adjustment of the number of sub-periods and the campaign mutation. None of the variants was accepted by the Anderson-Darling test with p-values lower than $9.025 \cdot 10^{-5}$. For this reason, the Wilcoxon test was applied to compare the best variant of Figure 6 (D1NSR) with the same configuration, removing only one operator at each time. Variants were named as follows: in

Table 8: GA variant acronym summary.

Acronym: Description	Acronym: Description
NSF : Static number of sub-periods	D0 : No diversification process
NSM : Number of sub-periods ranging between the number of shifts and the original number of sub-periods	D1 : Main diversification process (Sec. 4.5)
NSR : Number of sub-periods ranging between 1 and the original number of sub-periods	D2 : Alternative diversification process (Sec. 4.5.2)
H0 : Only random solutions	P1 : Initial population with normal size
H10 : 10 hot-start solutions	P2 : Initial population size doubled
H50 : 50 hot-start solutions	P5 : Initial population size multiplied by 5
H100 : 100 hot-start solutions	P10 : Initial population size multiplied by 10



(a) Instances aggregated by number of sub-periods.

(b) Instances aggregated by diversification process.

Figure 5: Performance charts aggregated for GA variants - part one.

D0NSR the diversification is turned off, in D1NSF the adjustment of the number of sub-periods is not allowed and D1WCM defines the variant without mutations per campaign.

All tests were performed with the alternative hypothesis indicating that D1NSR is better than the other variants (smaller gaps). When compared to D0NSR and D1NSF, the null hypotheses were rejected ($p\text{-}value^{-,79,3600}_{D1NSR\text{-}D0NSR,W} = 2.584 \cdot 10^{-10}$ and $p\text{-}value^{-,75,3600}_{D1NSR\text{-}D1NSF,W} = 5.729 \cdot 10^{-5}$), indicating that both operators had influences in the final results. In the case of campaign mutation, differences were not detected to demonstrate the operator's influence ($p\text{-}value^{-,80,3600}_{D1NSR\text{-}D1WCM,W} = 0.2547$). Probably, it happens due to the action of other tools of the method that attenuated this mutation influence.

Figure 7 shows one run of the GA presented here with each type of diversification for the same instance (2 paper machines, 15 periods, 8 grammages and 8 sub-periods). A fixed number of sub-periods (NSF) is considered for all runs in order to avoid being influences of sub-periods amount variation. The GA without diversification process (D0) (Fig. 7(a)) converges quickly. Moreover, D2 (Fig. 7(c)) presents a worse average value and converges more slowly. In Figure 7(b), D1 presents the best results by mixing intensification and diversification processes. These processes are clearly activated by analysing the average curve in Figure 7(b). Significant increments in the average curve indicate that the diversification process has started and significant reductions mean that the intensification process has begun. Note that the diversification process

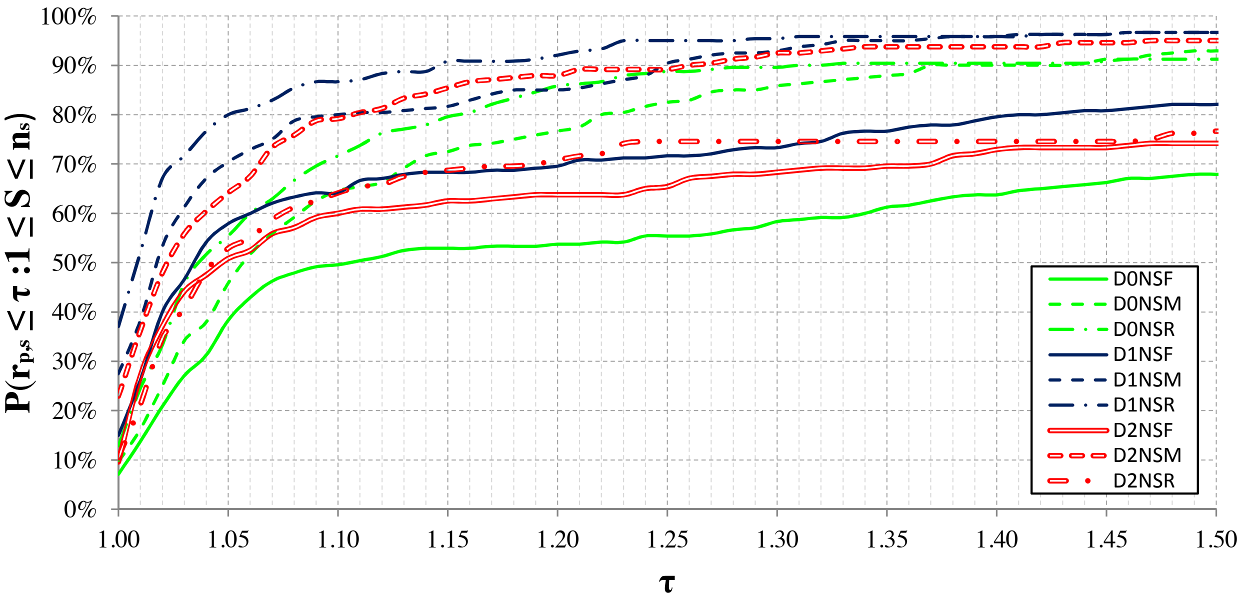
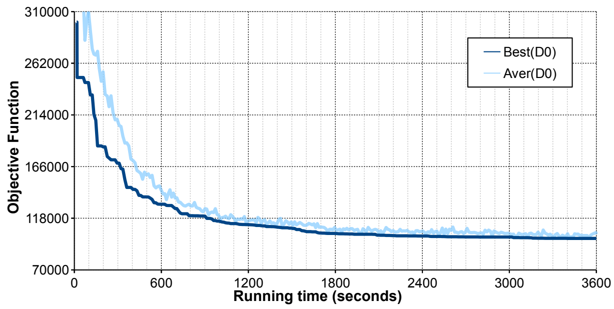
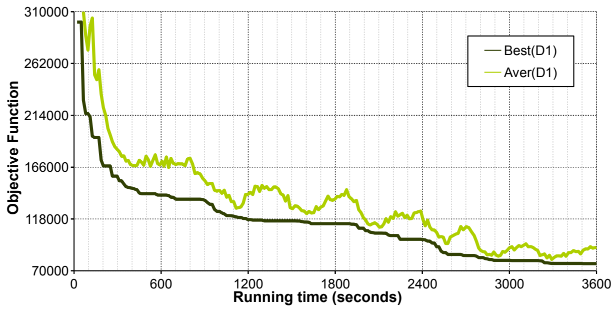


