A Mathematical Model for the Synchronized and Integrated Two-Level Lot Sizing and Scheduling Problem

Claudio Fabiano Motta Toledo
Universidade Estadual de Campinas (UNICAMP)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Departamento de Engenharia de Sistemas (DENSIS)
Rua Albert Einstein, 400 · Caixa Postal 6101 · Cep 13083–970, Campinas, SP · Brazil
email: claudio@densis.fee.unicamp.br

Alf Kimms
Chair of Logistics and Traffic Mgmt.
Dept. of Technology and Operations Mgmt.
Mercator School of Management
University of Duisburg–Essen, Campus Duisburg
47048 Duisburg · Germany
email: alf.kimms@uni-duisburg-essen.de

Paulo Morelato França
Universidade Estadual de Campinas (UNICAMP)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Departamento de Engenharia de Sistemas (DENSIS)
Rua Albert Einstein, 400 · Caixa Postal 6101 · Cep 13083–970, Campinas, SP · Brazil
email: franca@densis.fee.unicamp.br

Reinaldo Morabito
Universidade Federal de São Carlos (USFCar)
Departamento de Engenharia de Produção (DEP)
Via Washington Luiz, km.235 · Caixa Postal 676 · Cep 13.565-905, São Carlos, SP · Brazil
email: morabito@power.ufscar.br

Abstract

The present paper proposes a multi-level lot sizing and scheduling problem with parallel machines, capacity constraints and sequence-dependent setup costs and times. This problem was motivated by a real situation found in some industrial settings, mainly soft drink companies. In this kind of industrial process, the production involves two interdependent levels with decisions about raw material storage and soft drink bottling. The various raw materials are stored in tanks, from which they flow to bottling production lines. The challenge is to simultaneously determine the lot sizing and scheduling of raw materials in tanks and also in the bottling lines, where setup costs and times depend on the previous items stored and bottled. We call this problem a Synchronized and Integrated Two-Level Lot Sizing and Scheduling Problem (SITLSP). A mixed-integer linear model is proposed with various combined constraints that are not usually dealt with in the literature. This complex model was solved by the GAMS/CPLEX software. The lack of similar
models led us to create a set of instances to evaluate the model and some computational results are reported here.

**Keywords:** lot sizing and scheduling, mixed–integer linear modeling, parallel machines.

**Introduction**

The problem studied here was motivated by a real situation found in some industrial settings, such as soft drink companies. In this kind of industry, production involves two interdependent levels with decisions about raw material storage and soft drink bottling. To make the explanation easier, the problem is posed as a soft drink industrial problem. Taking this into account, what we refer to as product in the problem is a pair of flavor of soft drink and type of bottle. Each of the soft drinks is available in one or more bottle types, such as glass bottles, plastic bottles (PET), cans or bag–in–boxes of different sizes.

To fill the soft drink into a bottle, that is, to produce a certain product, a bottling production line is used. Several products share a common production line, which cannot produce more than one product at a time. Thus, from time to time the product being produced on a line must be switched. Nevertheless, whenever the product switches, a setup time (of up to several hours) is required to prepare the production line for the next product. Production lines can maintain the setup state. When the production of a product is preempted for a while and continued after some idle time, then no setup is required. Setup times are sequence-dependent, meaning that the order in which products are produced affects the required time to perform a setup. Moreover, more than one production line exists and, in general, a subset of the lines is capable of producing a certain product, so that products can be produced using production lines in parallel.

The raw material (i.e., the soft drink) that is bottled on a production line comes from a storage tank with limited storage capacity. Of course, several soft drinks can not be put in a tank simultaneously. Hence, from time to time a tank must be filled with a particular soft drink, which might be another drink or the same as before. Nevertheless, whenever a tank is filled, a significant (sequence-dependent) setup time occurs to clean and fill the tank, even if the same soft drink is filled in the tank as before. For technical reasons, a tank can only be filled up when it is empty. During the time of filling up a tank, nothing can be pumped to a production line from that tank. Note that various tanks exist so that several production lines can produce different products at the same time.

It is important to note that there is no fixed assignment between tanks and production lines. Every tank can be connected to every production line. Moreover, it is possible to have one tank being connected to several production lines at the same time, as well as one production line being connected to several tanks (containing the same soft drink) at the same time.

