Multiple Machine Continuous Setup Lotsizing with Sequence-dependent Setups

Bernardo Almada-Lobo[a*] Diego Klabjan[b] Maria Antónia Carravilla[a,c]
José F. Oliveira[a,c]

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

[b] *Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois*

[c] *Instituto de Engenharia de Sistemas e Computadores do Porto Campus da FEUP, Rua Dr. Roberto Frias 378, Porto 4200-465, Portugal*

Abstract

We address the short-term production planning and scheduling problem coming from the glass container industry. A furnace melts the glass that is distributed to a set of parallel molding machines. Both furnace and machine idleness are not allowed. The resulting multi-machine multi-item continuous setup lotsizing problem with a common resource has sequence-dependent setup times and costs. Production losses are penalized in the objective function since we deal with a capital intensive industry. We present two mixed integer programming formulations for this problem, which are reduced to a network flow type problem. The two formulations are improved by adding valid inequalities that lead to good lower bounds. We rely on a Lagrangian decomposition based heuristic for generating good feasible solutions. We report computational experiments for randomly generated instances and for real-life data on the aforementioned problem, as well as on a discrete lotsizing and scheduling version.

Keywords: continuous setup lotsizing, sequence-dependent setup, integer programming, Lagrangian decomposition, heuristics

1 Introduction

Process industries are capital intensive leading to a strong focus on improving efficiencies and reducing costs to remain competitive. It is imperative that demand is satisfied in the most cost-effective manner. The main operational driver is to maximize the facilities throughput by means of a specialization of processes to decrease downtimes. We deal with the short term production planning problem faced by a glass container manufacturing company. The reader is referred to Almada-Lobo et al. [2008] for an overview of the long-term planning problem. Glass containers are intermediary in nature, and can be considered as almost a commodity. It is a semi-continuous manufacturing process, where a common resource (furnace) produces the glass to be distributed to a set of parallel machines that will form the containers. Significant machine setup times and costs are incurred for switchovers from one product to another. The problem is to find production

*Corresponding author. *E-mail address*: almada.lobo@fe.up.pt; tel: +351225082133

orders that maximize the "good tonnage" produced, while meeting a deterministic demand without backlogging. Thus, one has to minimize the production losses due to machine switchovers and furnace under capacity utilization, as well as holding costs. Additional complicate requirements are taken into account, such as minimum lot-sizes, machine balancing and furnace idleness. The resulting lotsizing and scheduling problem is an extension of the standard continuous setup lotsizing problem (CSLP). The glass container industry works under the make-to-stock, serving dynamic markets. The make-to-stock policy is not only driven by demand seasonality, but also by the requirements to meet customers' needs until the next production run of the desired glass containers. The level of stock of containers held by this industry is of significant importance in considering the economics of manufacturing, and fluctuates with changes in demand. In order to meet peak demand requirements, stock is increased when demand is low. The number of months of demand that are covered from stock ranges from two to three. Although new furnaces cost millions of euros, the strategy of pulling the maximum amount of glass out of the furnace does not come from the need to decrease marginal costs linked to this furnace sunk (fixed) cost, but from the need to be competitive operationally and increase the gross margin (related to variable costs). Moreover, due to economies of scale in natural gas consumption (the main industrial cost of this process) and to other technical constraints, it is imperative that furnaces are run near their capacity. Hence, production losses are not only triggered by setups (as usually considered), but also from not using the maximum furnace capacity.

There is a wide variety of lotsizing and scheduling models, involving different features and assumptions. The models that integrate lotsizing and scheduling are discussed in Drexl and Kimms [1997]. Typically, the planning horizon is divided into a finite number of periods. In large-bucket problems several products, setups may be produced, performed per period, respectively. The capacitated multi-item lotsizing problem (CLSP) is a typical example of a large-bucket model (Almada-Lobo et al. [2007]). On the other hand, in small time bucket models, at most one setup may be executed per period and, therefore, they are applicable for developing short-term production schedules. This is the case for the discrete lotsizing and scheduling problem (DLSP), where a discrete production policy is assumed, in which at most one product is produced at full capacity, known as the "all-or-nothing" assumption (Fleischmann [1994]). Dematta and Guignard [1994a] study the multi-machine DLSP arising at a tile manufacturing company, without considering setup times. In the subsequent work, Dematta and Guignard [1994b] analyze the same problem but incorporate sequence-dependent setup times that reflect production losses. An exact solution method for DLSP with sequence-dependent setup costs and times is proposed in Salomon et al. [1997]. Jans and Degraeve [2004] address several specific industrial extensions of the multi-machine DLSP with sequence-independent setup costs motivated by a real world production planning problem arising at a tire manufacturer. CSLP relaxes the "all-or-nothing" requirement, as production can take any value up to the capacity. Karmarkar and Schrage Karmarkar and Schrage [1985] present the single-machine version of this problem without setup costs and label it the *production cycling problem*. The authors apply a Lagrangian relaxation approach to decouple the problem that provides lower bounds used in a branch-and-bound algorithm. Karmarkar et al. [1987] study the single item version of CSLP, for both uncapacitated and capacitated cases. Sandbothe Sandbothe [1996] tackles single-machine multi-item CSLP with sequence independent setup costs and no setup times with a three-step heuristic. Hindi Hindi [1995] develops a tabu-search procedure to the single-item CSLP with startup costs. Wolsey Wolsey [1997] surveys work that can be used to strengthen the formulations of the single-machine multi-item CSLP with both sequence independent and sequence dependent changeovers. Vanderbeck Vanderbeck [1998] solves the single-machine multi-item CSLP with sequence independent setups using an integer programming column generation algorithm and develops a dynamic programming procedure for the single-item subproblem. Constantino Constantino [2000] derives valid inequalities for the single-machine multi-item CSLP with sequence independent setups and implements a branch-and-cut algorithm. The reader is referred to Jans and Degraeve [2007] for an up-to-date overview of the existing algorithms for solving dynamic lotsizing problems, focusing specially on meta-heuristics.

All the aforementioned manuscripts address the single-machine CSLP. It is well known that solving CSLP is at least as hard as solving the associated DSLP. Vanderbeck [1998] questions whether the decomposition approach is practical for the generalization of CSLP to the case of multiple machines. In our problem, the total amount of the renewable, continuous resource (molten glass) available at any time is limited. Since the production rate of a product on a machine depends on the amount of the continuous resource allotted to it at a time, machines may have to produce below their own capacity. Thus, the production environment at stake does not allow an extension of DSLP and, consequently, we focus on the difficult CSLP. Dematta and Guignard [1995] also deal with multi-machine CSLP, but changeover time (and production losses) are assumed to be negligible. Productivity losses from making too many small batches are usual in lotsizing models. To the best of our knowledge, this is the first work to address the production losses of not using all of a resource, which is critical in some process industries.

We solve a mixed integer programming formulation of an extension of CSLP that appears in short-term glass container production planning and scheduling. We employ a Lagrangian decomposition approach to decouple the problem into more manageable pieces. The Lagrangian relaxation problem is modeled as a network flow type problem. We use the solution of the decomposition to develop a model-based Lagrangian heuristic by means of an efficient subgradient optimization procedure for solving the Lagrangian dual and a simple primal heuristic for yielding feasible solutions. On top of this, we implement valid inequalities that enable us to considerably improve the quality of lower bounds.

The main contributions of our work are as follows. To the best of our knowledge, this is the first work on multi-machine CSLP with sequence dependent setup times and costs and production loss costs. We solve a relevant industrial problem of a major competitive capital intensive industry. A novel Lagrangian relaxation of a proposed formulation is designed in such a way that it results in an easily solvable subproblem. In order to achieve this we relax the original formulation. We stress that a straightforward application of Lagrangian does not produce satisfactory results (these experiments are not shown here). Another major contribution of our work is a set of valid inequalities to improve the quality of the lower bounds. An excellent feature of these inequalities is the fact that their impact increases as the number of products and periods increases. Finally, we validate our approach with both real-life data and random instances. The random instance generator is designed in such a way that some settings reflect the features of CSLP instances, and others reflect those of DLSP instances.

The reminder of the paper is organized as follows. In Section 2 we describe the underlying production planning problem and present a mathematical model of an exact formulation. Section 3 reduces it into a simplified model, which is an extension of CSLP. The same section is also dedicated to a reduction of these models into network flow problems by means of a Lagrangian relaxation of the problem. The overall algorithm underlying the heuristic based on the Lagrangian approach is presented in Section 4. Computational results are given in Section 5.