Figure 6: Performance charts for all GA variants - part one.

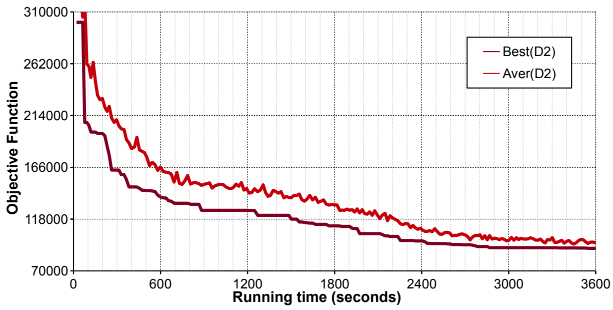
and the use of individuals of different sizes are chosen in the first part, and the variant with the best results is the D1NSR.



(a) GA behaviour without diversification.



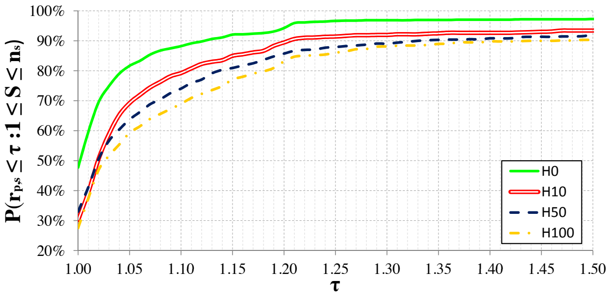
(b) GA behaviour with diversification 1.



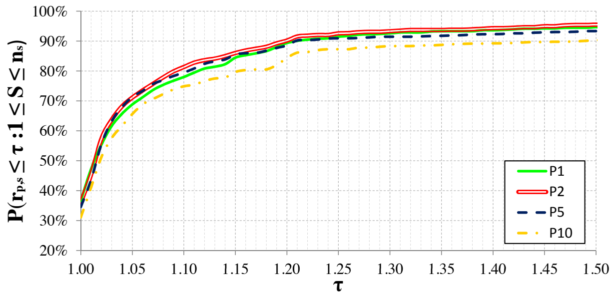
(c) GA behaviour with diversification 2.

Figure 7: GA behaviours with different types of diversification.

Four different numbers of hot-solutions and four different sizes of initial populations are used, in a total of 16 GA variants. Figure 8(a) shows the performance curves aggregated by the number of hot-start solutions and Figure 8(b) shows the overall performance by the initial population size. An unexpected result can be observed in the first chart. The performance increases as less improved solutions are introduced in the initial population. This behaviour is better analysed with the help of Figures 9 and 10. These runs show that sometimes the constructive heuristic does not deliver solutions that are better than the best solution created randomly, for example, the variant with zero hot-solutions (curve Best(H0)) and the variant with 100 hot-solutions (curve Best(H100)) in Figure 9. In this example, the GA with H0 starts with a Best!!!eh Best mesmo ou deveria ser better - verificar!!! value below H100. Another unexpected result occurs in Figure 10, where the same methods have changed their position concerning the quality of the solution. The curve Best(H100) starts with better results than Best(H0). However, the time spent to construct these 100 solutions is sufficient for variant H0, which substantially improves its solution. Furthermore, the use of these hot-solutions makes the convergence process slower in several cases. This can be observed by the distance between *Best* and *Aver* curves in both Figures 9(c) and 10(c), for example.

