Annual Production Budget in the Beverage Industry

Luis Guimarães[a], Diego Klabjan[b], Bernardo Almada-Lobo[a]

[a] *Faculdade de Engenharia da Universidade do Porto, Porto, Portugal*
[b] *Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois*

Abstract

Driven by a real-world application in the beverage industry, this paper provides a design of a new VNS variant to tackle the annual production budget problem. The problem consists on the assigning and scheduling of production lots in a multi-plant environment, where each plant has a set of filling lines that bottle and pack drinks. Plans also consider final product transfers between the plants. Our algorithm fixes setup variables for family of products and determines production, inventory and transfer decisions by solving a linear programming (LP) model. As we are dealing with very large problem instances, it is inefficient and unpractical to search the entire neighborhood of the incumbent solution at each iteration of the algorithm. We explore the sensitivity analysis of the LP to guide the partial neighborhood search. Dual re-optimization is also used to speed-up the solution procedure. Tests with instances from our case study have shown that the algorithm can substantially improve the current business practice, and it is more competitive than state-of-the-art commercial solvers and other VNS variants.

Keywords: Long-term production planning, Beverage industry, Very large neighborhood search, Mathematical programming

1. Introduction

The beverage industry is a sub-sector of the food industry, the second largest sector in the European manufacturing industry in terms of value added. It supplies a variety of products from wine, beer and spirits to mineral and sparkling water and soft drinks. Markets world wide are strongly affected by cultural differences, especially in Europe. This effect creates the environment for the appearance of small to medium size companies that are specialised in local products and/or local brands. Nevertheless, there are a number of large multinational companies able to compete in markets across the globe offering a wide variety of products, such as soft drinks. Today's competition in this sector leads companies to expand their product portfolio, which combined with the advanced technology present in modern production sites, raises the need for efficient production planning. Moreover, production sites in this industry tend to be geographically disperse allowing companies to satisfy local demands at lower costs. Production planning is often conducted considering only one plant at time, ignoring the potential benefits of coordination. This paper is inspired by a real industrial case from a company competing in the beer and soft drink industries. The focus is to define a long-term production plan to a series of production (filling) lines located in different plants. The scheduling of product families at each filing line is the basis for product production, inventory, and transfer decisions. Transfer decisions represent movements of finished products and come from the fact that demand observed at a geographical area around each plant can be satisfied by other production sites to cope with under capacity of a given plant. Under these conditions, plants act both as production and distribution centers since warehouses are located near them and have individual demand. Decisions are traditionally made for a rolling planning horizon of 12 to 18 months with a monthly bucket. Real-world production planning problems often result in untractable models, and even simplified versions result in NP-hard problems. However, only realistic modelling of the problem features can help managers in their decisions, which was already pointed out as a field of future research of two previous literature reviews on production

Email addresses: guimaraes.luis@fe.up.pt (Luis Guimarães), d-klabjan@northwestern.edu (Diego Klabjan), almada.lobo@fe.up.pt (Bernardo Almada-Lobo)

planning problems [1, 2]. Furthermore, to deal with the complexity of industrial applications, Jans and Degraeve [2] encourage the use of metaheuristics. Naturally, the large scale instances that arise in our application demand their use. Metaheuristics are frameworks used to solve combinatorial optimization problems, guiding other simple heuristics to search for high quality solutions. Local search (or neighborhood search) is among these heuristics. They attempt to iteratively improve an incumbent solution by replacing it with a better solution found in its neighborhood leading to a local optimum. Several schemes have been developed to overcome the entrapment in local optima. Variable Neighborhood Search (VNS) is a local search framework based on systematic change of neighborhoods both to find local optimum and to perturb the solution to emerge from entrapment [3]. We make use of VNS principles and of our problem formulation to develop a heuristic for the problem. Our local search attempts to find a better assignment of families to filling lines, and the subsequent decisions on production, inventory and transfers are achieved through linear programming (LP). This neighbor evaluation methodology can be expensive as we are dealing with very large neighborhoods, therefore we test different techniques to speed-up the local search. The final tableu of the LP simplex algorithm provides valuable information that we use to guide the search. Other, speed-up techniques involve the use of dual-reoptimization to quickly identify and get rid of low quality solutions. Tests performed on a set of randomly generated instances attested the algorithm superiority over commercial solvers and other VNS variants for medium and large size instances. Later the benchmark against industry current practice revealed its potential cost saving capability. The remainder of the paper is as follows. In Section 2 we start by describing the production process of the two different types of products (beer and soft drinks) tackled here. Planning process and major planning constraints are introduced in Section 3. We also integrate long-term planning in the industry planning framework and pinpoint industry practices. Section 4 presents the real-world case study and related work in the literature. Section 5 is dedicated to the problem modelling and solution methodology. Section 6 reports numerical experiments, first on a collection of randomly generated instances to asses the quality and robustness of the algorithm, and later on real-world instances, together with a benchmark against the current practice at a company. Finally, the paper ends with a short summary and outlook.

2. The beer and soft-drink production process

Beer and soft drinks industries share some common features in their production process. Both encompass two main production stages: liquid production (stage I) and liquid bottling (stage II).

Stage I of the beer production process, also known as brewing, it has the purpose of converting the sugars present in the starch source into alcohol through a reaction of a yeast. Different beers have different recipes that determine their production process. Yet, generally there are three main processes in beer production: wort preparation, fermentation and maturation, and filtering. Wort preparation consists in the extraction of the fermentable sugars from usually barley malt and, in addition, of hop. Fermentation follows next and its goal is to transform the sugars into ethanol through the action of fermenting yeasts. Undesirable substances from the censorial point of view, are removed in maturation in a series of chemical, biological and physical steps. Fermentation and maturation processes have the longest processing times, depending on the beer recipe, they can last from 4 days to 3 weeks. The beer resulting from the previous process is turbid, therefore a filtration process is conducted. During this step for some flavoured beer, syrups or concentrates are also added. Non-alcoholic beers pass through a stripping process to remove the ethanol. For more detail about the beer production process the reader is referred to [4–6].

On the other hand, soft drinks are beverages consisting primarily of carbonated water, sugar, and flavourings. Stage I of soft drink production starts with water clarification. Liquid flavour preparation follows next, and is conducted in specialized mixing tanks. Sugar, flavour concentrates and water are pumped in a specific sequence and then carefully mixed. Sophisticated machines control the flow of the ingredients to ensure the perfect recipe. Carbonation is generally the last step in soft drink production, normally performed just before liquid bottling. For more information on soft drink production see [7].

In the second stage, different sized cans, glass bottles (disposable and reusable), kegs and plastic (PET) bottles (less common in the beer case) are filled with beer and soft drinks. A filling line consists on a series of conveyor belts and machines that wash, fill, seal, label and pack the bottles, cans or kegs [8, 9]. The first step involves washing and disinfection of containers, which afterwards pass through an inspection to guarantee the absence of potential hazards. The next machine performs the filling and capping of containers. To ensure product shelf life over a determined period, a pasteurization step follows container filling. For soft drinks, the pasteurization step may take place in a mixing tank instead. Its duration depends on the product features. Labelling is carried out next. Filled containers inspection certificates that the specified volume has been introduced and no defects occurred during the process. Packing containers into paper-boxes, packs or other selling units precedes palletization and storage. Since these processes are done in series from hereafter, we will refer to the set of machines that compose a filling line as a whole.

3. Planning production in the beverage industry

Planning production in the beverage industry, specifically in the beer industry, is a complex process. Not only are there several processes involved, but also increasing competitiveness of the market forces companies to enlarge their product portfolio posing new challenges and raising the need for decision support tools to help managers.

3.1. Main planning constraints

One of the main planning constraints is related to the sales profile of these products. Sales of beer and soft drinks have high seasonality and variability. Beer and soft drinks consumption peaks at Easter and Christmas, but Summer is by far the highest point in terms of sales. Moreover, there is a clear increase of sales in the second half of each month. On the other hand, capacity remains almost constant throughout the year and it can be evaluated by the number of production hours available. Product demand is also affected by other sources of variability, such as brand management and clients commercial policy. Some of the most important costumers of these companies are large retailers with extremely aggressive marketing strategies that require almost instantaneous response from suppliers. These sales characteristics stress production and lead the industry to work on a make-to-stock basis. But, diversity of the product range makes sales hard to forecast.

Looking at the industry supply chain, a typical beverage industry company has one or more plants relatively close to the geography of demand, in order to avoid transfer costs, which otherwise would have assumed an important percentage of the total cost. Within each plant, stages I and II of the production process are most of the time divided and buffers may exist between stages, with typically a single unit of stage I supplying a series of parallel filling lines. It is a common practice in industry to consider the filling stage (II) as the bottleneck of the entire production process, due to several reasons. For the beer case, buffers between the different main processes of stage I allow it to be more flexible. Moreover, the high number of different products that have to be manufactured in stage II correspond to a few different types of beer (or syrups for the soft drinks) in stage I, since SKU (Stock Keeping Unit) differences may rely on a different container, label or package affecting only stage II.

Filling lines are usually divided according to their technological aspects (e.g. filling lines for kegs are unable to fill bottles or cans). Furthermore, an important distinction is made between filling lines for disposable and reusable bottles, since an extra step and machine are needed to conduct an additional washing procedure in the latter case. Hence, disposable bottles filling lines can not fill reusable bottles, but no restrictions are present the other way around. Even so, filling lines are relatively flexible and often a certain product can be assigned to several alternative lines, even within the same plant, but with throughput rates (measured in terms of litres per minute (l/min)), that might be substantially different.