2 The problem statement and model

The glass container manufacturing process begins with the mixture of raw materials that is transported into the furnace where it is melted at around 1500°C. Since the batch material takes about 24 hours to pass through the melting stage, the furnace capacity is measured in melted tons per day. Natural gas is the energy source used in this process. The glass paste is cut into gobs and distributed by the feeders to a set (typically ranging from 2 to 6) of parallel independent section (IS) glass molding machines that shape the finished product at 600°C. The formed containers are then passed sequentially through conditioning, surface treatment, automatic inspection, and packaging procedures.

There is enough capacity downstream of the molding machines to process all the work coming from upstream. Even if some problems arise at the end of the production line (the packaging area), the conveyor belt has buffer areas to temporarily stock intermediate products, avoiding molding machines to stop. Since the production scheduling is only constrained by the glass production and the containers forming, this process can be considered to be single level; the transformation of the molten glass into a finished product.

Due to high sequence dependent setup times involved in a color changeover, the color of glass melted on each furnace is likely to remain constant in short or medium term. The long-term production planning output schedules color campaigns on the furnaces of several sites and assigns product orders to the furnaces. Given this output, the objective of short-term planning is to assign (and sequence) products to machines on a daily basis within each color campaign (the analysis is conducted furnace by furnace) that ensures the satisfaction of customers' due dates and maximizes the productivity, i.e., minimizes the loss of production due to changeovers and unused capacity of the furnace, as well as holding costs.

2.1 The main requirements

Since only one color of glass is produced at a time in each furnace, machines served by the same furnace always form containers of the same color. Furnaces are operated continuously (except when they are being repaired) and machine lines operate on a 24 hour seven days a week basis. Therefore, there is little flexibility for varying output to match fluctuations in demand. Due to economies of scale in natural gas consumption and to structural constraints, machine idleness is not allowed. This machine balancing constraint forces machines fed by the same furnace to operate the same amount of time. Each machine can only run one product at a time.

Each molding machine has four main characteristics: the number of individual sections (container making units assembled side by side, ranging from 6 to 12), the number of mold cavities per section, i.e., the number of gobs to be formed in parallel (ranging from 1 to 3: in a double-gob machine two gobs are shaped at the same time within a section), the center distance, i.e., the distance between the molds in a double-gob or triple-gob machine (either 41/4 inches, 5 inches, 51/2 inches or 61/4 inches) and the type of the manufacturing process ("blow-blow", "press and-blow", and "narrow-neck press-and-blow" techniques). The first two features determine the maximum throughput of the machine, while the last two restrict the set of products that can be allocated to a machine. Figure 1 schematizes a double-gob machine of eight sections.

The processing time of each product per mold cavity of each machine is constant (in the glass terminology, the cavity rate is referred to as the number of containers formed per minute in a mold cavity). Hence, the production rate of a product on a machine depends on the respective cavity

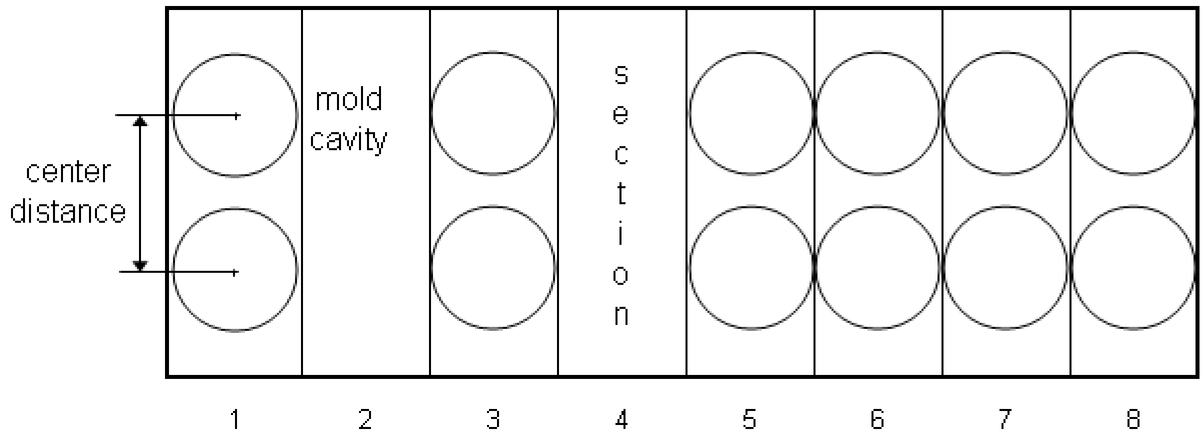

Figure 1: Double-gob IS machine of eight sections

rate and on the total number of active mold cavities (number of active sections $\times$ the number of gobs formed in parallel). One major advantage of IS machines is the possibility to independently stop some sections. Moreover, some flexible machines may operate different gob configurations (e.g., the same machine may run either a double or a triple gob setting).

Due to the semi-continuous nature of this manufacturing process, the daily throughput of the furnace is determined by the daily output of its associated machines. If the product mix on the adjacent machines (that are fed by the same furnace) processing at full speed demands too much from the furnace output (above its daily capacity), it may be necessary to stop some machine sections and/or change (if possible) the number of gobs to be formed in parallel. On the other hand, if the mix of products at a certain day pulls too little molten glass from the furnace (usually happens when the products are lightweight), the natural gas consumption economies of scale are minimal and may lead to prohibitive industrial costs.

During a product changeover on a machine, the furnace keeps feeding the machine, however, the gobs are discarded and melted down again in the furnace. Hence, sequence-dependent setups consume part of the furnace capacity. This operation is performed by specialized workers in the first shift of the day. As a result, the minimum time slot considered by the planner is a day (i.e., a machine can only be assigned a product per day).

In summary, the most important requirements are:

- each product can be carried over to the next time period,
- at most one product can be produced on a machine in any time period,
- active mold cavities are reconfigurable at the end of each time period, but the number must be within a certain range,
- the furnace can be idle,
- a product changeover at a machine uses the capacity of the furnace and therefore there is a corresponding cost, and

- a machine can only be idle at the tail of the planning horizon (they cannot be restarted during the horizon).

2.2 The comprehensive formulation

Throughout the exposition, t denotes time periods, which range from 1 to T, i and j index products, which are labeled from 1 to N, and k denotes machines, which range from 1 to K. In general, we denote by $[M]$ the set $\{1, 2, \ldots, M\}$, and by $\nu(\cdot)$ optimal values of underlying optimization problems.

We are given the following data:

d_{it}	demand for product i at the end of period t (expressed in tons)
$\overline{n}_{ik}$	the maximum number of mold cavities of machine k in which product i can be produced
$\underline{n}_{ik}$	the minimum number of mold cavities of machine k in which product i can be produced
p_{ik}	quantity of product i produced per mold cavity of machine k in a period (tons)
s_{ijk}	setup time of a changeover from product i to product j, $j \neq i$ on machine k (tons)
c_{ijk}	cost incurred to set up machine k from product i to product j, $j \neq i$
h_i	holding cost of carrying one ton of product i from one period to the next
C	melting capacity of the furnace in a period (tons).

As discussed earlier, during changeover, gobs, which are measured in tons, are returned back to the furnace. We call this the setup time even though it is measured in tons. We denote by ω the conversion factor between the idle time of the furnace and the unit of cost (usually the monetary unit).

We use the following decision variables:

$$Y_{it}^k = \begin{cases} 1 & \text{if product } i \text{ is assigned to machine } k \text{ in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$Q_t = \begin{cases} 1 & \text{if the furnace is active in period } t \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{ijt}^k = \begin{cases} 1 & \text{if product } j \text{ is scheduled in period } t \text{ and product } i \text{ in period } (t-1), \\ & \text{both on machine } k \\ 0 & \text{otherwise} \end{cases}$$

N_{it}^k = number of active mold cavities of machine k dedicated to product i in period t

I_{it} = stock of product i at the end of period t (tons)

$\overline{Id}_t$ = idle capacity of the furnace in period t (tons).