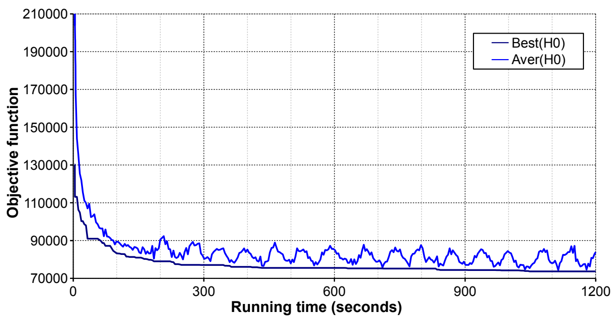


(a) Instances aggregated by number of hot-solutions.

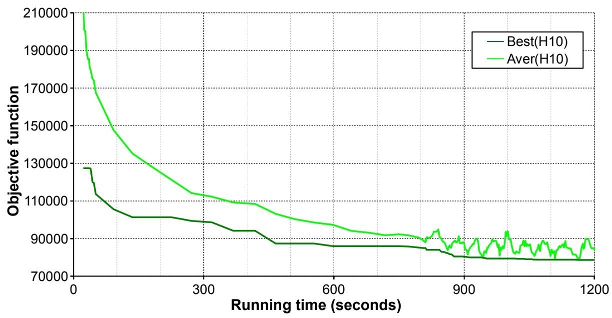


(b) Instances aggregated by initial population size.

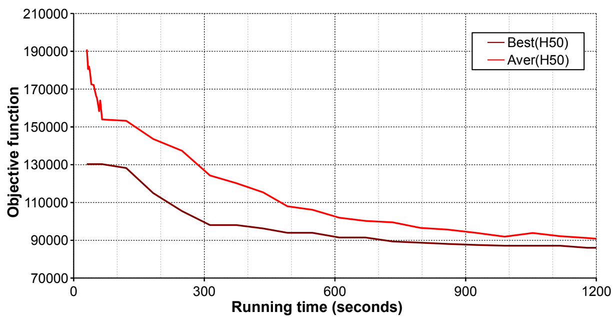
Figure 8: Performance charts aggregated for GA variants - part two.



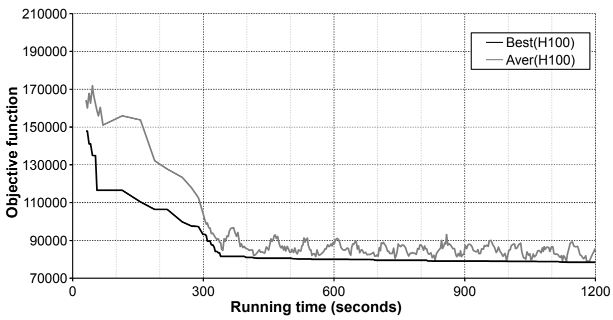
(a) GA behaviour without hot-solutions.



(b) GA behaviour with 10 hot-solutions.

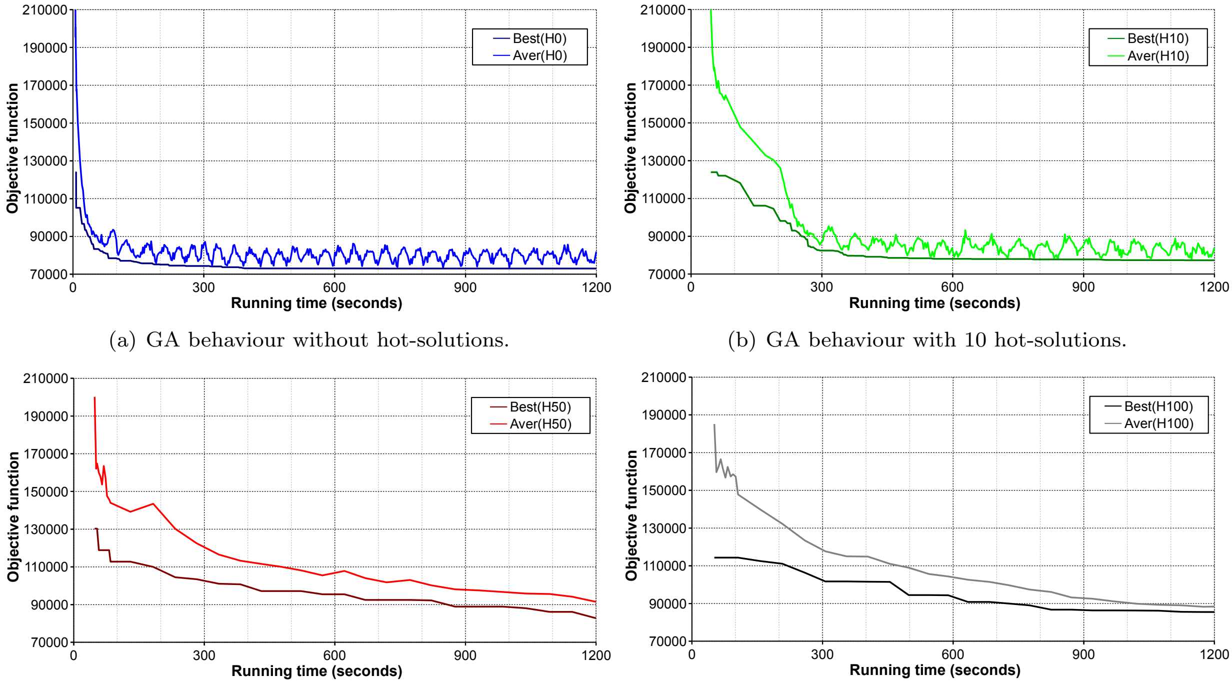


(c) GA behaviour with 50 hot-solutions.