Each filling line can only produce a single product at any time, being adjusted to fill a certain liquid, container type and size, and final package. A product changeover may involve several changes in the filling line and possibly a cleaning step. Liquid type switchovers always involve the cleaning of the filling line and sometimes the setup of the pasteurization machine. On the other hand, switches on the container type and/or size and final packaging trigger mechanical adjustments in most machine settings. These operations consume scarce production time (capacity) and can cause loss of material, that depend on the production sequence. Therefore we have the presence of sequence-dependent changeover times and costs. The increase of the number of products that took place in recent years has reduced the operational times of filling lines as more setups are needed. In addition, market pressure to work with less stock and to deliver products more frequently has also increased the number of production batches, reducing their size and consequently leveraging the appearance of extra setups.

Filling lines operate on a shift basis and their capacity can be translated into the number of hours available for production. Some of the filling lines operate around the clock, therefore overtime is not always an option. Investing in new lines is also problematic as it greatly increases fixed costs. Some investments can be made in order to make filling lines more flexible, but they have to be carefully studied since their cost can be significant. Yet, such changes would only produce effects in the long-term, and short-term capacity can be considered constant. All the aforementioned reasons raise the issue of efficient production planning as it can guarantee a better utilization of resources and, ultimately, the competitiveness of the company.

In the presence of a multi-plant environment further planning features appear. Some product specialization is possible, aiming to achieve better throughput rates or standard quality requirements in stage I due to larger production batches whose process is easier to control. Nevertheless, for standard products production near the consumption location should yield low cost production plans due to shorter transfer costs.

3.2. Production planning systems in the beverage industry

To face the constraints described above and the different nature of decisions and actions, production planning in the beverage industry is made by several company echelons with different aims and planning horizons. Although decisions are strongly dependent, it is virtually impossible to sustain a single decision model for the entire decision-making process as it would be extremely hard to maintain, solve and interpret. Moreover, market dynamics also determine that high detailed plans for a distant future are in most occasions useless. Planning decisions are therefore made in a hierarchical process composed of three levels: strategic (long-term) planning, tactical (medium-term) planning and operational (short-term) planning. Long term planning assesses investments in the installed capacity, trying to balance capacity with demand for a planning horizon of 12 to 18 months. Concerning tactical planning, the focus is to derive plans for operations, essentially production and distribution, aiming at cost efficiency. Here planning horizons commonly span from 4 to 12 weeks. The lower level of the hierarchy schedules operations to the available resources looking at a very short planning period from 1 day up to 1 week. These levels operate in a rolling horizon approach, only a few periods in the begin of the planning horizon are actually executed, furthermore the output of an upper level constitutes an important input for the following level.

4. The case study

Our study is motivated by a Portuguese company that competes in the beverage industry with sales across the globe. The company holds many nationally very popular brands of beer, soft drinks, and mineral and sparkling water. Production sites are spread around the country, accounting for 8 plants and more than 20 filling lines. Mineral and sparkling water plants are located near a water source, while other production sites are responsible for beer and soft drink production. Only planning of beer and soft drinks has to be done simultaneously as both product types share common production resources and this will be the scope of our study. The aim is to create the annual production budget (PB).

PB is part of the company's annual budgeting process. The budgeting is a vital tool to align company goals and translate the strategy defined into the next 12 months. Annual budgeting starts in mid September and lasts until late October. The first main task is the creation of an annual sales budget (SB). SB is driven by a monthly sales forecast for each product in the following year. In parallel, the production departments of each plant schedule the filling lines maintenance calendar and estimate throughput rates for each product. These throughput rates are approximations based on the previous years and also reflect expected gains or losses of efficiency. Embedded in these estimations are the sequence-dependent setups witnessed in the years before. The goal of the PB is not to obtain a detailed schedule for production lines, but rather an estimation of the adequacy of resource capacity to SB. Therefore, production sequencing is disregarded and capacity loss due to sequence-dependent setups is incorporated in throughput rates.

The PB is conducted by the planning department and aims at validating the SB from an industrial and economical point-of-view. Besides SB and throughput rates, the available capacity is an input determined from the filling lines maintenance calendar and the number of available days for production. Capacity is estimated per filling line and divided into three categories: normal workdays, Saturdays and holidays, and Sundays. This distinguishes normal capacity from overtime. SB is generally distributed among the plants according to the past years sales. However, technological constraints, production quality assurance or product specialization can imply a pre-determined plant. Technological constrains are related to production and filling equipments required to produce certain products. Production quality assurance deals with situations in which minimum batch sizes and/or production frequency may not be achieved if forecasts for family's demand are disaggregated.

PB only accounts for the filling stage, since this stage is considered the production process bottleneck. Therefore, the number of working shifts is decided only for the filling stage. PB decides on the assignment of products to the different filling lines in the planning horizon and the definition of production lotsizes. Through this step, plant inventory and inter-plant transfers are also determined. The objective is to satisfy SB while minimizing inventory holding costs, setup costs, inter-plant transfer costs and overtime costs.

PB conclusion triggers the creation of the transportation and materials procurement budget.

4.1. Company practice and opportunities

The creation of the PB is a hard time-consuming task. The planner is challenged with over 150 products divided by approximately 60 product families and 14 different filling lines, although technological constrains restrict the problem size. One of the strategies used by the company to overcome this problem is the choice of a preferential machine to supply the demand of each product at each plant. Frequently, more than one filling line of the same plant can produce a certain product gross requirement, but the definition of a preferential filling line rule automatically fixes allocation, turning the act of planning easier. Nevertheless, throughout the process is natural that some filling lines exceed their normal capacity. Requirements can be moved to another filling line of the same plant, can anticipate raising inventory holding costs or can be moved to filling lines of other plants originating transfers and/or inventory costs. Another possibility is to use overtime capacity, which is limited to a certain maximum. PB is done once a year and the main key performance indicators regard average filling line utilization, total inventory, transfer and overtime costs.

4.2. Related work

Post literature has focused on the operational and tactical levels and the integration of both in beverage and related industries (e.g. soft-drinks [10], foundries [11], glass industry [12] and animal feed [13]). Nevertheless, some work is also available on medium to long-term planning mostly in terms of mathematical formulations. These models often include production, distribution and inventory management with capacity investments. Chandra and Fisher [14] show that the integration of production and distribution in a single decision model yields better results than optimizing separate models. In [15] decisions on production, transportation, purchasing and warehouse capacity extension are made for a multi-plant and multi-warehouse environment. Martin et al. [16] study a real world case in the flat glass business. Production, distribution, and inventory operations are managed in a single model. An application in the chemical process industry is presented by Timpe and Kallrath [17]. Batch and campaign production in a multi-plant production system are decided along with distribution and marketing decisions. A real-world problem in steel manufacturing is approached by Sambasivan and Yahya [18]. Almada-Lobo et al. [19] present a long term production planning model in the glass industry. A multi-plant production system where each plant has a set of production lines is considered but with common demand, i.e. demand is not attached to a specific location. The significant sequence-dependent setup costs and time that arise in this industry lead to the need of sequencing of product families. A VNS-based heuristic is used to solve the problem. Another industrial example at a metal item manufacturer is given by Dhaenens-Flipo and Finke [20]. The authors formulate an integrated production-distribution model considering a production system composed of multiple factories having several parallel production lines. Sequence-dependent setups are managed through predefined sequences. The need of explicitly considering them at this level is again due to their magnitude. Distribution decisions relate to transfers among plants-warehouses and warehouse-clients. Other examples occur in production environments where items are produced in a series of processes occurring in different plants [21].

5. Solution methodology

In this section we first present a mixed integer optimization model representing the problem that arises in the PB, hereafter called long-term production planning problem (LT3P). Based on this model a solution procedure is described aiming to achieve good quality solutions in limited computational time, which is not accomplished by exact methods.

5.1. Mathematical formulation

The model considers a multi-plant environment with P plants. Each plant has its own individual demand and storage capacity. As mentioned before, a certain plant can supply demand for another plant but additional transfer cost has to be accounted for. Common filling lines force us to simultaneously plan both beer and soft drink products. Furthermore, since filling lines are considered the production bottleneck of the production system decisions are taken only for this step. Products sharing common production features, the same container type and final package, are grouped into product families. The model considers a planning horizon divided into T periods, usually months. To formulate the model the following nomenclature is used:

Indices

i	product: $i \in \mathcal{N} = \{1, \ldots, N\}$
f	family: $f \in \mathcal{F} = \{1, \ldots, F\}$
j, k	plant: $j, k \in \mathcal{P} = \{1, \ldots, P\}$
m	filling line: $m \in \mathcal{M} = \{1, \ldots, M\}$
t	period: $t \in \mathcal{T} = \{1, \ldots, T\}$

Sets

$\mathcal{M}_j$	set of filling lines belonging to plant j
$\mathcal{F}_m$	set of families that can be produced on filling line m
$\mathcal{N}_m^{line}$	set of products that can be produced on filling line m
$\mathcal{N}_f^{fam}$	set of products belonging to family f

Parameters

cap_{mt}	available capacity at filling line m in period t (in time units)
j_m	plant of filling line m
f_i	family of product i
d_{ijt}	demand of product i at plant j at the end of period t
h_{ijt}	unitary holding cost of product i at plant j at the end of period t
r_{ijkt}	unitary transfer cost of product i from plant j to plant k in period t
c_{fmt}	setup cost of family f on filling line m in period t
p_{fmt}	throughput rate of family f on filling line m in period t
b_{imt}	upper bound on production quantity of product i on filling line m in period t

To capture decision making the following variables are defined:

X_{imt}	production quantity of product i on filling line m in period t
I_{ijt}	stock of product i at plant j at the end of period t
W_{ijkt}	transfer quantity of product i from plant j to plant k in period t
Y_{fmt}	(=1) if a setup occurs to family f on filling line m in period t, (=0) otherwise.