We assume that I_{i0} denotes the initial inventory of product i. The short term scheduling problem is modeled as the following MILP formulation, denoted by F_1.

$$\nu(F_1) = \min \sum_{i,j,k,t} c_{ijk} \cdot Z_{ijt}^k + \omega \cdot \sum_t \overline{Id}_t + \sum_{i,t} h_i \cdot I_{it} \tag{1}$$

$$I_{it} + d_{it} - I_{i(t-1)} = \sum_k \left(p_{ik} \cdot N_{it}^k - \sum_j s_{jik} \cdot Z_{jit}^k \right) \qquad i \in [N], t \in [T] \tag{2}$$

$$\sum_{i,k} p_{ik} \cdot N_{it}^k + \overline{Id}_t = C \cdot Q_t \qquad t \in [T] \tag{3}$$

$$N_{it}^k \leq \overline{n}_{ik} \cdot Y_{it}^k \qquad i \in [N], k \in [K], t \in [T] \tag{4}$$

$$N_{it}^k \geq \underline{n}_{ik} \cdot Y_{it}^k \qquad i \in [N], k \in [K], t \in [T] \tag{5}$$

$$\sum_i Y_{it}^k \leq 1 \qquad k \in [K], t \in [T] \tag{6}$$

$$\sum_i Y_{it}^k \geq \sum_i Y_{i(t+1)}^k \qquad k \in [K], t \in [T-1] \tag{7}$$

$$Q_t = \sum_i Y_{it}^k \qquad k \in [K], t \in [T] \tag{8}$$

$$Y_{jt}^k + Y_{i(t-1)}^k \leq Z_{ijt}^k + 1 \qquad i \in [N], j \in [N] \setminus \{i\}, k \in [K], t \in [T] \tag{9}$$

$$(I_{it}, \overline{Id}_t, Q_t) \geq 0, N_{it}^k \text{ integer}, (Y_{it}^k, Z_{ijt}^k) \text{ binary}.$$

The objective function (1) aims at minimizing the sum of sequence dependent changeover, and holding and furnace idleness costs. Idleness is an opportunity cost for not pulling the maximum out of the furnace. Constraints (2) balance the inventory flow for two consecutive periods and together with $I_{it} \geq 0$ ensure that demand is met without backlogging. Note that the parameter p_{ik} ("good" tonnage of product i produced per mold cavity of machine k in a day) is given by

$$p_{ik} = CR_{ik} \cdot w_i \cdot 24 \cdot 60 \cdot \eta_k,$$

where CR_{ik} denotes the cavity rate of product i on machine k, w_i the weight of product i and η_k the efficiency of machine k .

Constraints (3) restrict the furnace melted tonnage per period to its capacity and define its idleness ($\overline{Id}_t$). In constraints (4), Y_{it}^k is forced to be one if a production occurs for product i on machine k in period t and the number of active mold sections (N_{it}^k) is limited by the respective pair machine/product capacity. The technological constraints, such as product i not able to be processed on machine k, are reflected in the parameter $\overline{n}_{ik}$ that would equal to zero in such circumstances. In case of a production, constraints (5) activate a minimum number of mold cavities on a machine. Nonzero $\underline{n}_{ik}$ implies a manager's decision based on an intrinsic restriction from the underlying production process. Constraints (6) prevent a machine from processing simultaneously more than one product. Intermittent machine idleness is not allowed by constraints (7), forcing idle periods to be placed at the end of the planning horizon (after an idle period, the machine remains idle until the end of the planning horizon). Machines fed by the same furnace must be active in the same periods of time, which is ensured by (8). Constraints (9) guarantee the coherency between variables Y_{it}^k and Z_{ijt}^k. Finally, (10) represent the integrality and non-negativity constraints. Note that the integrality condition of Q_t is not necessary.

In addition, the short-term planning process must also respect management rules of the different production sites like, for instance, the changing of a lot on a machine being possible only on working days and on some predefined shifts (since it is undertaken by teams of highly skilled workers), or the number of changes per week limited per facility. The number of available mold equipments may also limit the number of machines on which a product can be allocated simultaneously. Therefore, job splitting may not be allowed. All such restrictions are easy to incorporate in the model by using the existing variables.

3 The solution methodology

Clearly, formulation F_1 is very hard to solve. This model is simplified by relaxing the integrality of N_{it}^k and introducing continuous variables X_{it}^k to capture the approximate quantity (expressed in tons) of product i produced on machine k in period t, and by assuming null initial inventory for every product.

Let $M_{ik} = \overline{n}_{ik} \cdot p_{ik}$ be an upper bound on the quantity of product i to be produced on machine k per time period and let $m_{ik} = \underline{n}_{ik} \cdot p_{ik}$ be a lower bound on the same quantity. The model F_2 reads

$$\nu(F_2) = \min \omega \cdot C \cdot \sum_t Q_t - \omega \cdot \sum_{i,k,t} X_{it}^k + \sum_{i,j,k,t} (c_{ijk} - \omega \cdot s_{ijk}) \cdot Z_{ijt}^k + \sum_{i,t} h_i \cdot \left(\sum_k \sum_{s=1}^{t} X_{is}^k - \sum_{s=1}^{t} d_{is} \right) \tag{10}$$

$$\sum_k \sum_{s=1}^{t} X_{is}^k - \sum_{s=1}^{t} d_{is} \geq 0 \qquad i \in [N], t \in [T] \tag{11}$$

$$\sum_{i,k} X_{it}^k + \sum_{i,j,k} s_{ijk} \cdot Z_{ijt}^k \leq C \cdot Q_t \qquad t \in [T] \tag{12}$$

$$X_{it}^k + \sum_j s_{jik} \cdot Z_{jit}^k \leq M_{ik} \cdot Y_{it}^k \qquad i \in [N], k \in [K], t \in [T] \tag{13}$$

$$X_{it}^k + \sum_j s_{jik} \cdot Z_{jit}^k \geq m_{ik} \cdot Y_{it}^k \qquad i \in [N], k \in [K], t \in [T] \tag{14}$$

$$\text{constraints } (6) - (9)$$

$$(X_{it}^k, Q_t) \geq 0, (Y_{it}^k, Z_{ijt}^k) \in \{0, 1\}. \tag{15}$$

This model is an extension of the standard CSLP, which is computationally NP-hard. Clearly, there is no known polynomial algorithm to check feasibility a priori. We first argue that F_2 is a relaxation of F_1.

Proposition 1. We have $\nu(F_1) \geq \nu(F_2)$.

Proof. Let us define X_{it}^k as

$$X_{it}^k = p_{ik} \cdot N_{it}^k - \sum_j s_{jik} \cdot Z_{jit}^k \qquad i \in [N], k \in [K], t \in [T]. \tag{16}$$

We can remove inventory variables from model F_1 assuming, without loss of generality, null initial inventory level for every product (i.e., $I_{i0} = 0$ for every i). This fact, together with (16), allows us to replace (2) by (11). These constraints state that the cumulative production for item i is at least equal to the cumulative demand up to each period t. In addition, incorporating (16) into (4) and (5) yields constraints (13) and (14). We can argue that constraints (12) hold as follows:

$$C \cdot Q_t = \sum_{i,k} p_{ik} \cdot N_{it}^k + \overline{Id}_t \geq \sum_{i,k} p_{ik} \cdot N_{it}^k = \sum_{i,k} X_{it}^k + \sum_{i,j,k} s_{ijk} \cdot Z_{ijt}^k.$$

We note that both furnace capacity and setup times are expressed in tons. Thus, variable $\overline{Id}_t$ in F_1 represents the unused tonnage of the furnace in an active period t. By using (3) and considering (16), we derive the objective function (10). Clearly, from (16), X's only take integer values in F_1 since N and Z are integer and binary variables, respectively. On the other hand, X's are continuous variables in F_2. If S and R are the feasible regions of polytopes F_1 and F_2, respectively, then we have just established that $S \subseteq R$. This clearly shows that $\nu(F_2) \leq \nu(F_1)$. $\square$

3.1 A network reformulation

We now reduce F_2 to a network-flow type problem. We first observe that the following constraints are an alternative formulation to constraints (7) and (9):

$$\sum_j Z_{jit}^k \geq \sum_j Z_{ij(t+1)}^k \qquad i \in [N], k \in [K], t \in [T-1]. \tag{17}$$