(d) GA behaviour with 100 hot-solutions.

Figure 9: GA behaviour with normal initial population size for a different number of hot-solutions.



(a) GA behaviour without hot-solutions.

(b) GA behaviour with 10 hot-solutions.

(c) GA behaviour with 50 hot-solutions.

(d) GA behaviour with 100 hot-solutions.

Figure 10: GA behaviour with doubled initial population size for a different number of hot-solutions.

When compared to the other approaches, the final variant chosen involves: a number of sub-periods (NSR) to be released, a main diversification procedure (D1), zero hot-start solutions (H0) and initial population with size doubled (P2) comparatively to the other generations (NSRD1H0P2), hereafter called GA in the following sections.

5.2.2. Comparison with other approaches

This section compares the results of the GA with the VNS method proposed by [7] for a much related problem. It is important to observe that the VNS in [7] was adapted to the problem characteristics considered here. More precisely, two important points considered in the original VNS are overlooked here: the use of production cycles and the backlog cover. For further information about these characteristics, the readers are referred to [7].

According to these adaptations, the comparisons below are not totally fair; however, they provide a good idea about the performance of the method presented here. Furthermore, the results are also compared to the model solutions obtained by the Cplex solver. These comparisons start by addressing all three methods (GA, VNS and Cplex) for single paper machine instances. Then, some comparisons are made between the GA and Cplex for the multiple paper machine instances, which is the main focus of this paper.

Single paper machine instances

Table 9 shows the percentage of solution feasibility and optimality obtained by each method. Cplex displays optimality only for the smallest problem instances (100% of optimality) and starts having problems finding feasible solutions when the planning horizon reaches 30 days. Both GA and VNS always find feasible solutions for single paper machine instances. The optimality is only analysed for the first test class, as for the other classes the optimal solution is not available (marked in the table with a dash).

Table 10 shows the percentage of the first, second and third best solutions obtained by each method. Note that the sum of these three values for a certain method is equal to this method's feasibility percentage (Table 9). Cplex is more effective when solving small and medium instances, while the GA outperforms

Table 9: Percentage of solution feasibility (for each method) and solution optimality (for Cplex). Instances with one paper machine split by planning horizon size, number of grammage and number of sub-periods.

$\|T\|$	$\|K\|$	$\|S\|$	Cplex	GA	VNS
7	8	3	100.00%(100.00%)	100.00%(0.00%)	100.00%(8.00%)
7	8	4	100.00%(0.00%)	100.00%(-)	100.00%(-)
7	16	3	100.00%(0.00%)	100.00%(-)	100.00%(-)
7	16	4	100.00%(0.00%)	100.00%(-)	100.00%(-)
15	8	3	100.00%(0.00%)	100.00%(-)	100.00%(-)
15	8	4	100.00%(0.00%)	100.00%(-)	100.00%(-)
15	16	3	100.00%(0.00%)	100.00%(-)	100.00%(-)
15	16	4	100.00%(0.00%)	100.00%(-)	100.00%(-)
30	16	3	80.00%(0.00%)	100.00%(-)	100.00%(-)
30	16	4	20.00%(0.00%)	100.00%(-)	100.00%(-)

the other methods in large problem instances (bottom rows of Table 10). In addition, the VNS shows more regularity, since it occupies second places more frequently.

Table 10: Percentage of first, second and third best solutions per method and test class.

$\|T\|$	$\|K\|$	$\|S\|$	Cplex	GA	VNS
7	8	3	(100% / 0% / 0%)	(0% / 10% / 90%)	(8% / 86% / 6%)
7	8	4	(98% / 2% / 0%)	(0% / 8% / 92%)	(14% / 82% / 4%)
7	16	3	(58% / 40% / 2%)	(2% / 8% / 90%)	(42% / 50% / 8%)
7	16	4	(20% / 70% / 10%)	(0% / 10% / 90%)	(80% / 20% / 0%)
15	8	3	(88% / 12% / 0%)	(12% / 78% / 10%)	(0% / 10% / 90%)
15	8	4	(64% / 26% / 10%)	(26% / 54% / 20%)	(10% / 20% / 70%)
15	16	3	(36% / 46% / 18%)	(58% / 40% / 2%)	(6% / 14% / 80%)
15	16	4	(2% / 52% / 46%)	(96% / 4% / 0%)	(2% / 44% / 54%)
30	16	3	(0% / 18% / 62%)	(100% / 0% / 0%)	(0% / 82% / 18%)
30	16	4	(0% / 0% / 20%)	(100% / 0% / 0%)	(0% / 100% / 0%)

Table 11 shows the results of each method for all single paper machine instances. Columns Min, Aver and Max show, respectively, the minimum, average and maximum Gap for each method, considering the average gap of 5 runs for each instance in the case of GA and VNS. These results demonstrate that both VNS and GA become more competitive in larger instances.