The model is stated as follows:

$$\min \quad Obj_1 = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{P}} \sum_{t \in \mathcal{T}} \left(h_{ijt} \cdot I_{ijt} + \sum_{k \in \mathcal{P}} r_{ijkt} \cdot W_{ijkt} \right) + \sum_{f \in \mathcal{F}_m} \sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}} c_{fmt} \cdot Y_{fmt} \tag{1}$$

$$I_{ij,t-1} + \sum_{m \in \mathcal{M}_j} X_{imt} + \sum_{k \in \mathcal{P} \setminus \{j\}} W_{ikjt} = I_{ijt} + d_{ijt} + \sum_{k \in \mathcal{P} \setminus \{j\}} W_{ijkt}, \quad \forall\, i \in \mathcal{N},\ j \in \mathcal{P},\ t \in \mathcal{T} \tag{2}$$

$$\sum_{i \in \mathcal{N}_m^{line}} \frac{X_{imt}}{p_{f_i mt}} \leq cap_{mt}, \quad \forall\, m \in \mathcal{M},\ t \in \mathcal{T} \tag{3}$$

$$X_{imt} - b_{imt} \cdot Y_{f_i mt} \leq 0, \quad \forall\, i \in \mathcal{N}_m^{line},\, m \in \mathcal{M},\, t \in \mathcal{T} \tag{4}$$

$$(I_{ijt},\, W_{ijkt},\, X_{imt}) \geq 0,\; Y_{fmt} \in \{0, 1\} \tag{5}$$

The objective function (1) minimizes the sum of the holding, transfer and setup costs. Inventory balance constraints (2) control product flow in each plant. Demand for product i at plant j in period t is either met by available stock, production within the plant or from transfers from other plants, not considering backlogging. Without loss of generality, we assume that transfer cost r_{ijkt} satisfies the triangular inequality, $r_{ijkt} \leq r_{ijlt} + r_{ilkt}$ for all $i \in \mathcal{N}$, $(j, k, l) \in \mathcal{P}$, $t \in \mathcal{T}$. Links between setup and production variables are guaranteed in (4). Production of product i on filling line m in period t can only occur if the filling line has been set up for the respective family f_i ($Y_{f_i mt} = 1$). Additionally, production is limited to $b_{imt} = min\,\{cap_{mt} \cdot p_{f_i mt},\, \sum_{j \in \mathcal{P}} \sum_{u=t}^{T} d_{iju}\}$.

The problem described is similar to the single stage, multi-plant, multi-item and multi-period capacitated lot sizing problem (MPCLSP) described in [18, 22, 23]. Few papers address this variant of the standard capacitated lot sizing problem (CLSP). In [22] the authors describe a heuristic to solve the problem based on transfers of production lots. The paper [18] presents a heuristic based on Lagrangian relaxation. The authors dualize capacity constraints and solve the N uncapacitated subproblems via reformulation into a set of shortest path problems with common fixed-charge constraints. Computational experiments are conducted with instances of up to 15 products, 6 periods and 4 plants. Nascimento et al. [23] propose a greedy randomized adaptive search procedure (GRASP) combined with path-relinking. Results are compared to the method described in [18] and the authors claim to achieve a better performance in terms of the mean gap of the linear relaxation of the problem. In addition, the proposed heuristic was also tested in the parallel machine lot sizing problem, which in fact is a special case of MPCLSP, when transfers among plants are discarded. In this scenario each plant corresponds to a machine.

Although similar to the MPCLSP, our model has different assumptions. As we can not agregate machine resources of the same plant due to technological constraints, each plant may have one or more machines, contrarily to the MPCLSP that assumes a single machine. Moreover, setup times are not considered here as throughput rates used by the company already include them considering an average lotsize. Still, such a generalization could be easily made considering s_{fmt}, the time to set up family f on machine m in period t, and replacing constraints (3) by:

$$\sum_{i \in \mathcal{N}_m^{line}} \frac{X_{imt}}{p_{f_i mt}} + \sum_{i \in \mathcal{F}_m} s_{fmt} \cdot Y_{fmt} \leq cap_{mt}, \quad \forall\, m \in \mathcal{M},\, t \in \mathcal{T}.$$

In addition, production costs are neglected, which could be overcome by introducing the parameter v_{imt} defining the unitary production cost of product i on machine m in period t, and adding production cost into the objective function (1):

$$\min \quad Obj_2 = Obj_1 + \sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}_m^{line}} v_{imt} \cdot X_{imt}.$$

Finally, setups are not considered in terms of products but rather of product families. Such assumption relies on the purpose of our model. The setup costs try to minimize the number of production lines producing the same family as an indicator of future capacity losses at the operational level when sequence dependent setup times are considered. Products within each family are strongly related representing minor setups among them as few characteristics vary from one to another. In other words, sequence-dependent setups among products of different families are much more costly in terms of time and cost. Hence, by minimizing the number of times a family is produced we are also faced with a reduced number of major setups between families in detailed plans. Regardless these observations, if each family is only composed by a single product we end up with setups defined by product.

Further modifications to the model incorporate other important planning decisions. One of them is the use of overtime. Overtime can be used to face lack of production capacity and is especially important during peak seasons. Distinction is made between overtime on Saturdays and holidays (type I) and overtime on Sundays (type II). Type I overtime is less costly than type II. To introduce these decisions in the model we first need to define the following parameters and decision variables.

Parameters

co^{I}_{mt} (co^{II}_{mt}) unitary cost of an extra time unit of type I (type II) overtime on filling line m in period t

mo^{I}_{mt} (mo^{II}_{mt}) maximum available overtime capacity of type I (type II) on filling line m in period t

Variables

O^{I}_{mt} (O^{II}_{mt}) overtime of type I (type II) used on filling line m in period t

To integrate overtime decisions in the model, objective function (1) must be transformed into:

$$\min \quad Obj_3 = Obj_1 + \sum_{m\in\mathcal{M}}\sum_{t\in\mathcal{T}}\left(co^{I}_{mt}\cdot O^{I}_{mt} + co^{II}_{mt}\cdot O^{II}_{mt}\right), \tag{6}$$

and constraints (3) become:

$$\sum_{i\in\mathcal{N}^{line}_{m}} \frac{X_{imt}}{p_{f_i mt}} \leq cap_{mt} + O^{I}_{mt} + O^{II}_{mt}, \quad \forall\, m\in\mathcal{M},\ t\in\mathcal{T}. \tag{7}$$

Finally, the following constraints impose limits on overtime utilization:

$$0 \leq O^{I}_{mt} \leq mo^{I}_{mt},\ 0 \leq O^{II}_{mt} \leq mo^{II}_{mt}, \quad \forall\, m\in\mathcal{M},\ t\in\mathcal{T}. \tag{8}$$

Hence, the overall LT3P model reads:

$$\min Obj_3$$
$$\text{satisfying}\,(2),(4),(7)-(8),$$
$$(I_{ijt},\ W_{ijkt},\ X_{imt},\ O^{I}_{mt},\ O^{II}_{mt}) \geq 0,\ \ Y_{fmt}\in\{0,1\}.$$

5.2. Solution procedure

Standard single-item CLSP has been proven to be NP-hard [24], so are the respective multi-item and multi-plant versions. In this paper, we present results that confirm the difficulty of solving to optimality moderate and large size instances, thus motivating heuristics to find approximate solutions to the problem. We propose heuristics inspired by the VNS principles. As briefly mentioned before, VNS systematically exploits the change of neighborhood both in descent to local optima and in escape from them [3]. VNS relies on local search heuristics that starting from an initial solution x attempt to find an improvement within a neighborhood $N(x)$. Until such an improvement is possible, the heuristic iterates, otherwise it stops. To create a VNS scheme, one must provide a set of pre-defined neighborhoods structures $N_k(k = 1, \ldots, k_{max})$ and an initial solution x. The initial solution can be obtained from any simple construction heuristic. A basic scheme of VNS (see [25]) combines stochastic and deterministic changes of neighborhoods using the following three steps that are repeated until the stopping criteria is reached.

Shaking: The stochastic component of the method, where a point x' is randomly generated from $N_k(x)$ in order to avoid cycling.

Local Search: The local search heuristic is applied to x' until a local optimum (x'') is achieved.

Move and Neighborhood Change: If the local optimum x'' found during the search is better than the incumbent best solution x, then x'' is accepted and replaces x, setting $k = 1$. Otherwise, the algorithm proceeds to the next neighborhood structure $k = k + 1$ (if $k > k_{max}$, then $k = 1$).