Contrarily to formulation F_2, here it is mandatory that Z_{iit}^k equals to one when machine k is set up for product i from period $t-1$ to period t (a phantom setup) and $s_{iik} = 0$. Thus we define $s_{iik} = c_{iik} = 0$ and we also use Z_{iit}^k. Constraints (17) ensure a balanced network flow of each machine configuration state and carry the setup state of the machine into the next period (as done by constraints (9)). They impose an output setup performed in period $t+1$ for product i to be preceded by an input setup in period t for the same product. Moreover, these constraints force idle periods to be placed at the end of the planning horizon. In case production stops in period $t-1$, period t contains no setups (idle period), i.e. $\sum_{i,j} Z_{ijt}^k = 0$ and, by constraints (17), $\sum_{i,j} Z_{ijs}^k = 0$ for every $s > t$. As a result, constraints (17) also replace constraints (7). We also observe that

$$Y_{it}^k = \sum_j Z_{jit}^k, \tag{18}$$

i.e., we conclude that product i is only assigned to machine k in period t if an input setup is performed for product i. Variables Y_{it}^k can be eliminated from model F_2. After dividing objective function (10) by ω, we reduce model F_2 to the following network formulation P:

$$\nu(P) = \min C \cdot \sum_t Q_t - \sum_{i,k,t} X_{it}^k + \sum_{i,j,k,t} \left(\frac{c_{ijk}}{\omega} - s_{ijk}\right) \cdot Z_{ijt}^k + \sum_{i,t} \frac{h_i}{w} \cdot \left(\sum_k \sum_{s=1}^t X_{is}^k - \sum_{s=1}^t d_{is}\right)$$

$$
\begin{aligned}
&\text{constraints (11) and (12)} \\
X_{it}^k + \sum_j s_{jik} \cdot Z_{jit}^k \leq & M_{ik} \cdot \sum_j Z_{jit}^k && i \in [N], t \in [T], k \in [K] && (19)\\
X_{it}^k + \sum_j s_{jik} \cdot Z_{jit}^k \geq & m_{ik} \cdot \sum_j Z_{jit}^k && i \in [N], t \in [T], k \in [K] && (20)\\
\sum_{i,j} Z_{ijt}^k \leq & 1 && t \in [T], k \in [K] && (21)\\
\sum_j Z_{jit}^k \geq & \sum_j Z_{ij(t+1)}^k && i \in [N], t \in [T-1], k \in [K] && (22)\\
Q_t = & \sum_{i,j} Z_{ijt}^k && t \in [T], k \in [K] && (23)
\end{aligned}
$$

$$
X_{it}^k \geq 0, (Z_{ijt}^k, Q_t) \in \{0,1\}. \tag{24}
$$

By the aforementioned arguments this is an equivalent formulation to F_2. The only constraints that link the parallel machines together are (11) and (12). If we dualize these constraints by multiplying them by non-negative vectors of dual multipliers λ_{it} and π_t, respectively, the Lagrangian problem PLD is stated as

$$
\begin{aligned}
\nu(PLD) = \min_{X,Q,Z} \quad & C \cdot \sum_t Q_t - \sum_{i,k,t} X_{it}^k + \sum_{i,j,k,t} \left(\frac{c_{ijk}}{\omega} - s_{ijk} \right) \cdot Z_{ijt}^k + \sum_{i,t} \frac{h_i}{w} \cdot \left(\sum_k \sum_{s=1}^t X_{is}^k - \sum_{s=1}^t d_{is} \right) \\
& + \sum_{i,t} \lambda_{it} \cdot \left(\sum_{s=1}^t d_{is} - \sum_k \sum_{s=1}^t X_{is}^k \right) + \sum_t \pi_t \left(\sum_{i,k} X_{it}^k + \sum_{i,j,k} s_{ijk} \cdot Z_{ijt}^k - C \cdot Q_t \right)
\end{aligned}
$$

$$
\text{subject to} \qquad (19) - (24).
$$

Reorganizing the terms of the objective function yields

$$
\begin{aligned}
\nu(PLD) = \min \sum_{i,k,t} & \left(\pi_t - 1 - \sum_{s=t}^T (\lambda_{is} - \frac{h_i}{w}) \right) \cdot X_{it}^k + C \cdot \sum_t (1 - \pi_t) \cdot Q_t \\
& + \sum_{i,j,k,t} \left(s_{ijk} \cdot (\pi_t - 1) + \frac{c_{ijk}}{\omega} \right) \cdot Z_{ijt}^k + \sum_{i,t} d_{it} \cdot \sum_{s=t}^T (\lambda_{is} - \frac{h_i}{w}).
\end{aligned}
$$

Due to the machine balancing constraints, any feasible solution satisfies $Q_1 = \ldots = Q_l = 1$ and $Q_{l+1} = \ldots = Q_T = 0$, for a particular l. Given a fixed l, $\nu(PLD)$ decouples by machine into a set of single machine models PLD_k^l as follows:

$$
\nu(PLD) = \min_l \left[C \cdot \sum_{t=1}^l (1 - \pi_t) + \sum_k \nu(PLD_k^l) \right] + \sum_{i,t} d_{it} \cdot \sum_{s=t}^T (\lambda_{is} - \frac{h_i}{w}),
$$

where

$$\nu(PLD_k^l) = \min_{X,Z} \sum_{i,t} \left(\pi_t - 1 - \sum_{s=t}^{T} (\lambda_{is} - \frac{h_i}{w}) \right) \cdot X_{it}^k + \sum_{i,j,k,t} \left(s_{ijk} \cdot (\pi_t - 1) + \frac{c_{ijk}}{\omega} \right) \cdot Z_{ijt}^k,$$

$$\begin{aligned} \text{subject to } & (19), (20), (22) \\ & \sum_{i,j} Z_{ijt}^k = \begin{cases} 1, & t \in [l], k \in [K] \\ 0, & t \in [T] \setminus [l], k \in [K] \end{cases} \qquad (25) \\ & X_{it}^k \geq 0, Z_{ijt}^k \text{ binary.} \end{aligned}$$

If we consider the production of product i on machine k in period t, then $\sum_j Z_{jit}^k = 1$, $\sum_j Z_{ji't}^k = 0$ for $i' \in [N] \setminus \{i\}$ and $Q_t = 1$. Given this condition, we can calculate X_{it}^k by solving the following problem:

$$\begin{aligned} \theta(i, j, k, t, \lambda, \pi) = \max_X X_{it}^k \cdot & \left(1 - \pi_t + \sum_{s=t}^{T} (\lambda_{is} - \frac{h_i}{w}) \right) \\ \text{subject to } X_{it}^k & \leq M_{ik} - s_{jik} \\ X_{it}^k & \geq m_{ik} - s_{jik} \\ X_{it}^k & \geq 0. \end{aligned}$$

Note that if $1 - \pi_t + \sum_{s=t}^{T} (\lambda_{is} - h_i/w) \geq 0$, then $X_{it}^k = [M_{ik} - s_{jik}]^+$, otherwise $X_{it}^k = [m_{ik} - s_{jik}]^+$. Clearly, the amount of product i to be produced in period t results from a tradeoff between multipliers λ_{it} and π_t, i.e., a tradeoff between an eventual stockout of product i in period t (constraint (11) is violated) and an excess of furnace production in period t (violation of constraint (12)). Given multipliers λ's and π's, in each Lagrangian iteration we solve N^2TK problems of the form $\theta(i, j, k, t, \lambda, \pi)$ to determine a priori the production amounts of each assigned product.

We have established that PLD_k^l reduces to

$$\nu(PLD_k^l) \;=\; \min_Z \sum_{i,j,t} \left(-\theta(i, j, k, t, \lambda, \pi) + s_{jik} \cdot (\pi_t - 1) + \frac{c_{jik}}{\omega} \right) \cdot Z_{jit}^k$$

$$\begin{aligned} \text{subject to} \quad & (22), (25) \\ & Z_{ijt}^k \text{ binary.} \end{aligned}$$

Note that (25) depend on l.

Consider machine k and given multipliers λ's and π's let

$$w_{ijt}^k = \begin{cases} -\theta(i, j, k, t, \lambda, \pi) + s_{jik} \cdot (\pi_t - 1) + \frac{c_{jik}}{\omega} & \text{if } i \in [N] \\ \infty & \text{otherwise.} \end{cases}$$

Next we show how to efficiently solve PLD_k^l. Let us define an acyclic graph G_l with $V = [N] \times [l], A = [N] \times [N] \times [l]$ for each machine k, where each node (i, t) represents the product i to be produced in period t on machine k, and each arc $a : (i, t) \to (j, t + 1)$ corresponds to the setup from product i to product j at the beginning of period $t + 1$ on machine k. Each of these arcs has weight w_{ijt}^k. Next we define a new network $G_l^0 = (V^0, A^0)$ by adding source node $(s, 0)$ and arcs $(s, 0) \to (j, 1)$ for every j with weight w_{sj1}^k, where s is the product produced in period 0, and a target node $(v, l + 1)$ and arcs $(i, l) \to (v, l + 1)$ for every i with zero weight. This network is illustrated in Figure 2.