Figure 11 shows the performance chart of the three methods for all instances. Cplex presents a higher amount of best results; however, its performance curve is quickly overcome by the GA curve around $\tau = 1.02$, which means that GA finds more solutions with at most 2% of the gap for the best solution of all three methods. VNS outperforms the Cplex curve around $\tau = 1.14$. Furthermore, GA reaches 100% of the instances with $\tau < 1.26$, which means that any solution obtained by the GA in these tests is at most 26% worse than any other method solutions.

Multiple paper machine instances

In Table 12, the results demonstrate the difficulty of solving the problem with two paper machines. Cplex were unable to prove the solution optimality, even for instances with short planning horizons (7 days) and only a few grammages (8 grammages). For instances with medium planning horizon (15 days), Cplex found it difficult to obtain feasible solutions, while GA was able to obtain them for most instances. For instances with 30 days, Cplex did not find any feasible solution and GA was able to find them for just a few instances.

Several characteristics can cause difficulties in determining feasible solutions within the production systems with multiple machines. The critical units in a pulp and paper mill are stopped only in special cases, as

Table 11: Methods performance with minimum, average and maximum gap for each instance class with one paper machine and considering the GA and VNS average values (of the 5 algorithm runs).

$\|T\|$	$\|K\|$	$\|S\|$	Cplex			GA			VNS		
			Min	Aver	Max	Min	Aver	Max	Min	Aver	Max
7	8	3	**0.01%**	**0.01%**	**0.01%**	0.48%	2.76%	5.97%	0.28%	1.26%	4.31%
7	8	4	**6.21%**	**9.38%**	**13.59%**	6.55%	11.78%	17.02%	6.47%	10.39%	14.93%
7	16	3	**7.28%**	**13.49%**	24.07%	13.86%	18.78%	26.69%	8.20%	13.60%	**21.38%**
7	16	4	17.84%	24.56%	**31.92%**	21.41%	28.62%	37.86%	**16.48%**	**23.27%**	32.00%
15	8	3	**5.84%**	**14.12%**	**21.84%**	8.20%	15.88%	22.96%	9.93%	18.59%	26.50%
15	8	4	**9.75%**	**17.73%**	**25.48%**	11.59%	18.97%	25.55%	13.47%	21.11%	28.43%
15	16	3	**13.50%**	22.96%	30.50%	15.09%	**22.41%**	**27.05%**	17.88%	26.05%	33.74%
15	16	4	18.96%	30.89%	46.53%	**16.97%**	**24.77%**	**31.23%**	24.04%	31.07%	39.79%
30	16	3	25.28%	43.10%	69.59%	**11.99%**	**17.02%**	**23.87%**	23.66%	33.07%	46.30%
30	16	4	38.97%	50.60%	62.22%	**12.63%**	**14.58%**	**16.52%**	26.45%	30.47%	34.48%

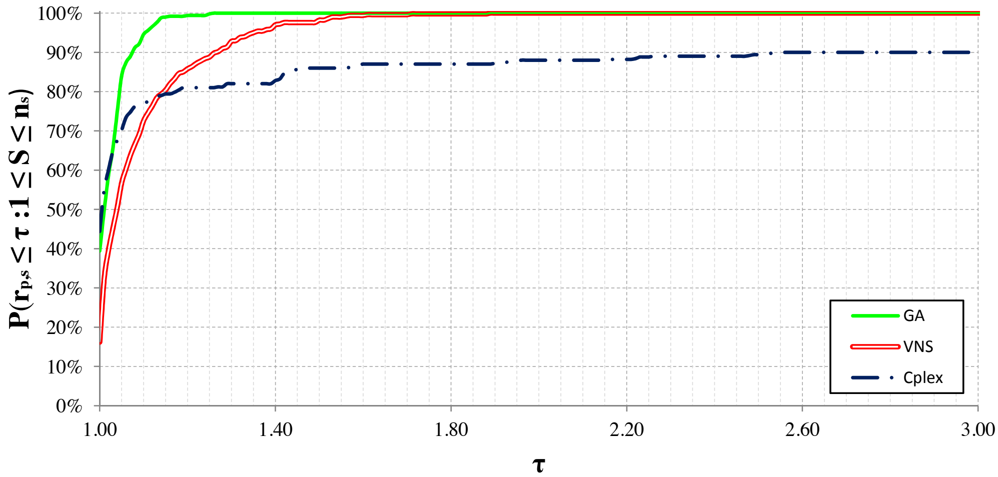


Figure 11: Performance charts for all three methods.

Table 12: Percentage of feasibility and optimality for both methods, Cplex and GA. Instances with two paper machines split by planning horizon size, number of grammage and number of sub-periods.