Several VNS variants have been developed since it first appeared to solve many combinatorial optimization problems. One of the best known variants is the variable neighborhood descent (VND) method, which is a deterministic version of VNS where several neighborhood structures in sequence within the Local Search phase are searched, but no Shaking step is performed. Furthermore, VND can replace the Local Search phase in VNS giving origin to General VNS (GVNS). When local search is costly in computational terms, these methods can suffer from efficiency problems. The Reduced VNS (RVNS) is useful in such cases, as it is a pure stochastic method where random points are generated from N_k and the incumbent solution is updated in case of an improvement. Naturally, this variant reduces the effectiveness of the search. VNS design and consequently its efficiency and effectiveness are closely related to the selection of neighborhoods and their order. VNS conducts the search through different neighborhoods usually in increasing distances, evaluated by some metric (or quasi-metric). In 0-1 mixed integer problems like ours, the distance between two solutions can be based on the Hamming distance (Δ_H) that states the number of elementary changes in 0-1 variables to turn one solution into another. Most applications rank the neighborhoods in increasing order of their complexity, which usually corresponds to a bigger Hamming distance. Moreover, the use of nested neighborhoods, i.e., $N_1(x) \subset N_2(x) \subset \ldots \subset N_{k_{max}}(x)$, is often a common choice. Still, the understanding of the problem structure can be crucial to a suitable choice of neighborhoods and their sequence. VNS also represents a trade-off between intensification of the search (Local Search) and diversification (Shaking), which is important to balance.

5.2.1. Solution representation

We use an incomplete representation of the solution, considering the different permutations of the binary family setup variables Y_{fmt}, which are controlled by our algorithm. Given a fixed set of binary values (setup pattern Y'_{fmt}), the remaining problem can be solved optimally as a linear program (LP). The neighborhood structures are induced from the changes in the setup pattern.

5.2.2. Initial solution

Finding a feasible solution for the LT3P is difficult, specially in tight capacity scenarios. In our case, it implies selecting a setup pattern, which can be translated into a plan that verifies demand and capacity constraints. We overcome this problem by introducing artificial decision variables defined as the initial stock of product i at plant j (I^0_{ij}). This initial stock is heavily penalized in the objective function using h^0_{ij}. Doing so, we allow any setup pattern to be feasible, which is also important during the execution of the algorithm. Under these conditions, the term $\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{P}} h^0_{ij}\cdot I^0_{ij}$ becomes a measure of the infeasibility. Even though solutions with empty setups or with setups for every family in each period are now feasible, we want to test the impact of the initial solution on the search efficacy and effectiveness. For that purpose we have developed three procedures to define an initial setup pattern.

1. *LotForLot* is inspired in a lot-for-lot policy. The procedure works period-by-period and plant-by-plant identifying the total gross requirements for a certain product family $R_{fjt} = \sum_{i\in\mathcal{N}^{fam}_f} d_{ijt}$. If $R_{fjt} > 0$, a setup will be triggered on machine $m \in M_j$ having the highest throughput rate p_{fmt}.
2. The second and third procedures are both inspired by the work of Nascimento et al. [23]. Ignoring capacity constraints and/or inter-plant transfers, the problem can be compared to F uncapacitated lot sizing problems on parallel machines, which are solvable through the optimal algorithm of Sung [26].
   a) *Uncap* works plant by plant and a minimum production schedule is found for each family satisfying the demand for the incumbent plant (j) having as potential sources the filling lines belonging to that same plant ($\mathcal{M}_j$), thus only ignoring capacity constraints.
   b) *UncapNoTransf* attempts to find a production schedule for each family satisfying the demand for all plants having as potential sources the set of available filling lines ($\mathcal{M}$), therefore ignoring both capacity constraints and transfer costs.

Next, we describe the general procedure for *UncapNoTransf* as it is a generalization of *Uncap*. For each family f let ϕ_{fmut} be the production cost of family on production line m in period u to meet requirements D_{fut} from periods u to t for all plants:

$$D_{fut} = \sum_{i\in\mathcal{N}^{fam}_f}\sum_{j\in\mathcal{P}}\sum_{s=u}^{t} d_{ijs},$$

$$\phi_{fmut} = c_{fmu} + v'_{fmt}\cdot D_{fut} + \sum_{s=u+1}^{t} h'_{fmt}\cdot D_{fst},$$

where v'_{fmt} and h'_{fmt} correspond to weighted average production and holding costs of the family as follows:

$$v'_{fmt} = \frac{\sum_{i\in\mathcal{N}^{fam}_f} v_{imt}\cdot d_{ijt}}{\sum_{i\in\mathcal{N}_f} d_{ijt}}, \qquad h'_{fmt} = \frac{\sum_{i\in\mathcal{N}^{fam}_f} h_{ijt}\cdot\sum_{u=t+1}^{T} d_{ijt}}{\sum_{i\in\mathcal{N}_f}\sum_{u=t+1}^{T} d_{ijt}}.$$

Moreover, let φ_{ft} be the minimum production cost from period 1 up to period t ($\varphi_{f0} = 0$). Quantity φ_{ft} can be obtained recursively using:

$$\varphi_{ft} = min\left\{\varphi_{f,u-1} + \phi_{fmut}\right\} \quad u \in 1 \ldots t,\ t \in \mathcal{T}.$$

A dynamic programming forward recursion algorithm has been used to solve each subproblem and thus fixing the setup pattern. The assumptions made during this procedure allow us to have a rough approximation of a possible interesting setup pattern.

5.2.3. Neighborhood structures

Neighbors of an incumbent solution x are obtained by slightly changing the setup pattern and solving afterwards the resulting LP. The set of all possible minor changes, also called moves, constitute the neighborhood $N(x)$. We have defined three different type of moves:

a) *insertion*(f, m, t) consists in changing the setup state of family f on machine m at period t from 0 to 1, therefore neighborhood $N_I(x)$ include all possible changes of the variables Y'_{fmt} from 0 to 1,

b) *remove*(f, m, t) is the inverse move of *insertion*, thus $N_R(x)$ are the potential changes of variables Y'_{fmt} from 1 to 0,

c) *transfer*(f, m_o, t_o, m_d, t_d) reallocates a production lot by means of moving the setup of family f from its origin (machine m_o at period t_o) to a new destination (machine m_d at period t_d). The neighborhood $N_T(x)$ corresponds to all possible moves where $Y'_{fm_ot_o} = 1$ and $Y'_{fm_dt_d} = 0$.

All neighbor solutions of $N_I(x)$ and $N_R(x)$ will have a $\Delta_H = 1$, while to neighbor solutions of $N_T(x)$ have $\Delta_H = 2$. The size of the neighborhood of $N_T(x)$ can be controlled by setting limits to t_d, i.e. $t_d = [t_o - \delta_b, t_o + \delta_f]$, when δ_b and δ_f control the backward and forward searching range, respectively. Note that when $t_o \neq t_d$ and only in this case m_o may equal to m_d.

These three types of neighborhoods try to explore different ideas. The *insertion* move attempts to find a new family allocation such that the additional setup cost incurred is shorter than the savings resulting from production, holding, transfer and overtime costs. On the other hand, its inverse *remove* tries the opposite allowing an increase of the other costs through the mitigation of setup costs. Moves resulting from *transfer* are more difficult to interpret. When changing a setup within the same machine we attempt to introduce or eliminate inventory, whether we try backward or forward movements, and possibly decrease overtime costs. Moving setups to other machines in the same period tries to save setup costs that may benefit potential transfers. Moving to other machines in different periods can cause or eliminate inventory, transfers and overtime and eventually reduce setup costs. Nevertheless, when evaluating a neighbor, the LP mathematical model is optimized over the entire planning horizon, therefore performing a change in the setup pattern can have multiple effects on the production, inventory, transfer and overtime decisions, and the above description is myopic in those cases. In fact, this constitutes the reason for the partial solution representation, since heuristically determining production, inventory, transfer and overtime quantities may wrongly reject interesting setup patterns. The price to pay for such decisions is a more costly local search in terms of computational times.

5.2.4. Algorithm design

In theory, the larger the neighborhood, the better is the quality of locally optimal solutions, and the greater the accuracy of the final solution obtained. Standard VNS examines the entire neighborhood during local search. For large problem instances it is impractical to search the neighborhood explicily as it can be too time consuming. In practice, strategic/tactical decisions can be taken in a relatively wide time window, but if one wants to test different scenarios varying data inputs the available time for response is substantially shorter. It is required to partially search the neighborhood in an efficient manner.

To speed-up the algorithm, the evaluation of each neighbor can incorporate rules to quickly identify expensive neighbors and save time in the LP optimization. Since moves are performed based on a known solution, plenty of information is available. In addition, changes in the setup pattern are usually rather small. Solving the LP from scratch can be very time-consuming, thus the previous best found solution constitutes the initial basis in the new LP and then it is just re-optimized. To early discard expensive neighbors, let f^t_{best} denote the best solution found to date and f^s_{best} the respective setup pattern cost. The remaining costs related to production, inventory and transfers calculated by the LP are expressed as $f^o_{best} = f^t_{best} - f^s_{best}$. After generating a new neighbor, $f^s_{neighbor}$ can be easily computed based on f^s_{best}, therefore we can reject a neighbor whose $f^s_{neighbor} > f^t_{best}$ without making any iteration in the LP. Besides,

$f^t_{best} - f^s_{neighbor}$ is the maximum value that $f^o_{neighbor}$ can take before being refused. If the LP is solved using a dual method at each iteration, the dual solution corresponds to a lower bound on $f^o_{neighbor}$, and therefore we can stop the method as soon as it exceeds $f^t_{best} - f^s_{neighbor}$, potentially saving precious computational time. If the method does not stop the LP optimization in course, it means that we have found a new best solution. This technique has already been explored by Meyr [27].