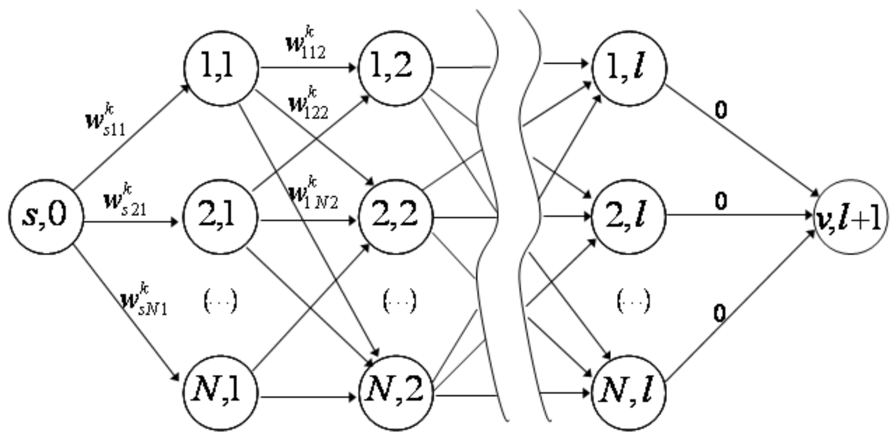

Figure 2: Network representation of problem PLD_k^l.

We solve PLD_k^l by finding a shortest path on the acyclic graph from node $(s, 0)$ to node $(v, l+1)$. We refer the reader to the $O(m)$ reaching algorithm described in Ahuja et al. [1993] for solving the shortest path problem in acyclic networks. Here m is the number of arcs in the network, which in our case is $O(lN^2)$. The shortest path problem is a special version of the minimum cost flow problem with zero lower flow bound on each unit capacity arc, which aims to send 1 unit of flow from node s to node v along the path with the minimum cost. It is well known that in a feasible and bounded minimum cost flow problem with node supplies and arc flow bounds that are integer, there exists an optimal integral flow vector (see, e.g., Bertsekas [1998]).

Problem PLD_k^l is not directly a minimum cost flow problem, but nevertheless integrality of Z is automatic, as shown in the following theorem.

Theorem 1. *Problem PLD_k^l exhibits the integrality property, i.e., its LP relaxation exhibits an optimal integral solution.*

Proof. If we sum (22) over all i we obtain that $\sum_{i,j} Z_{jit}^k \geq \sum_{i,j} Z_{ij(t+1)}^k$. From (25) it follows that both sides are 1 if $t + 1 \leq l$. We conclude that in (22) we have equalities, which model that flow in must equal flow out. Now it is clear that PLD_k^l is equivalent to the shortest path problem on the network in Figure 2. □

Let us also consider another relaxation where we relax (11), (12) and (23). The resulting Lagrangian relaxation is denoted by $PLDW$ (not presented here). Theorem 1 and the well known result from Geoffrion [1974] show that $\nu(PLDW) = \nu(PLP)$, where the LP relaxation of P is denoted by PLP. We conclude that $\nu(PLP) = \nu(PLDW) \leq \nu(PLD) \leq \nu(P)$.

We implemented an algorithm based on $PLDW$, but the results were not satisfactory and are not presented.

3.2 Valid Inequalities

In this section we present four classes of valid inequalities to tighten the network formulation P.

From (11), (18) and $X_{it}^k \leq M_{ik} \cdot Y_{it}^k$, which follows from (13), we obtain that

$$\sum_{j,k} \sum_{s=1}^{t} M_{ik} \cdot Z_{jis}^k \geq \sum_{s=1}^{t} d_{is} \qquad i \in [N], t \in [T] \tag{26}$$

are valid inequalities. Clearly then every valid inequality for this knapsack type problem is valid for P. There are many known valid inequalities.

The second class of inequalities exploit the fact that once a furnace is idle, it remains inactive until the end of the time horizon.

Proposition 2. The following set of inequalities

$$\sum_{j} Z_{jit}^k \leq \sum_{j} Z_{ij(t+1)}^k + Q_t - Q_{t+1} \qquad i \in [N], t \in [T-1], k \in [K] \tag{27}$$

are valid for P.

Proof. Let Q, Z be a feasible solution to P and let us fix i, t, k. Clearly, $0 \leq \sum_{\bar{t}}(Q_{\bar{t}-1} - Q_{\bar{t}}) \leq 1$ since $Q_{\bar{t}-1} \geq Q_{\bar{t}}$ for every $\bar{t}$.

In case $Q_{s-1} - Q_s = 1$ for an s, then $Q_1 = Q_2 = \ldots = Q_{s-1} = 1$ and $Q_s = \ldots = Q_T = 0$. It follows that $Q_{s-1} - Q_s = 1$ and $Q_{\bar{t}-1} - Q_{\bar{t}} = 0$ for every $\bar{t} \neq s$. If $\sum_j Z_{jit}^k = 0$, then product i is not produced in period t and (27) is valid since the right-hand side is nonnegative. If $\sum_j Z_{jit}^k = 1$ (product i produced in period t on machine k), then clearly $Q_t = 1$. In this case we distinguish two further cases: if $Q_{t+1} = 0$ (i.e., the production stops in period t), then it follows from (23) that $\sum_{i,j} Z_{ij(t+1)}^k = 0$ and therefore (27) holds; if $Q_{t+1} = 1$, then constraints (22) and (23) imply $\sum_j Z_{ij(t+1)}^k = 1$, validating (27).

If such an s does not exist, then $Q_1 = Q_2 = \ldots = Q_T = 0$ and hence $\sum_j Z_{jit}^k = \sum_j Z_{ijt}^k = 0$ for every i and t. We conclude that (27) clearly holds. $\square$

We note that if $Q_t = Q_{t+1}$, then (27) together with (22) impose $\sum_j Z_{jit}^k = \sum_j Z_{ij(t+1)}^k$ and, therefore, there is balanced flow through each node.

The third set of inequalities is based on those presented in Pochet and Wolsey [2006].

Proposition 3. The inequalities

$$\sum_{j} Z_{ji(t-1)}^k + \sum_{j:j \neq i} Z_{jit}^k \leq 1 - \sum_{j:j \neq i} Z_{jjt}^k \quad i \in [N], t \in [T] \setminus \{1\}, k \in [K]. \tag{28}$$

are valid for P.

Proof. Let us consider a Z feasible to P and we fix i, t, k. For ease of notation we introduce $W_{jt}^k = \sum_{u:u\neq j} Z_{ujt}^k$, which equal to 1 if start-up occurs for product j on machine k in period t, and 0 otherwise. We can now rewrite (28) as

$$\sum_j Z_{ji(t-1)}^k + W_{it}^k \leq 1 - \sum_{h:h\neq i} \left(\sum_j Z_{jht}^k - W_{ht}^k \right) \quad i \in [N], t \in [T] \setminus \{1\}, k \in [K]. \tag{29}$$

To show (29), we consider three cases.

1) Let us first consider $\sum_j Z_{ji(t-1)}^k + W_{it}^k = 0$. Then $\sum_{h:h\neq i} \left(\sum_j Z_{jht}^k - W_{ht}^k \right) = \sum_{h:h\neq i} Z_{hht}^k \leq 1$, where we used (21). This establishes (29).

2) Let now $W_{it}^k = 1$. It implies that product i is not produced in period $t-1$ and $\sum_j Z_{ji(t-1)}^k = 0$. Hence the left-hand side of (29) equals 1. Clearly then $W_{ht}^k = 0$ for every $h, h \neq i$. We also have $\sum_j Z_{jht}^k = 0$ for every $h, h \neq i$. We conclude that $\sum_{h:h\neq i} \left(\sum_j Z_{jht}^k - W_{ht}^k \right) = 0$ and thus the right-hand side in (29) equals 1.