$\|T\|$	$\|K\|$	$\|S\|$	Cplex	GA
7	8	6	100.00%(0.00%)	100.00%(-)
7	8	8	100.00%(0.00%)	100.00%(-)
7	16	6	100.00%(0.00%)	100.00%(-)
7	16	8	70.00%(0.00%)	100.00%(-)
15	8	6	100.00%(0.00%)	100.00%(-)
15	8	8	90.00%(0.00%)	100.00%(-)
15	16	6	30.00%(0.00%)	94.00%(-)
15	16	8	30.00%(0.00%)	92.00%(-)
30	16	6	0.00%(0.00%)	16.00%(-)
30	16	8	0.00%(0.00%)	12.00%(-)

breakdowns and maintenance, for example. For this reason, the capacity constraints of the paper machines and the digester are considered as equalities rather than limited by these capacities. It may be difficult to define a production plan in which the capacities of all critical units are fully utilized without violations in the pulp inventories limits. Furthermore, the smoothing in the speed variation limits further more this set of feasible solutions, thus justifying the methods' difficulties in finding feasible solutions for some instances.

Table 13 shows the percentage of first and second places of both methods. Cplex had more first places only in the smallest test set, showing the difficulties to solve problem instances with multiple machines.

Table 13: Percentage of first and second positions aggregated per method and test class.

$\|T\|$	$\|K\|$	$\|S\|$	Cplex	GA
7	8	6	(66.00% / 34.00%)	(34.00% / 66.00%)
7	8	8	(4.00% / 96.00%)	(96.00% / 4.00%)
7	16	6	(0.00% / 100.00%)	(100.00% / 0.00%)
7	16	8	(0.00% / 70.00%)	(100.00% / 0.00%)
15	8	6	(0.00% / 100.00%)	(100.00% / 0.00%)
15	8	8	(0.00% / 90.00%)	(100.00% / 0.00%)
15	16	6	(2.00% / 28.00%)	(94.00% / 0.00%)
15	16	8	(2.00% / 28.00%)	(92.00% / 0.00%)
30	16	6	(0.00% / 0.00%)	(16.00% / 0.00%)
30	16	8	(0.00% / 0.00%)	(12.00% / 0.00%)

The same pattern is shown in Table 14, where the GA had better results for almost all test classes. In the first test set, Cplex produced better solutions; however, GA was very competitive and the difference between both average values was less than 2%. On average, GA presented better results for all other test classes and the difference between the GA and Cplex averages was about 30% in some classes.

Table 14: Methods performance with minimum, average and maximum gap for each instance class with two paper machine and considering the GA and VNS average values (of the 5 algorithm runs).

$\|T\|$	$\|K\|$	$\|S\|$	Cplex			GA		
			Min	Aver	Max	Min	Aver	Max
7	8	6	**58.41%**	**63.81%**	**68.44%**	61.03%	65.50%	70.62%
7	8	8	**68.59%**	76.24%	80.38%	69.92%	**71.90%**	**74.13%**
7	16	6	72.19%	80.51%	88.87%	**61.80%**	**69.24%**	**73.37%**
7	16	8	79.70%	87.25%	93.76%	**64.35%**	**69.89%**	**77.02%**
15	8	6	54.33%	62.68%	71.50%	**43.06%**	**48.51%**	**55.68%**
15	8	8	79.67%	86.73%	93.88%	**43.36%**	**48.43%**	**54.24%**
15	16	6	78.18%	82.61%	84.85%	**51.29%**	**59.90%**	**72.48%**
15	16	8	84.91%	86.55%	88.74%	**50.86%**	**55.17%**	**57.90%**

Figure 12 presents the performance curve of the GA compared to the curve of the Cplex concerning instances with two machines. The GA curve dominated the Cplex curve in the chart, whch means that GA had a higher percentage of better solutions and a higher percentage for each τ chosen. GA started with 74.4% of better solutions and the curve converged in 81.4%, reached with $\tau = 1.44$. Cplex increased from 7.4% to 62%, reaching it at $\tau = 8.59$.

Both methods (GA and CPLEX) were rejected by the Anderson-darling test for the two paper machine instances ($p\text{-}value_{GA,A}^{M2,3600} = 8.05 \cdot 10^{-6}$ and $p\text{-}value_{CPLEX,A}^{M2,3600} = 0.01933$). The null hypothesis was rejected in the Wilcoxon test with $p\text{-}value_{GA\text{-}CPLEX,W}^{M2,62,3600} = 1.06 \cdot 10^{-10}$, which ensures that the proposed method has superior results for this set of tests, within the predefined degree of confidence.