Although these rules can save valuable computational time, for large size instances the result of exhaustively exploring the entire neighborhood still remains unsatisfactory. The ultimate goal is to somehow explore only a portion of each large neighborhood and still find the local optima, or at least find an improvement move, if such a move exists. Traditional techniques to improve efficiency of VNS such as the aforementioned RVNS often compromise efficacy, especially because neighbors are selected randomly. Hung et al. [28] propose in the context of Tabu Search the usage of ranking heuristics based on the information provided by the LP to prune the search of the neighborhood. Their techniques were able to reduce running time through one of two strategies: explore a portion of the ranked neighbors according to the heuristics or to evaluate sequentially the ranked neighbors until an improvement is found. They have developed heuristics to rank neighbors that are obtained either by *insertion* or *remove* moves.

RVNS can be a solution to explore large neighborhoods due to its smaller CPU effort vital in the case study. Still, the randomness of the Shaking Phase can lead the algorithm to randomly suggest expensive neighbors too often, despite that the dual reoptimization process may perform an early rejection. Inspired by the work [28], we have designed new rules to improve the standard RVNS. The idea is to associate a probability to each neighbor according to its potential cost savings. For that purpose we make use of the information available after solving the LP. Let us define β_{mt} and π_{imt} as the shadow prices (dual variables) of constraints (7) and (4), respectively. Additionally, we define SL_{mt} as the surplus of capacity on machine m in period t in the current best solution. An $insertion(f, m, t)$ move can be evaluated through the criteria presented in Algorithm 1. Initially the potential improvement of an $insertion(f, m, t)$ move is the cost of the extra setup that has to be performed (line 1). Then the maximum production quantity of family f is determined considering the surplus of capacity on machine m in period t (line 2). The procedure then iterates through the products belonging to family f ($\mathcal{N}^{fam}_f$) selecting the one with maximum value of π_{imt} (line 5). Let Θ be the set of products selected previously. The potential improvement is increased by the term $\pi_{i_{min}mt} \cdot a$, where a may equal the total demand of the selected product at the plant of machine in the incumbent period (line 7) or the maximum of the remaining surplus of capacity (line 11). In both situations, the remaining surplus of capacity, and Θ are updated (lines 8-9, 12-13). The algorithm loops until no remaining surplus is available or all products belonging to the family have been selected ($\Theta = \mathcal{N}^{fam}_f$). Move $remove(f, m, t)$ is evaluated according to Algorithm 2. Its initial potential improvement is the saving coming from removing the existent setup (line 1). The potential improvement is then updated (line 3) using β_{mt} over the total production quantity of family f determined in line 2. Evaluating

Algorithm 1 Potential improvement of an $insert(f, m, t)$ move

```
 1: Potential improvement: $Imp_{fmt} := -c_{fmt}$
 2: Maximum family production: $MaxProd = \frac{SL_{mt}}{p_{fmt}}$
 3: Current produced products: $\Theta \leftarrow \emptyset$
 4: while $MaxProd > 0$ and $\Theta \subset \mathcal{N}^{fam}_f$ do
 5:     $i_{min} = argmax\{\pi_{imt} \mid i \in \mathcal{N}^{fam}_f \setminus \Theta\}$
 6:     if $MaxProd > d_{i_{min} j_m t}$ then
 7:         $Imp_{fmt} = Imp_{fmt} + \pi_{i_{min}mt} \cdot d_{i_{min},j_m,t}$
 8:         $MaxProd = MaxProd - d_{i_{min} j_m t}$
 9:         $\Theta \leftarrow \Theta \cup \{i_{min}\}$
10:     else
11:         $Imp_{fmt} = Imp_{fmt} + \pi_{i_{min}mt} \cdot MaxProd$
12:         $MaxProd = 0$
13:         $\Theta \leftarrow \Theta \cup \{i_{min}\}$
14:     end if
15: end while
16: return $Imp_{fmt}$
```

Algorithm 2 Potential improvement of a *remove*(f, m, t) move

```
Potential improvement: $Imp_{fmt} := c_{fmt}$
Actual family production: $X^f_{mt} = \sum_{i \in \mathcal{N}_f^{fam}} X_{imt}$
$Imp_{fmt} := Imp_{fmt} + \beta_{mt} \cdot \frac{X^f_{mt}}{p_{fmt}}$
return $Imp_{fmt}$
```

a *transfer*(f, m_o, t_o, m_d, t_d) move is hard because it introduces more changes in the model and therefore the available information is less reliable, yet it can be seen as a combination of an *insertion*(f, m_d, t_d) and a *remove*(f, m_o, t_o), thus summing both potential improvements. Shadow prices can be seen as the marginal utility of the resources. Move *insertion*(f, m, t) explores the marginal utility of an additional setup of family f on machine m in period t assuming that it remains valid for the maximum between the surplus of capacity and the total demand of the family. The same principle is behind the evaluation of *remove*(f, m, t), but this time making use of the marginal utility of freeing” capacity. The potential improvements are only estimations of the real improvement on the objective function, thus we should not restrict too much the search based on them. After calculations are made, the candidate neighbors $r \in N_k(x)$ are sorted according to the potential improvement (Imp_{fmt}). Let $\sigma(r)$ be the rank of neighbor r. The probability $\mu(r)$ of choosing a candidate neighbor is given by:

$$\mu(r) = \frac{bias(r)}{\sum_{r' \in N_k(x)} bias(r')}$$

where $bias(r)$ is called the bias function. Pure RVNS makes use of a random bias, i.e. $bias(r) = 1$. Since our idea is to prioritize candidates at the top of the list, any one of the following bias functions can be used: linear bias $bias(r) = 1/\sigma(r)$, log bias $bias(r) = log^{-1}(\sigma(r) + 1)$ and exponential bias $bias(r) = e^{-\sigma(r)}$. Exponential bias is the most extreme case were mostly the top candidates are chosen, liner bias is less extreme than exponential and log bias is the least differentiator function. We call this enhancement to standard RVNS as Adaptive Reduced Variable Neighborhood Search (ARVNS).

6. Production plans validation and comparison

In this section we present computational experiments divided in two sections. First we validate our solution procedures on a randomly generated set of small to medium sized instances (*Tests I*). Afterwards, we use the algorithm with the best performance to solve two real-world instances based on the annual production budget of the case study for the years of 2010 and 2011 (*Tests II*). All heuristics were implemented in C++, compiled using Microsoft Visual Studio 2008 and run on an Intel Core i7 Q720 1.60 GHz processing unit with 6 GB of random access memory, using a single core. IBM ILOG Cplex 12.1 was used both as mixed integer and liner programming solver and was limited to one thread to have a fair comparison.

6.1. Tests I

This set of tests is designed to validate the proposed algorithm and prove its superiority against other variants reported in the literature. Nevertheless, the features from the case study instances are kept, such as the absence of setup times and production costs, and the use of overtime. All parameters with the exception of demand are considered to be time independent, for example $p_{fmt} = p_{fm}, \forall t \in \mathcal{T}$. This applies only for the generated instances since our heuristics can manage time dependent parameters. Input parameters of each problem instance were generated based on the uniform distribution. The ranges used for the parameters are given in Table 1.
Available capacity of all machines of the same plant is calculated according to:

$$cap_{mt} = \sum_{m \in \mathcal{M}_{j_m}} \frac{\sum_{i \in \mathcal{N}_f^{fam}} d_{ij_m t}}{p_{fmt}} \cdot (\alpha|\mathcal{M}_{j_m}|)^{-1},$$

with $\alpha = 1.25$. The maximum amount of both types (mo^I_{mt}, mo^{II}_{mt}) of overtime is set to 10% of the available capacity. Test classes are defined by the quintuplet (F, N, P, M, T). The number of families F and the number of plant P are

Table 1: Parameter ranges

d_{ijt}	U[40,180]	p_{fmt}	U[1,5]
h_{ijt}	U[0.2,0.4]	co^{I}_{mt}	U[1,2]
r_{ijkt}	U[0.2,0.4]	co^{II}_{mt}	U[3,6]
c_{fmt}	U[50,950]		

always less or equal to the number of products N and the number of machines M, respectively. The process to assign product to families and machines to plant is the same. For example, if 5 products have to be assigned to 3 families, the first 3 are assigned each one to a different family and the remaining 2 are randomly allocated to a family. Tests were conducted using $F = \{5, 10, 15\}$, $N = \{10, 20, 30\}$, $P = \{3\}$, $M = \{4, 6\}$ and $T = \{6, 9, 12\}$ and for each combination 10 different instances were generated, corresponding to a total of 180 instances.