3) Let us now assume that $\sum_j Z_{ji(t-1)}^k = 1$. Then product i is produced in period $t-1$ and hence no start-up for product i occurs in period t. It means that $W_{it}^k = 0$ and the left-hand side of (29) is thus 1. If product i is produced also in period t, then clearly $W_{ht}^k = \sum_j Z_{jht}^k = 0$ for every $h, h \neq i$. If product i is not produced in period t, then any setup for product $h \neq i$ in period t must be accompanied by a start-up, i.e., $\sum_{j,h:h\neq i} Z_{jht}^k = \sum_{h:h\neq i} W_{ht}^k$. We conclude that the right-hand side of (29) is 1.

From the three cases it follows that $\sum_j Z_{ji(t-1)}^k + W_{it}^k \leq 1$. Thus case 1 covers the case $\sum_j Z_{ji(t-1)}^k + W_{it}^k = 0$, while the remaining two cases cover $\sum_j Z_{ji(t-1)}^k + W_{it}^k = 1$. This argument shows that the three cases cover all possibilities. □

For the remaining class of inequalities, let $M_i^* = \max_k M_{ik}$ and we define

$$\delta_t = \left(N - \left\lfloor \frac{(t-1) \cdot K}{\min_i \lceil \frac{\sum_s d_{is}}{M_i^*} \rceil} \right\rfloor \right)^+.$$

Proposition 4. The inequalities

$$\delta_t \leq \sum_{\substack{i,j,k \\ j\neq i}} \sum_{s=t}^{T} Z_{jis}^k \quad t \in [T] \tag{30}$$

are valid for P.

Proof. In a feasible solution to P any product has a minimum number of production time slots given by $\min_i \lceil \frac{\sum_s d_{is}}{M_i^*} \rceil$. At the end of period $t-1$, we might have faced the entire production requirements of at most $\left\lfloor \frac{(t-1)\cdot K}{\min_i \lceil \frac{\sum_s d_{is}}{M_i^*} \rceil} \right\rfloor$ products. Thus δ_t is a lower bound on the number of start-ups that must be performed in periods $t, t+1, \ldots, T$. It is easy to see that in period t the minimum number of start-ups for the remaining planning horizon is given by δ_t. Thus (30) are valid for P. □

4 The Lagrangian Heuristic

In this section we exploit the problem structure and build a heuristic method to obtain feasible solutions based on Lagrangian relaxation.

The success of any Lagrangian approach depends upon three features: the tightness of the lower bound provided by the sub-problem, the ability to produce good primal feasible solutions, and the efficiency in solving the Lagrangian dual. A successful technique to solve the Lagrangian dual is the well-known subgradient optimization algorithm (see, e.g., Held et al. [1974]). Let $PLD(\lambda^m, \pi^m)$ denote the dual function at iteration m. In order to compose a search direction to update the multipliers, let us define two subgradients of $PLD(\lambda^m, \pi^m)$ based on

$$\zeta_t^m = \sum_{i,k} (\overline{X}_{it}^k)^m + \sum_{i,j,k} s_{ijk} \cdot (\overline{Z}_{ijt}^k)^m - C \cdot (\overline{Q}_t)^m \quad t \in [T] \text{ and} \tag{31}$$

$$\Omega_{it}^m = \sum_{s=1}^{t} d_{is} - \sum_k \sum_{s=1}^{t} (\overline{X}_{is}^k)^m \quad i \in [N], t \in [T], \tag{32}$$

where $\overline{X}, \overline{Z}, \overline{Q}$ denote an optimal solution to $PLD(\lambda^m, \pi^m)$. Lagrangian multipliers are updated according to the recursions

$$\pi_t^{m+1} = [\pi_t^m + \varphi^m \cdot \zeta_t^m]^+ \text{ and } \lambda_{it}^{m+1} = [\lambda_{it}^m + \tau^m \cdot \Omega_{it}^m]^+,$$

where φ^m and τ^m are the step sizes in iteration m and $[\cdot]^+$ ensures their projection onto the nonnegative orthant. A suitable step size is crucial for fast convergence of the subgradient method. Let μ^m be a parameter satisfying $0 < \mu^m \leq 2$, UB an upper value on the dual function PLD and $\| \cdot \|$ the Euclidean norm. We use the following stepsizes:

$$\varphi^m = \mu^m \cdot \frac{UB - \nu\,(PLD(\lambda^m, \pi^m))}{\| \zeta^m \|^2} \text{ and } \tau^m = \mu^m \cdot \frac{UB - \nu\,(PLD(\lambda^m, \pi^m))}{\| \Omega^m \|^2}.$$

The main advantage of this relaxation is that it yields a simple sub-problem, since solving each PLD_k is equivalent to finding a shortest path on an acyclic graph for each l. This property enables a large number of subgradient iterations in order to solve the Lagrangian dual.

In our implementation, we chose $\mu^1 = 1.1$ as an initial value, and if no improvement of the lower bound is obtained in 10 successive iterations, we set $\mu^m = 0.5 \cdot \mu^{m-1}$ and reset μ^m back to 1.1 whenever we get an improved solution. This algorithm is stopped when the gap between the known upper bound and the Lagrangian bound is less than 0.15%, or after 30 iterations without a lower bound improvement.

An important component of the Lagrangian solution is deriving feasible solutions to F_2. A solution to F_2 is characterized by the setup pattern Z_{ijt}^k and the production quantities X_{it}^k that are assigned according to this setup pattern. We fix the setup variables of F_2 with the values from the underlying PLD solution, and solve the remaining linear program to optimality to obtain the production amounts.

5 Computational Results

Computational experiments were performed on an ASUS personal computer with 3.0 GHz CPU and 2GB of random access memory. CPLEX 10.1 from ILOG was used as the mixed integer programming solver and the Lagrangian approach was coded in OPL version 5.1 also from ILOG.

Test problems were generated using the following generator. The number of machines K equals to three, the number of products N were 5, 10, and 15 and the number of periods T were 30 and 45. External demand occurs at the end of each fifth period for all products (time between orders equals 5 time periods) and it is drawn from uniform distribution $U(400, 1000)$. Let Cut denote the (approximate) furnace capacity utilization. We consider medium capacitated problems ($Cut = 0.6$) and high capacitated problems ($Cut = 0.8$). Furnace daily capacity C is given by $\sum_{i,t} d_{it}/(T \cdot Cut)$. Setup times ($s_{ijk}$) were generated based on $U[0.10 \cdot \frac{C}{K}, 0.20 \cdot \frac{C}{K}]$ for every $i \neq j$, and are zero for $i = j$. If we assume that the cost c_{ijk} captures the wasted tonnage, then they are measured in tons and $c_{ijk} = s_{ijk}$ and the weighting factor ω equals to 1. Since the glass containers are almost considered as a commodity, holding costs (h_i) are the same for all products and they are 0.20.

Regarding the upper and lower bounds, two different settings are analyzed. The first setting S_1 considers upper bounds on the production of product i on machine k (M_{ik}) obtained from a normal distribution with an expected value of C/K and a coefficient of variation of 0.1, while the respective lower bound (m_{ik}) is derived based on the expression $U[0.4, 0.8] \cdot M_{ik}$. This setting reflects the unique properties of the glass container industry. The second setting S_2 considers $M_{ik} = m_{ik} = C/K$ and reflects the features of DLSP instances (as the discrete production policy takes place), and is used for comparison purposes. We note that the parameters of this generator were based on our case study data. For instance, the coefficients of the uniform distribution that randomly generates m_{ik} followed from the fact that some machines can stop at most 20% of their sections, while other can stop almost 60% of them.

For each quadruplet N, T, Cut and S (with $K = 3$), ten different instances were generated. In addition, for $K = 4$, $Cut = 0.8$ and S_1, ten instances were generated for each pair (N, T). Hereafter we present for each instance type the average of the values obtained across the 10 instances. Figure 3 illustrates a feasible solution for an instance of type $K = 3/S_1/Cut = 0.8/N = 10/T = 30$. Here, the furnace is active throughout the entire planning horizon, and it is heavily loaded in the last third of the plan. Capacity is clearly tight, therefore the furnace (and the associated machines) could not be stopped beforehand.

An instance of type $K = 3, S_1, N = 15, T = 45$ produces an IP in formulation F_2 with $6,330$ rows, $32,418$ columns and $194,895$ nonzeros, while $K = 4, S_1, N = 15, T = 30$ contains $5,372$ rows, $26,562$ columns and $156,619$ nonzeros. These are fairly large IPs that are very hard to solve to optimality in reasonable time.