Bottling is done to meet some given demand (per week) with a production planning horizon of four weeks. The decision maker has to find out how many units of what product should be produced when and on what production line. To be able to do so, the tanks must be filled appropriately. The objective is to minimize the total sum of setup, inventory holding, and production costs. Thus, we face a multi–level (two levels) lot sizing and scheduling problem with parallel production lines and capacity constraints as well as sequence-dependent setup times.
A lot sizing and scheduling problem with parallel processors (lines and tanks) and sequence-dependent setup times has to be solved in each one of these two levels. The solutions for the problem proposed here have to integrate these two lot sizing and scheduling problems. Furthermore, a line can produce a soft drink only if the raw material of this soft drink is stored in a connected tank at the same time. Thus, the production in lines and the storage in tanks must be compatible with each other during the production time horizon and we have a synchronization problem here. For these reasons, we call the present problem a Synchronized and Integrated Two–level Lot Sizing and Scheduling Problem (SITLSP).

This problem covers several issues of lot sizing and scheduling that have been treated in the literature before. The challenging aspect is the combination of all these issues. Capacitated lot sizing (and scheduling) is a topic of broad interest and has attracted many researchers. We refer to [5] and the references therein for overviews. While capacitated lot sizing (and scheduling) is an \( \mathcal{NP} \)-hard optimization problem, finding a feasible solution is easy (e.g., a lot–for–lot–like policy), if no setup times are to be taken into account. If setup times are present, the problem of finding a feasible solution is \( \mathcal{NP} \)-complete already. See, for example, [6] for a discussion of lot sizing and scheduling with sequence-dependent setup costs. A more recent paper on sequence–dependent setup costs (with parallel machines) is [8]. One of the few (optimal) procedures for solving lot sizing and scheduling problems with sequence-dependent setup times is found in [7], where more references on former studies are given. Heuristics dealing with sequence-dependent setup times are described in [12] and [13].

For multi–level lot sizing problems, some publications exist (see, e.g., [1], [16] and [17]), but for multi–level lot sizing and scheduling, the literature is much more sparse (see, e.g., [9] and [10]), because multi–level problems are difficult to tackle, even if the capacity is unlimited. We note that there are parallel machines in our problem. A study on parallel machines with constant demand is found in [3], but in our case the demand is dynamic. In [2], the benefits of dedicating one or more parallel machines to a single problem is discussed. The work in [4] deals with rolling production schedules for lot sizing with parallel machines. More recently, a lot sizing problem in a semiconductor assembly with parallel machines was described in [11] and [15].

To the best of our knowledge, the only work that comes close to our case is the one described in [14], which is a multi–level extension of [13]. In that approach, however, it might be necessary to split up a lot in smaller lots in order to not lose generality. However this would not be a useful idea in our case, because every new lot for the tanks requires a new setup, which is not desired.

To sum up, it seems that for the problem at hand no existing approach is directly applicable. Thus, formulating a specific model seems to be a very first step to close that gap. In the next section (A Model formulation), we propose a model for the SITLSP. In the section on Computational Results we present some computational results to solve the model using the modeling language GAMS in combination with the CPLEX solver. Finally, in the section on Conclusions we present concluding remarks and discuss perspectives for future research.
A Model Formulation

A mathematical model as the one given below serves several purposes. First, it provides an unequivocal definition of the problem better than a verbal description can do. Second, small (artificially created) instances can then be solved using commercial software packages to find an optimum solution to serve as a benchmark for heuristics (as mentioned, we used GAMS/CPLEX to test the model). For instance, see [18] where interesting reformulations are presented to make standard software applicable. Finally, the model itself might be used in future research to develop tailor-made mathematical programming solution approaches. Hence, we believe that a mathematical model for a complex real-world problem as the one at hand is a contribution in its own right.

The notation being used in the model description below is summarized in the appendix. Suppose that we have \( J \) kinds of soft drinks (raw materials) in different containers and \( J \) kinds of products. We consider \( L \) production lines as well as \( \tilde{L} \) tanks. The planning horizon is subdivided into \( T \) (macro-)periods.

Furthermore, for modelling reasons, we assume that at most \( S \) lots per macro-period can be scheduled on each production line, and at most \( \tilde{S} \) lots per macro-period and per tank can be scheduled. Values \( S \) and \( \tilde{S} \) can be set to sufficiently large numbers so that this assumption is not restrictive for practical purposes. Furthermore, for modelling reasons — and we will describe this in more detail later on — we assume that each macro-period is subdivided into \( T^m \) micro-periods.