We have run each test instance using Cplex 12.1 on the mathematical formulation LT3P presented in Section 5 with a maximum running time of 600s. Thus, at the end of each run we potentially have an upper bound (the current best integer solution found by the branch-and-cut algorithm) and a lower bound also provided by the same algorithm. The mean gap obtained through Cplex 12.1 is our solution evaluation metric. The percentage *Gap* to the best known lower bound is then computed as:

$$Gap = \frac{z_h - z_{lb}}{z_{lb}} \cdot 100,$$

where z_h is the solution obtained by the method under evaluation ans z_{lb} is the best lower bound known provided by Cplex 12.1. All instances were feasible without considering initial inventory. Furthermore, for some problems the optimal solution was found. Table 2 reports the number of optimal solutions (out of 10) found in each test class by Cplex 12.1. Naturally, as the number of famlies, machines and periods increase, the number of intances solved until optimality decrease sharply. We then tested our solution approach for two variants: RVNS ans ARVNS. RVNS random bias function, while ARVNS makes use of a linear bias function. Neighborhoods were ordered according to *insertion*, *transfer* and *remove*, as it was proved during pre-testing to be the most promising sequence. The maximum number of successive iterations without improving was used as stopping criterion and set to 1000. Both methods were run for the three different types of initial solutions (*LotForLot*, *Uncap* and *UncapNoTransf*). Five runs were executed for each configuration: initial solution and solution approach variant.

Table 2: Number of optimal solutions found by Cplex 12.1 for the different test classes

		T		
F	*N*	6	9	12
		P = 3, *M* = 4		
5	10	9	4	1
10	20	4	1	0
15	30	0	0	0
		P = 3, *M* = 6		
5	10	8	3	2
10	20	4	0	0
15	30	3	0	0

Tables 3 and 4 report the average solution gap and Tables 5 and 6 present the average running times for the three methods under evaluation in the different test classes. The performance of our solution approach clearly depends on the initial solution. *LotForLot* yields the overall best mean gap and *Uncap* generally outperforms *UncapNoTransf*, in particular when using ARVNS. Running times increase as the problem size increases, specially when using *LotForLot* as initial solution. Regarding solution quality, for small sized instances exact methods have, as expected, the best mean gaps. Nevertheless, for medium size instances, such as test class (15,30,3,4,12) and (15,30,3,6,12) our solution

Table 3: Results for the average Gap (%) for the different test classes with $P = 3$ and $M = 4$ for the three methods under evaluation. Best average gaps are in boldface.

			$T = 6$			$T = 9$			$T = 12$		
Initial Solution	F	N	*RVNS*	*ARNVS*	*Cplex*	*RVNS*	*ARNVS*	*Cplex*	*RVNS*	*ARNVS*	*Cplex*
LotForLot	5	10	4.09	2.83	**0.10**	7.08	5.47	**2.90**	9.29	7.55	**4.81**
	10	20	4.94	3.36	**1.70**	7.58	5.90	**5.31**	11.97	**9.96**	10.49
	15	30	7.13	5.19	**3.08**	8.54	6.66	**6.41**	10.09	**8.31**	10.52
	Mean		*5.39*	*3.79*	**1.63**	*7.73*	*6.01*	**4.87**	*10.45*	*8.61*	**8.60**
Uncap	5	10	4.26	3.29	**0.10**	7.45	6.70	**2.90**	9.51	8.86	**4.81**
	10	20	5.04	3.49	**1.70**	7.38	5.95	**5.31**	11.56	**10.25**	10.49
	15	30	7.36	5.56	**3.08**	8.29	6.56	**6.41**	9.62	**8.36**	10.52
	Mean		*5.55*	*4.11*	**1.63**	*7.71*	*6.40*	**4.87**	*10.23*	*9.16*	**8.60**
UncapNoTransf	5	10	4.94	4.68	**0.10**	8.03	6.47	**2.90**	10.03	8.47	**4.81**
	10	20	5.88	4.01	**1.70**	8.15	6.41	**5.31**	11.92	10.54	**10.49**
	15	30	7.86	5.59	**3.08**	8.49	7.22	**6.41**	9.64	**8.72**	10.52
	Mean		*6.23*	*4.76*	**1.63**	*8.22*	*6.70*	**4.87**	*10.53*	*9.24*	**8.60**

approaches are more competitive than exact methods both in solution quality and running time. Thus, we conclude that our solution approaches are more competitive for large-scale problems. Exact methods are less attractive as problems grow in size, specially as the number of periods increases. Enhanced neighbor selection present in ARVNS proved to be profitable. ARVNS is always superior to standard RVNS and generally takes less running time. Iterations in ARVNS take longer to perform because both evaluation of potential improvement and sorting of neighbors have to be done to properly calculate $\mu(r)$. Yet, this type of neighbor selection allows the search to converge faster and to a better local optimum. Hence, ARVNS is a very promising tool to effectively explore large neighborhoods and therefore to be used in real-world problems.

6.2. Tests II

The second set of instances are based on real data from the case study. There are two instances corresponding to the annual production budget of 2010 and 2011, respectively. Both only consider the planning of beer and soft drinks plants. The instance related to the year of 2009 comprises data from 3 plants, each one having a set of 1 to 5 filling lines, totalizing 10 filling lines. Sales budget forecasts are available over the next 12 months for a total of 125 products, which can be aggregate into 62 different product families. Technological restrictions limit family assignments to filling lines, nevertheless more than 100 family-filling line allocations are possible in each time period. In the year of 2011, again the total number of plants is 3, but the total number of filling lines increased to a total of 14, ranging between 4 to 5 in each plant. The total number of products also increased to over 160, which are now aggregated into 68 different product families. As a result, the number of possible family-filling line allocations are now over 120. For both instances, data relative to family throughput rates, product holding and transfer costs, and overtime costs are estimations made by the company based on previous years.

The benchmark was conducted on the following PB scenarios:

U1: Company's PB transformed into a solution of our optimization model (LT3P), thus allowing to compute the objective function.

U2: Fixed family allocation (setup pattern Y'_{fmt}) from the company's PB solving the subsequent problem optimally through the LT3P LP model.

U3: PB obtained using ARVNS with a maximum number of iterations without improvement of 1000, *LotForLot* as the initial solution strategy and the same neighborhood order from the previous tests (configuration with best overall performance in *Tests I*). The final solution is the best among 5 runs.

Table 4: Results for the average Gap (%) for the different test classes with $P = 3$ and $M = 6$ for the three methods under evaluation. Best average gaps are in boldface.

			$T = 6$			$T = 9$			$T = 12$		
Initial Solution	*F*	*N*	*RVNS*	*ARNVS*	*Cplex*	*RVNS*	*ARNVS*	*Cplex*	*RVNS*	*ARNVS*	*Cplex*
LotForLot	5	10	4.05	1.70	**0.23**	5.71	3.63	**2.14**	8.75	6.66	**5.54**
	10	20	3.95	2.22	**1.22**	6.55	4.12	**3.63**	7.84	**6.07**	6.44
	15	30	3.30	1.89	**1.17**	5.18	3.65	**3.45**	6.83	**4.89**	6.69
	Mean		*3.77*	*1.94*	**0.88**	*5.81*	*3.80*	**3.07**	*7.81*	**5.87**	*6.22*
Uncap	5	10	3.79	2.35	**0.23**	5.29	3.95	**2.14**	8.69	7.07	**5.54**
	10	20	4.50	2.42	**1.22**	6.04	4.38	**3.63**	8.04	**6.38**	6.44
	15	30	3.56	2.13	**1.17**	4.79	3.82	**3.45**	6.37	**5.05**	6.69
	Mean		*3.95*	*2.30*	**0.88**	*5.37*	*4.05*	**3.07**	*7.70*	**6.17**	*6.22*
UncapNoTransf	5	10	3.19	3.17	**0.23**	5.57	4.37	**2.14**	8.54	7.29	**5.54**
	10	20	3.75	2.73	**1.22**	6.23	4.80	**3.63**	8.17	6.49	**6.44**
	15	30	3.54	1.96	**1.17**	5.37	3.86	**3.45**	6.41	**4.87**	6.69
	Mean		*3.49*	*2.62*	**0.88**	*5.73*	*4.34*	**3.07**	*7.70*	**6.22**	**6.22**

Nowadays, PB is done using spreadsheets, but is mainly a manual process. Previous experience in PB creation constitutes the pillar of the planning process as it follows implicitly cost based decisions. A comparison with company planning is not always straightforward since a manual planning solution does not always strictly obeys all restrictions. Scenario *U2* tries to reduce this gap by creating the best possible scenario with the current family allocation and also shows the drawback of pre-defined family-filling line allocatios. Table 7 reports results for the three scenarios for the two real-world insatances. All costs are measured in terms of monetary units (m. u.). Not surprisingly, optimizing production, inventory, transfer and overtime decisions, for the company's family allocation (scenario *U2*) has a strong impact. Manually performing these decisions will likely lead to sub-optimality. Creating PB with ARVNS by relaxing family allocations can further improve these results. Scenario *U2* achieves a total cost saving of 24% and 34% in 2010 and 2011, respectively. A large portion of cost savings comes for overtime reduction, an interesting result since the company is obsessed with the holding costs. Our heuristic obtained the best plans, yielding 35% cost reduction in 2010 and 39% in 2011. Plans clearly show the existing trade-offs among costs. For example, in 2010 PB for both *U2* and *U3* have higher holding costs than in company's plan and still improved the solution, while in 2011 transfer costs suggested by *U3* increase as this can lessen overtime. Moreover, scenario *U3* always reduces the total number of setups. Excluding setup costs, all other costs are relatively easy to quantify and, therefore, very accurate. The direct potential cost savings from inventory, transfer and overtime costs in both years is quite impressive, representing 36% and 40%, respectively for 2010 and 2011. Savings in 2011 are bigger because we are considering more products and filling lines. Figure 1 and 2 help to understand the obtained results. For the instance in 2011, scenario *U1* comprehensibly shows the difficulty of dealing with peak demand that occurs during Summer. Inventory is built up early in the year to face seasonality, in addition during the summer season both transfers and overtime requirements increase. The two optimized scenarios can deal with this effect more smoothly. Scenario *U3* in the last part of the year uses notably less overtime, but it uses more inventory and transfers quantities compared to scenario *U2* as a means to achieve a more cost-efficient plan. The average running times of our heuristic were 100s and 180s for the 2010 and 2011 instances, respectively. This confirms the ability of the heuristic of effectively solving large problems.