We first compare the LP relaxation of models F_2 and F_2 strengthened by the four sets of valid inequalities described in Section 3. The optimal value of the LP relaxation of F_2 is denoted by $\nu(F_{2LP})$ and after adding the valid inequalities by $\nu(F^\star_{2LP})$. Tables 1 and 2 present the gaps $\frac{\nu(F^\star_{2LP}) - \nu(F_{2LP})}{\nu(F_{2LP})}$ for settings S_1 and S_2, respectively. The impact of the valid inequalities is larger

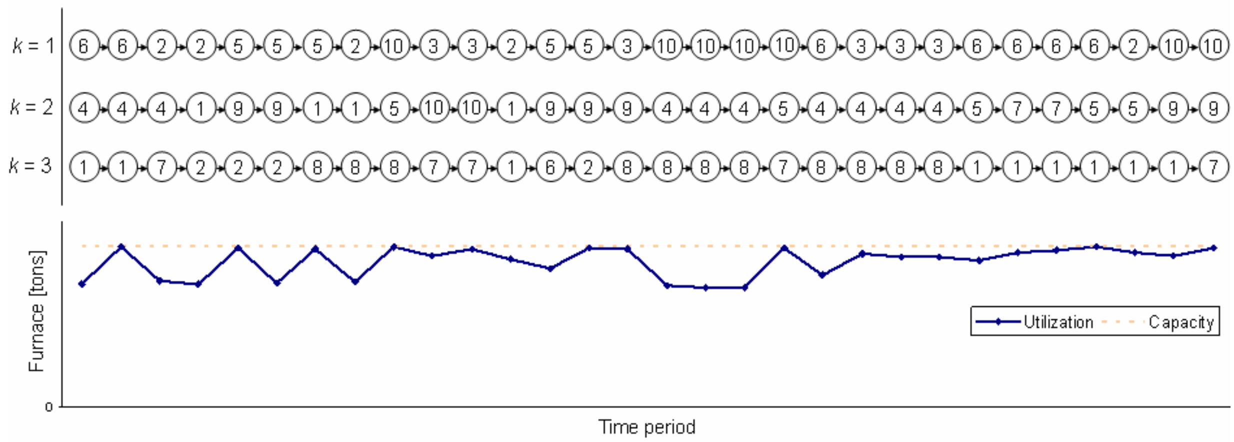

Figure 3: Solution example for the instance $K = 3/S_1/Cut = 0.8/N = 10/T = 30$

for instances with medium capacity utilization than for instances with high capacity utilization. Moreover, it is clear that this impact is more pronounced for setting S_2 than for S_1, and tends to increase as the number of products and periods increase.

Table 1: Comparison (%) of $F^{\star}_{2LP}$ and F_{2LP} for setting S_1, $K = 3$

	$Cut = 0.6$		$Cut = 0.8$	
N	$T = 30$	$T = 45$	$T = 30$	$T = 45$
5	2.2%	2.6%	0.1%	0.0%
10	2.9%	3.4%	0.4%	0.2%
15	2.5%	3.6%	1.0%	0.3%

Table 2: Comparison (%) of $F^{\star}_{2LP}$ and F_{2LP} for setting S_2, $K = 3$

	$Cut = 0.6$		$Cut = 0.8$	
N	$T = 30$	$T = 45$	$T = 30$	$T = 45$
5	6.3%	8.8%	2.3%	3.0%
10	7.1%	9.4%	2.4%	2.9%
15	8.3%	11.1%	2.5%	2.9%

Tables 3 and 4 display the minimum, average, and the maximum gap of the heuristic solution from the lower bound for settings S_1 and S_2, respectively. The heuristic finds a feasible solution for all problem instances, excepting the two most tightly capacitated instance: $S_1, Cut = 0.8/N = 15, T = 30$ and $S_1, Cut = 0.8, N = 15, T = 45$. The results indicate that for both S_1 and S_2 the heuristic performance deteriorates as the number of products increases. This situation is pronounced when the discrete production policy is relaxed and the furnace may run up to capacity (setting S_1). Regarding the number of periods, it seems that the performance of the heuristic behaves differently from S_1 to S_2. For setting S_1 its performance worsens as the number of products increases, whereas for S_2 the gap between the lower and the upper bound either tends to decrease as T increases ($Cut = 0.6$) or it is almost not influenced by T ($Cut = 0.8$). It is clear that the algorithm performs very well on S_2 since the largest gap is less than 10%. The performance on S_1 is not that encouraging. The total computational times and the number of iterations in each run are given in Table 5 for $S_1, Cut = 0.8$. We note that for all instances the total running time never exceeded 1 hour.

Table 3: Gap (%) between the lower and upper bounds for setting S_1, $K = 3$

	$Cut = 0.6$		$Cut = 0.8$	
N	$T = 30$	$T = 45$	$T = 30$	$T = 45$
5	4.0/5.7/7.4	5.9/8.0/12.3	8.9/12.5/17.5	19.4/22.8/26.9
10	7.3/11.9/19.0	11.8/23.9/30.8	20.7/24.8/28.7	30.8/47.6/53.1
15	13.4/22.0/29.9	29.1/37.4/47.3	29.2/38.5/49.2	

minimum / average / maximum gap (%)

Table 4: Gap (%) between the lower and upper bounds for setting S_2, $K = 3$

	$Cut = 0.6$		$Cut = 0.8$	
N	$T = 30$	$T = 45$	$T = 30$	$T = 45$
5	1.3/2.5/3.7	0.6/1.0/1.4	0.9/1.2/1.9	0.6/1.3/4.0
10	2.8/5.6/9.9	2.4/3.4/5.2	1.1/1.7/2.7	0.7/2.4/6.2
15	4.5/6.8/8.5	3.3/5.2/6.9	1.3/3.0/5.0	

minimum / average / maximum gap (%)

Table 5: Average running times for $Cut = 0.8$ and setting S_1, $K = 3$

	$T = 30$		$T = 45$	
N	# Iterations	CPU time (secs)	# Iterations	CPU time (secs)
5	32	63	37	134
10	55	655	59	1,174
15	76	1,864	81	2,976

Table 6 presents the average number of branch-and-bound nodes and the optimality gap (%) obtained by CPLEX 10.1 for the same instances as those presented in Table 3 within a one hour time limit on formulation F_2 strengthened by all the cuts developed in Section 3.2. An empty field means that CPLEX 10.1 was not able to generate any feasible solutions within the time limit. There was even an instance for $T = 45, N = 15, Cut = 0.6$ where CPLEX 10.1 was not able to find a solution. This instance was discarded and it was not included in the reported average. As the size of the instance gets bigger our results clearly outperform those obtained by CPLEX. Only for $N = 5, 10$ and $T = 30, 45$ with $Cut = 0.6$ and $N = 5$, $T = 30, 45$ with $Cut = 0.8$ CPLEX 10.1 outperforms our Lagrangian approach. In all other cases we produce substantially better gaps (and also lower running times). It is also clear from the substantially lower number of branch-and-bound nodes as N increases that LP relaxations become much more difficult. This is another advantage of our Lagrangian approach since we do not solve LP relaxations and thus our algorithm is more scalable.

Table 6: CPLEX 10.1 optimality gap (%) and nodes within the one hour time limit

	$Cut = 0.6$				$Cut = 0.8$			
	$T = 30$		$T = 45$		$T = 30$		$T = 45$	
N	Nodes	Gap	Nodes	Gap	Nodes	Gap	Nodes	Gap
5	215,701	0.7%	116,900	1.7%	61,201	1.4%	38,563	4.5%
10	26,712	8.3%	7,205	12.7%	10,481		4,183	
15	5,932	28.1%	1,268	58.7%	2,367		859	

Table 7 displays the same statistics as Table 3 for larger instances, made up of four machines for $Cut = 0.8$. Comparing the results in Tables 3 and 7 it is clear that the gap improves as the number of machines increases. Here again, the algorithm failed to find feasible solutions for hardiest instance ($K = 4, S_1, N = 15, T = 45$). We note that CPLEX 10.1 runs out of memory for instances $N = 15, T = 45, Cut = 0.8$ without finding any upper bound.