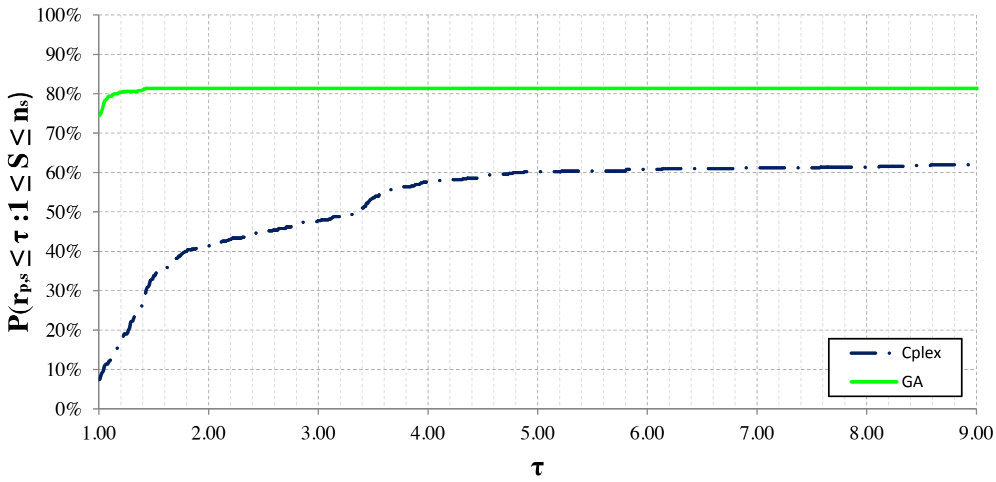


Figure 12: Performance charts to compare the Cplex and GA solution quality.

5.2.3. Additional computational tests

A sub-set of the two paper machine problem instances was also solved imposing time limits of 1800 seconds (half hour) and 10800 seconds (three hours) in order to verify the assymptotic performance of the proposed method in comparison with the CPLEX solver. This was done only to instances with multiple paper machines as these are the main focus of this paper. Only instances with 8 micro-periods per period (day) were considered in the tests, totaling 50 instances. The CPLEX was able to solve 27 instances within the time limit of 1800 seconds and 36 instances within the time limit of 10800 seconds. On the other hand, the proposed method was not able to solve only one instance of this sub-set of instances.

The solutions obtained by the GA cannot be accepted as normally distributed when applying the Anderson-Darling test to the cases: $p\text{-}value^{M2,1800}_{GA,A} = 5.28 \cdot 10^{-5}$ and $p\text{-}value^{M2,10800}_{GA,A} = 0.00749$, making not possible to perform the t-test. The results reported by the Wilcoxon tests were similar to those found for the time limit of 3600 seconds, in which the null hypotheses were rejected in both cases: $p\text{-}value^{M2,26,1800}_{GA\text{-}CPLEX,W} = 1.49 \cdot 10^{-8}$ and $p\text{-}value^{M2,35,10800}_{GA\text{-}CPLEX,W} = 2.63 \cdot 10^{-8}$.

6. Conclusions

This paper addressed the lot-sizing and scheduling problem in pulp and paper mills with multiple paper machines, extending previous studies concerning the single paper machine. This problem is represented as an MIP model by extending a literature formulation. A hybrid approach based on a GA and a residual LP is developed to solve this multi-machine problem. The method includes some well-known genetic operators and novel ingredients, namely: a diversification process, the indirect codification solution, mutation operators and the insertion of individuals of different sizes in the same population. Some computational tests were conducted to define the best combination between the features developed for the GA considering part of the problem instances. The complete computational tests were performed by considering only the best GA variant, the single paper machine approach proposed by [7] and the MIP solver Cplex 12.2. The computational results support the following conclusions:

- The approach provided here is competitive in medium to large problem instances when compared to other methods for single-machine problem instances. In the multi-machine case, the method outperforms Cplex even in small problem instances.
- The advantages/disadvantages introduced by each feature of the proposed GA (namely the use of a diversification process, the acceptance of individuals with different sizes in the same population, the use of hot-start solutions and the use of larger initial population) were carefully analyzed.

- Complexity increases when a bigger number of sub-periods is considered to create more flexibility between machines' production. In this case, adjusting the number of sub-periods in real time is more advantageous.
- For larger problem instances with multiple paper machines, the GA has some problem finding feasible solutions. A possible alternative to overcome this difficulty would be to solve a detailed plan just at the beginning of the planning horizon (first periods) and aggregate the subsequent decisions in the horizon (remaining periods), in a rolling horizon solution process.

Interesting future research topics would be to develop alternative mathematical models to represent this and other more general lot-sizing and scheduling problems in the pulp and paper industry. Concerning the GA, an interesting extension would be to treat individuals with different number of sub-periods per period, as the number of production campaigns varies between periods, according to demand and other characteristics. Using multiple populations and a multi-thread implementation can also be alternatives to boost the method's performance.

Acknowledgements

This paper was partly funded by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), process number 233210/2012-1, and Fundação de Amparo à Pesquisa do Estado de São Paulo (Fapesp), contract number 2011/00274-8. The second author would also like to thank to project "NORTE-07-0124-FEDER-000057" funded by the North Portugal Regional Operational Programme, under the National Strategic Reference Framework (NSRF), through ERDF, and by national funds through FCT. The authors are thankful to Gonçalo Figueira for kindly sending the codes of his generator and the VNS approach, and for the support provided when adapting the method.

References

[1] I.-L. Svensson, J. Jönsson, T. Berntsson, B. Moshfegh, Excess heat from kraft pulp mills: Trade-offs between internal and external use in the case of Sweden–Part 1: Methodology, Energy Policy 36 (11) (2008) 4178 – 4185.

[2] J. Jönsson, I.-L. Svensson, T. Berntsson, B. Moshfegh, Excess heat from kraft pulp mills: Trade-offs between internal and external use in the case of Sweden–Part 2: Results for future energy market scenarios, Energy Policy 36 (11) (2008) 4186 – 4197.