7. Discussion

This paper is motivated by a real production planning problem in the beverage industry. The goal is to produce a long-term plan assigning and scheduling product family production lots in a multi-plant environment, having each plant one or more production lines. Total setup, inventory, transfers and overtime costs constitute the objective to minimize. We first formulate the problem as a mixed integer program. Based on our mathematical formulation we

Table 5: Average running times for the different test classes with $P = 3$ and $M = 4$ for the three methods under evaluation. Best average running times are in boldface.

			$T = 6$			$T = 9$			$T = 12$		
Initial Solution	F	N	*RVNS*	*ARNVS*	*Cplex*	*RVNS*	*ARNVS*	*Cplex*	*RVNS*	*ARNVS*	*Cplex*
LotForLot	5	10	**1.15**	1.24	119.45	**2.15**	2.60	444.68	**3.35**	3.79	544.17
	10	20	**3.18**	3.29	411.41	**5.87**	7.42	574.44	**9.83**	11.24	601.67
	15	30	**6.23**	6.95	601.84	**12.73**	13.27	601.83	**18.64**	23.51	600.76
	Mean		**3.52**	*3.83*	*377.57*	**6.91**	*7.76*	*540.31*	**10.61**	*12.85*	*582.20*
Uncap	5	10	**1.03**	1.17	119.45	**2.01**	2.26	444.68	**2.91**	3.37	544.17
	10	20	2.92	**2.70**	411.41	**5.59**	6.15	574.44	**9.37**	9.47	601.67
	15	30	**5.73**	6.08	601.84	11.37	**10.64**	601.83	18.03	**17.71**	600.76
	Mean		**3.23**	*3.32*	*377.57*	**6.32**	*6.35*	*540.31*	**10.11**	*10.19*	*582.20*
UncapNoTransf	5	10	1.31	**1.14**	119.45	**2.03**	2.23	444.68	3.20	**3.13**	544.17
	10	20	3.23	**2.39**	411.41	6.06	**5.55**	574.44	10.00	**8.46**	601.67
	15	30	6.33	**5.28**	601.84	12.58	**10.09**	601.83	17.09	**15.95**	600.76
	Mean		*3.62*	**2.94**	*377.57*	*6.89*	**5.96**	*540.31*	*10.10*	**9.18**	*582.20*

have developed a heuristic suitable for the large size instances present in industrial applications. A partial solution representation of product family setup decisions (binary variables) was used and the production, inventory, transfer and overtime quantities (continuous variables) are determined by solving a linear program. We make use of the information provided by sensitivity analysis of the linear program to guide the local search. Neighbors are evaluated and sorted according to their potential improvement and neighbor selection is done according to this rank. We are dealing with very large problem instances in the case study and as tests proved, the heuristic is able to efficiently explore wide solution spaces. Another important feature of our heuristic is its flexibility, due to the partial solution representation. One can add different requirements to the model, for example production costs, family setup times, minimum family production batches, that were not considered in this application, without having to change the procedure. These adjustments are only needed in the mixed integer problem, which constitutes the base of the linear program.

Tests on real-world instances validated our approach, as we are able to notably improve current company practice. Therefore, this study can constitute the basis for the implementation of a decision support tool for long-term production planning within the company. The test of different planning scenarios and the introduction of a rolling horizon procedure for long-term planning can be features of the planning tool with great capability of enhancing current planning decisions.

It would be interesting to test the new heuristic in other large scale problems to further validate its potential.

References

[1] B. Karimi, S. F. Ghomi, J. Wilson, The capacitated lot sizing problem: a review of models and algorithms, Omega 31 (2003) 365–378(14).

[2] R. Jans, Z. Degraeve, Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches, European Journal of Operational Research 177 (2006) 1855–1875.

[3] P. Hansen, N. Mladenovi, J. Moreno Prez, Variable neighbourhood search: methods and applications, 4OR: A Quarterly Journal of Operations Research 6 (2008) 319–360.

[4] N. A. Eskin, Biochemistry of Foods, Second Edition., Academic Press, Inc., London, 1990.

[5] N. L. Kent, A. D. Evers, Technology of Cereals. An Introduction for Students of Food Science and Agriculture, Fourth Edition., Elsevier Science, Ltd., Kingston, Oxford., 1994.

[6] L. K. Kourtis, I. S. Arvanitoyannis, Implementation of HACCP system to the alcoholic beverages industry, Food Reviews International 17 (1) (2001) 1–44.

[7] C. Matthews, Soft Drinks — Production, in: B. Caballero (Ed.), Encyclopedia of Food Sciences and Nutrition, Academic Press, Oxford, 5352 – 5358, 2003.

[8] R. M. Cooke, A. Bosma, F. Hrte, A practical model of Heineken's bottle filling line with dependent failures, European Journal of Operational Research 164 (2) (2005) 491 – 504.

Table 6: Average running times for the different test classes with $P = 3$ and $M = 6$ for the three methods under evaluation. Best average running times are in boldface.

			$T = 6$			$T = 9$			$T = 12$		
Initial Solution	F	N	RVNS	ARNVS	Cplex	RVNS	ARNVS	Cplex	RVNS	ARNVS	Cplex
LotForLot	5	10	**1.67**	1.68	242.60	**2.94**	3.24	430.34	**4.97**	5.04	485.52
	10	20	4.73	**4.32**	423.43	8.82	**8.64**	601.40	15.05	**14.53**	602.02
	15	30	8.41	**7.86**	460.57	17.35	**16.96**	600.30	27.05	**26.89**	601.59
	Mean		4.94	**4.62**	375.53	9.70	**9.61**	544.01	15.69	**15.49**	563.04
Uncap	5	10	1.58	**1.48**	242.60	**2.44**	2.50	430.34	4.46	**4.09**	485.52
	10	20	3.66	**3.44**	423.43	7.91	**6.84**	601.40	12.11	**10.15**	602.02
	15	30	7.16	**5.84**	460.57	15.36	**11.69**	600.30	23.79	**19.03**	601.59
	Mean		4.13	**3.59**	375.53	8.57	**7.01**	544.01	13.45	**11.09**	563.04
UncapNoTransf	5	10	2.06	**1.42**	242.60	2.95	**2.40**	430.34	5.05	**4.19**	485.52
	10	20	4.54	**3.56**	423.43	8.14	**6.30**	601.40	12.24	**9.09**	602.02
	15	30	7.56	**5.52**	460.57	13.71	**11.03**	600.30	21.07	**15.33**	601.59
	Mean		4.72	**3.50**	375.53	8.27	**6.58**	544.01	12.79	**9.54**	563.04

[9] P. H. Tsarouhas, I. S. Arvanitoyannis, Assessment of operation management for beer packaging line based on field failure data: A case study, Journal of Food Engineering 98 (1) (2010) 51 – 59.

[10] D. Ferreira, R. Morabito, S. Rangel, Solution approaches for the soft drink integrated production lot sizing and scheduling problem, European Journal of Operational Research 196 (2009) 697–706.

[11] S. A. Araujo, M. N. Arenales, A. R. Clark, Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups, Journal of Heuristics 13 (2007) 337–358.

[12] B. Almada-Lobo, D. Klabjan, M. A. Carravilla, J. F. Oliveira, Single machine multi-product capacitated lot sizing with sequence-dependent setups, International Journal of Production Research 45 (20) (2007) 4873–4894.

[13] A. Clark, R. Morabito, E. Toso, Production setup-sequencing and lot-sizing at an animal nutrition plant through ATSP subtour elimination and patching, Journal of Scheduling 13 (2010) 111–121.

[14] P. Chandra, M. L. Fisher, Coordination of production and distribution planning, European Journal of Operational Research 72 (3) (1994) 503 – 517.

[15] J. K. Jolayemi, F. O. Olorunniwo, A deterministic model for planning production quantities in a multi-plant, multi-warehouse environment with extensible capacities, International Journal of Production Economics 87 (2) (2004) 99 – 113.

[16] C. H. Martin, D. C. Dent, J. C. Eckhart, Integrated Production, Distribution, and Inventory Planning at Libbey-Owens-Ford, Interfaces 23 (3) (1993) 68–78.

[17] C. H. Timpe, J. Kallrath, Optimal planning in large multi-site production networks, European Journal of Operational Research 126 (2) (2000) 422 – 435.