Table 7: Gap (%) between the lower and upper bounds for setting S_1, $K = 4$

	$Cut = 0.8$	
N	$T = 30$	$T = 45$
5	4.8/5.7/6.8	7.4/10.9/15.2
10	12.9/19.5/25.9	32.2/44.8/53.0
15	18.5/26.9/38.2	

minimum / average / maximum gap (%)

Figure 4 shows the trend of the gap between the lower and upper bound in the Lagrangian approach for an instance of the type $K = 3, S_1, Cut = 0.8, N = 5, T = 30$. After approximately 25 iterations with a lower bound improvement, the gap appears to stabilize.

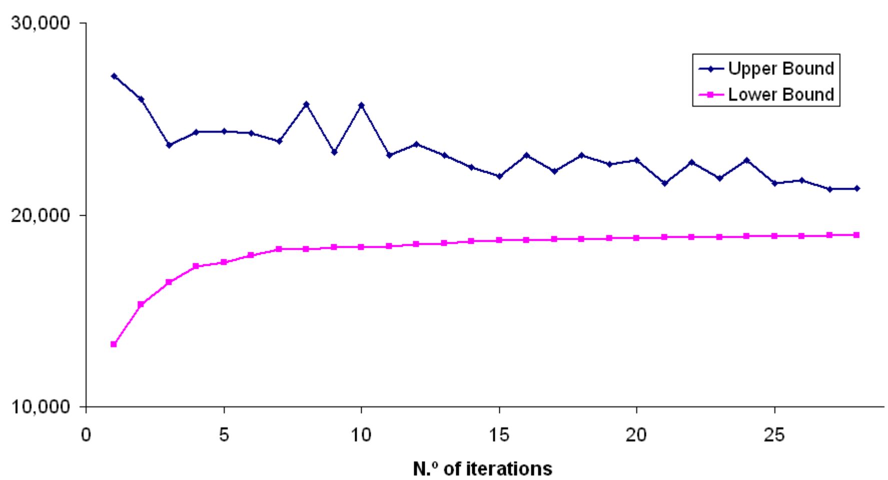

Figure 4: Solution example for the instance $K = 3, S_1, Cut = 0.8, N = 10, T = 30$

Finally, Table 8 gives the solution gap (%) and the main characteristics of different real-world instances of our problem. The results for these instances outperform considerably those obtained previously for randomly generated instances. Two main reasons that make the problem slightly easier for real-world instances are as follows. First, in real-world instances technology constraints do not allow products to be assigned to some machines, which reduces the number of variables. Second, the demand is not observed every fifth period for every product but it is more sparse (e.g., two orders of the same product may lag more than 10 time periods). It can be seen, despite the small sample dimension, that the gap increases with the number of time periods, and as the instances become more highly capacitated.

Table 8: Gap (%) between the lower and upper bounds for various real-life instances

N	T	K	Cut	m_{ik}/M_{ik}	Gap(%)
7	35	3	0.83	0.64	10.5
4	24	3	0.71	0.74	1.4
7	13	3	0.77	0.67	4.1
6	32	3	0.81	0.63	1.8
14	19	3	0.84	0.65	5.9
14	16	5	0.63	0.69	4.9
8	13	4	0.50	0.74	3.7
7	23	4	0.80	0.74	3.9
11	16	4	0.60	0.75	5.2
9	15	3	0.80	0.64	4.9
8	18	5	0.80	0.68	4.6
12	21	3	0.71	0.67	5.1

6 Conclusions

In this paper, we address the short-term production planning and scheduling problem faced by a glass container company, where a limited renewable, continuous resource is distributed to a set of parallel molding machines. After developing an exact formulation, we simplify it into an extension of the standard continuous lotsizing problem (CSLP). Computationally, the problem corresponding to this model is NP-hard. We then reduce it to a network flow type model, which is decoupled by machine through a Lagrangian relaxation scheme. Since the subproblems are easily solvable, we are able to run a large number of iterations in a short period of time. Feasible solutions are generated with a model-based Lagrangian heuristic. We carry out a set of computational experiments on relatively large real-world and randomly generated instances.

The contributions of this research are fourfold. First, we solve a relevant industrial problem within a very competitive industry. Second, we are not aware of any research tackling CSLP with multiple-machines with the presence of production losses (due to setups and capacity surplus). Due to its inherent complexity, the research community has overlooked multi-machine CSLP with sequence dependent setups. Third, we have employed an efficient Lagrangian based heuristic for this problem. Finally, we have implemented valid inequalities that enable us to reduce the integrality gap.

CSLP has been neglected by researchers due to its computational challenges. This study further suggests that this problem (clearly useful in practice) is a challenging area for future research.

Additionally, opportunity costs for not pulling the most out of a resource are critical in capital intensive industries. Though our computational results are very encouraging, there is room for improvements. We need to study under what conditions the presented valid inequalities are strong (define facets), since it seems that their impact on the improvement of the lower bound is dependent upon the instance type. Another important research question is to find valid inequalities to be added to (19)-(23) in order that its LP relaxation has no integrality gap (that provide a complete description of the respective polyhedron). Additionally, it is clearly important to find strategies to combine even stronger valid inequalities based on the polyhedral structure of this problem with tighter reformulations. These inequalities should take into account the furnace capacity constraints. Lagrangian relaxation appears to be very suitable for determining feasible solutions. An improvement heuristic may be developed to even further close the gap between the lower and upper bound.

References

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Englewood Cliffs, NJ, 1993.

B. Almada-Lobo, D. Klabjan, M. A. Carravilla, and J. F. Oliveira. Single machine multi-product capacitated lotsizing with sequence-dependent setups. *International Journal of Production Research*, 45:4873–4894, 2007.

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

D. Bertsekas. *Network optimization: continuous and discrete models.* Athena Scientific, Belmont, MA, 1998.

M. Constantino. A polyhedral approach to a production planning problem. *Annals of Operations Research*, 96:75–95, 2000.

R. Dematta and M. Guignard. Dynamic production scheduling for a process industry. *Operations Research*, 42:492–503, 1994a.

R. Dematta and M. Guignard. Studying the effects of production loss due to setup in dynamic production scheduling. *European Journal of Operational Research*, 72:62–73, 1994b.

R. Dematta and M. Guignard. The performance of rolling production schedules in a process industry. *IIE Transactions*, 27:564–573, 1995.

A. Drexl and A. Kimms. Lot sizing and scheduling - survey and extensions. *European Journal of Operational Research*, 99:221–235, 1997.

B. Fleischmann. The discrete lot-sizing and scheduling problem with sequence-dependent setup costs. *European Journal of Operational Research*, 75:395–404, 1994.

A. Geoffrion. Lagrangian relaxation for integer programming. *Mathematical Programming Study*, 2:82–114, 1974.

M. Held, P. Wolfe, and H. Crowder. Validation of subgradient optimization. *Mathematical Programming*, 6:62–88, 1974.

K. S. Hindi. Solving the single-item, capacitated dynamic lot-sizing problem with startup and reservation costs by tabu search. *Computers and Industrial Engineering*, 28:701–707, 1995.

R. Jans and Z. Degraeve. An industrial extension of the discrete lot-sizing and scheduling problem. *IIE Transactions*, 36:47–58, 2004.

R. Jans and Z. Degraeve. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *European Journal of Operational Research*, 177:1855–1875, 2007.

U. Karmarkar and L. Schrage. The deterministic dynamic product cycling problem. *Operations Research*, 33:326–45, 1985.

U. S. Karmarkar, S. Kekre, and S. Kekre. The dynamic lot-sizing problem with startup and reservation costs. *Operations Research*, 35:389–398, 1987.

Y. Pochet and L. A. Wolsey. *Production Planning by Mixed Integer Programming*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006.

M. Salomon, M. M. Solomon, L. N. Van Wassenhove, Y. Dumas, and S. Dauzere-Peres. Solving the discrete lotsizing and scheduling problem with sequence dependent set-up costs and set-up times using the travelling salesman problem with time windows. *European Journal of Operational Research*, 100:494–513, 1997.

R. A. Sandbothe. A user interactive heuristic procedure for solving the multiple product cycling problem. *Computers and Operations Research*, 23:897–907, 1996.

F. Vanderbeck. Lot-sizing with start-up times. *Management Science*, 44:1409–1425, 1998.

L. A. Wolsey. MIP modelling of changeovers in production planning and scheduling problems. *European Journal of Operational Research*, 99:154–165, 1997.