[3] H. Bouchriha, M. Ouhimmou, S. D'Amours, Lot sizing problem on a paper machine under a cyclic production approach, International Journal of Production Economics 105 (2) (2007) 318 – 328.

[4] S. Poltroniere, K. Poldi, F. Toledo, M. Arenales, A coupling cutting stock-lot sizing problem in the paper industry, Annals of Operations Research 157 (2008) 91–104.

[5] M. H. Correia, J. F. Oliveira, J. S. Ferreira, Integrated resolution of assignment, sequencing and cutting problems in paper production planning, International Journal of Production Research 50 (18) (2012) 5195–5212.

[6] M. O. Santos, B. Almada-Lobo, Integrated pulp and paper mill planning and scheduling, Computers & Industrial Engineering 63 (1) (2012) 1 – 12.

[7] G. Figueira, M. Oliveira Santos, B. Almada-Lobo, A hybrid VNS approach for the short-term production planning and scheduling: A case study in the pulp and paper industry, Computers & Operations Research .

[8] J. H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, University of Michigan Press, 1975.

[9] D. Whitley, A genetic algorithm tutorial, Statistics and computing 4 (2) (1994) 65–85.

[10] H. G. Goren, S. Tunali, R. Jans, A review of applications of genetic algorithms in lot sizing, Journal of Intelligent Manufacturing 21 (4) (2010) 575–590.

[11] C. F. M. Toledo, P. M. França, R. Morabito, A. Kimms, Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem, International Journal of Production Research 47 (11) (2009) 3097–3119.

[12] M. Santos, S. Massago, B. Almada-Lobo, Infeasibility handling in genetic algorithm using nested domains for production planning, Computers & Operations Research 37 (6) (2010) 1113–1122.

[13] P. Amorim, C. H. Antunes, B. Almada-Lobo, Multi-objective lot-sizing and scheduling dealing with perishability issues, Industrial & Engineering Chemistry Research 50 (6) (2011) 3371–3381.

[14] M. Mohammadi, S. F. Ghomi, N. Jafari, A genetic algorithm for simultaneous lotsizing and sequencing of the permutation flow shops with sequence-dependent setups, International Journal of Computer Integrated Manufacturing 24 (1) (2011) 87–93.

[15] C. F. M. Toledo, M. d. S. Arantes, R. R. R. Oliveira, B. Almada-Lobo, Glass container production scheduling through hybrid multi-population based evolutionary algorithm, Applied Soft Computing 13 (3) (2013) 1352–1364.
[16] V. C. Camargo, L. Mattiolli, F. Toledo, A knapsack problem as a tool to solve the production planning problem in small foundries, Computers & Operations Research 39 (1) (2012) 86–92.
[17] H. K. Chan, S. H. Chung, T. M. Chan, Combining genetic approach and integer programming to solve multi-facility economic lot-scheduling problem, Journal of Intelligent Manufacturing 23 (6) (2012) 2397–2405.
[18] C. F. M. Toledo, L. d. Oliveira, R. d. F. Pereira, P. M. França, R. Morabito, A genetic algorithm/mathematical programming approach to solve a two-level soft drink production problem, Computers & Operations Research 48 (0) (2014) 40 – 52.
[19] J. Puchinger, G. R. Raidl, Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification, in: Artificial intelligence and knowledge engineering applications: a bioinspired approach, Springer, 41–53, 2005.
[20] H. Meyr, Simultaneous lotsizing and scheduling by combining local search with dual reoptimization, European Journal of Operational Research 120 (2) (2000) 311–326.
[21] L. A. Wolsey, Solving multi-item lot-sizing problems with an MIP solver using classification and reformulation, Management Science 48 (12) (2002) 1587–1602.
[22] N. Dellaert, J. Jeunet, Solving large unconstrained multilevel lot-sizing problems using a hybrid genetic algorithm, International Journal of Production Research 38 (5) (2000) 1083–1099.
[23] I. Moon, B. Cha, H. Bae, Hybrid genetic algorithm for group technology economic lot scheduling problem, International Journal of Production Research 44 (21) (2006) 4551–4568.
[24] V. K. Koumousis, C. P. Katsaras, A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance, Evolutionary Computation, IEEE Transactions on 10 (1) (2006) 19–28.
[25] T. Vidal, T. G. Crainic, M. Gendreau, N. Lahrichi, W. Rei, A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems, Operations Research 60 (3) (2012) 611–624.
[26] Z. Michalewicz, A Survey of Constraint Handling Techniques in Evolutionary Computation Methods, Evolutionary Programming 4 (1995) 135–155.
[27] D. E. Goldberg, The design of innovation: Lessons from and for competent genetic algorithms, vol. 7, Springer, 2002.
[28] D. E. Goldberg, Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking, Complex Systems 5 (1990) 139–167.
[29] W. M. Spears, K. A. De Jong, An analysis of multi-point crossover, Tech. Rep., DTIC Document, 1990.
[30] A. Brindle, Genetic Algorithm for Function Optimization, Ph.D. thesis, Ph. D. dissertation, University of Alberta, 1981.
[31] E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming 91 (2) (2002) 201–213.