[18] M. Sambasivan, S. Yahya, A Lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers, Computers and Operations Research 32 (2005) 537–555.

[19] B. Almada-Lobo, J. F. Oliveira, M. A. Carravilla, Production planning and scheduling in the glass container industry: A VNS approach, International Journal of Production Economics 114 (1) (2008) 363–375.

[20] C. Dhaenens-Flipo, G. Finke, An Integrated Model for an Industrial ProductionDistribution Problem, IIE Transactions 33 (2001) 705–715.

[21] P. Kaminsky, D. Simchi-Levi, Production and Distribution Lot Sizing in a Two Stage Supply Chain, IIE Transactions 35 (11) (2004) 1065 – 1075.

[22] M. Sambasivan, C. P. Schimidt, A heuristic procedure for solving multi-plant, multi-item, multi-period capacited lot-sizing problems, Asia Pacific Journal of Operational Research 19 (2002) 87–105.

[23] M. C. Nascimento, M. G. Resende, F. M. Toledo, GRASP heuristic with path-relinking for the multi-plant capacitated lot sizing problem, European Journal of Operational Research 200 (3) (2010) 747 – 754.

[24] G. Bitran, H. Yanasse, Computational complexity of the capacitated lot size problem, Management Science 28 (1992) 1174–1186.

[25] N. Mladenovic, P. Hansen, Variable neighborhood search, Computers & Operations Research 24 (11) (1997) 1097 – 1100.

[26] C. S. Sung, A single-product parallel-facilities production-planning model, International Journal of Systems Science 17 (7) (1986) 983 – 989.

[27] H. Meyr, Simultaneous lotsizing and scheduling on parallel machines, European Journal of Operational Research 139 (2) (2002) 277 – 292.

[28] Y.-F. Hung, C.-P. Chen, C.-C. Shih, M.-H. Hung, Using tabu search with ranking candidate list to solve production planning problems with setups, Computers & Industrial Engineering 45 (4) (2003) 615 – 634.

Table 7: Results of the different scenarios for the two real-world instances

	U1	**U2**	**Savings (U2)**	**U3**	**Savings (U3)**
		2010			
Objective Function	1.962.720	1.509.115	453.605 (23%)	1.275.769	686.951 (35%)
Holding Costs	108.589	185.748	-77.159 (-71%)	134.426	-25.837 (-24%)
Tranfer Costs	204.129	162.247	41.883 (21%)	148.772	55.357 (27%)
Setup Costs	42.550	42.550	0 (0%)	41.650	900 (2%)
Total Number of Setups	851	851	0 (0%)	833	18 (2%)
Overtime Costs	1.607.451	1.118.570	488.881 (30%)	950.921	656.530 (41%)
Objective Function (without Setup Costs)	1.920.170	1.466.565	453.605 (24%)	1.234.119	686.051 (36%)
		2011			
Objective Function	3.259.777	2.163.237	1.096.540 (34%)	1.976.865	1.282.912 (39%)
Holding Costs	450.926	171.429	279.496 (62%)	163.060	287.866 (64%)
Tranfer Costs	317.965	312.157	5.808 (2%)	365.503	-47.538 (-15%)
Setup Costs	48.350	48.350	0 (0%)	46.050	2.300 (5%)
Total Number of Setups	967	967	0 (0%)	921	46 (5%)
Overtime Costs	2.442.536	1.630.334	812.202 (33%)	1.402.252	1.040.283 (43%)
Objective Function (without Setup Costs)	3.211.427	2.114.887	1.096.540 (34%)	1.930.815	1.280.612 (40%)

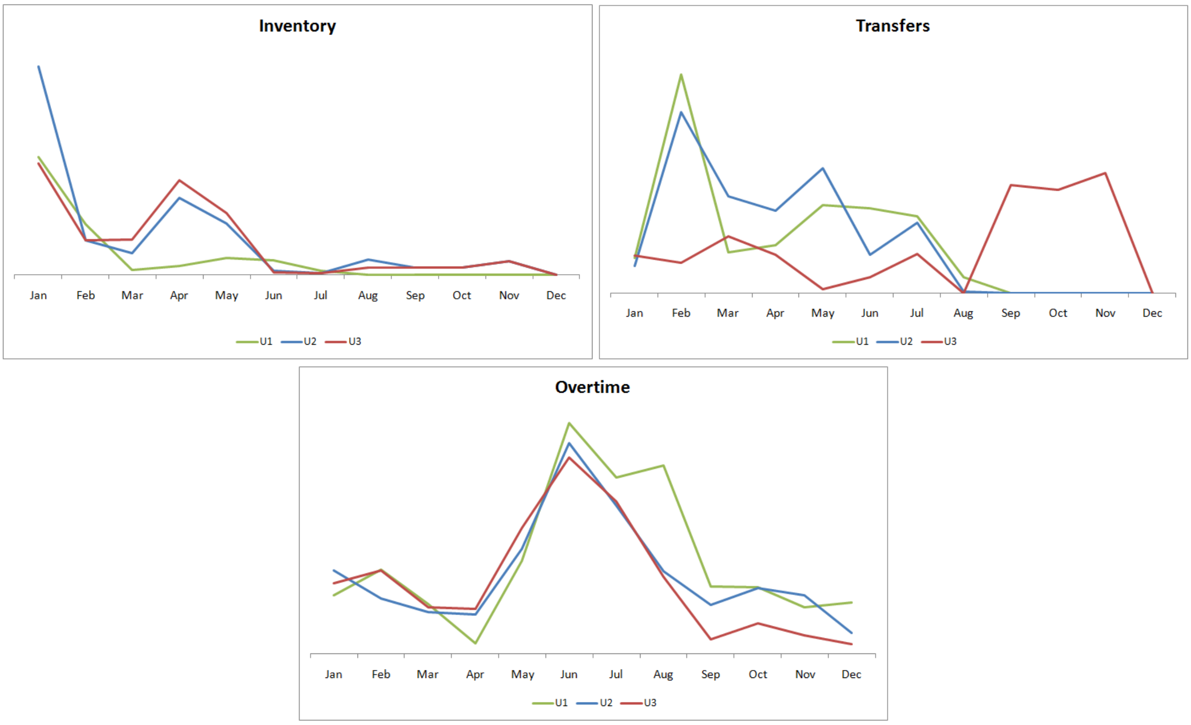

Figure 1: Comparison of inventory, transfer and overtime decisions of the different scenarios for the 2010 PB

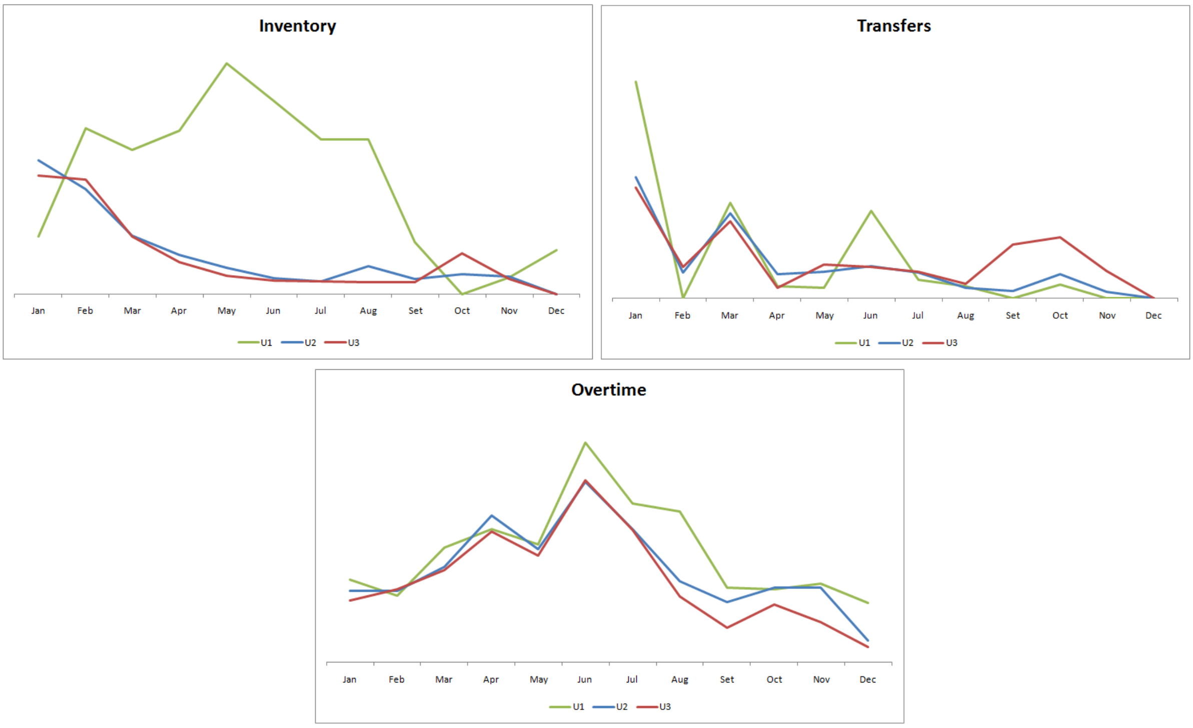

Figure 2: Comparison of inventory, transfer and overtime decisions of the different scenarios for the 2011 PB