As usual in lot sizing models, the objective is to minimize the total sum of setup costs, inventory holding costs, and production costs. The following expression formally defines these cost components. The first line in this expression represents the production lines and the second, the tanks. Note that a feasible solution, which fulfills all demand rights in time without violating all constraints to be described, may not exist. Hence, we allow that \( q^0_j \) units of the demand for product \( j \) may not be produced to guarantee that the model can always be solved. Of course, a very high penalty \( M \) is attached to such shortages (see the third line of the expression below) so that, whenever there is a feasible solution that fulfills all demand, we would prefer that one.

\[
\sum_{i=1}^{J} \sum_{j=1}^{J} \sum_{l=1}^{L} \sum_{s=1}^{T^m} s_{ijl} z_{ijls} + \sum_{j=1}^{J} \sum_{t=1}^{T} h_{jt} I_{jt} + \sum_{j=1}^{J} \sum_{l=1}^{L} \sum_{s=1}^{T^m} v_{ijl} q_{ijls} + \sum_{i=1}^{J} \sum_{j=1}^{J} \sum_{k=1}^{\tilde{L}} \sum_{s=1}^{T^m} \tilde{s}_{ijk} \tilde{z}_{ijks} + \sum_{j=1}^{J} \sum_{k=1}^{\tilde{L}} \sum_{l=1}^{T} \tilde{h}_{jkT} \tilde{l}_{jkT} + \sum_{j=1}^{J} \sum_{k=1}^{\tilde{L}} \sum_{s=1}^{T^m} \tilde{v}_{jk} \tilde{q}_{jks} + M \sum_{j=1}^{J} q^0_j
\]  

(1)

The underlying idea for modelling lot size plans in our application combines issues from the so-called general lot sizing problem (GLSP) (see [12]) and the so-called continuous setup lot sizing problem (CSLP) (see [5] for a comparison of these models). From the GLSP, we adopt the idea of defining a predefined number (\( S \) for the production lines and \( \tilde{S} \) for the tanks) of slots per macro-period, so that at most one lot can be scheduled per slot. What needs to be determined is for what product or soft drink, respectively, a particular slot should be reserved, and what lot size (a lot of size zero is possible) should be scheduled, given a valid reservation for a slot.
Figure 1 illustrates this idea. Let us take 2 macro–periods with $S = \overline{S} = 2$ slots, $J = 5$ raw materials, $J = 6$ products, $L = 3$ tanks and $L = 3$ lines. Suppose that $rma$ (raw material A) is used to produce $P1$ (product 1), $rmB$ produces $P2$ and $P3$, $rmC$ produces $P4$, $rmD$ produces $P5$ and $rmE$ produces $P6$. The number of products in lines and raw materials in tanks cannot be greater than the total number of slots ($S = \overline{S} = 2$) in each macro–period. Note that there is not a full slot occupation in figure 1. For instance, only one slot is occupied by raw materials and products, respectively, in tank $Tk3$ and line $L3$ during the first macro–period. In the second macro–period, there is no slot occupied in $Tk3$ and $L3$. The different slot sizes show us that there are different numbers of products in each slot. In this way, variables and constraints indexed by slots in the model will enable us to describe the lines and tank occupation simultaneously.

In a multi–level setting (recall that we have two levels, the tanks and the production lines), it is necessary to coordinate the lots scheduled on one level with the lots scheduled on the other level. For that reason we introduce small micro–periods in a CSLP–like manner, so that a lot on the production line level that must wait for a lot on the tank level requires the tank level lot to be ready at least one micro–period before the start of the production line level lot.

Figure2 illustrates this idea starting from the slot occupation in figure 1 and synchronizing the slot scheduled in the two levels. Each macro–period was divided into 5 micro–periods with same length. The acronyms $rma$, $rmB$, $rmC$, $rmD$ and $rmE$ in figure 2 represent the raw material setup time in a tank. The symbols $P1$, $P2$, $P3$, $P4$ and $P5$ represent the processing time of products. For instance, the two slots of $Tk1$ occupied by $rma$ indicate that the tank is refilled using the same raw material during the first macro–period. The slot occupation of $Tk1$ aims to meet the production requirements of $P1$ in line $L1$, where the two slots available were occupied by $P1$. The micro–periods enable us to synchronize the beginning of $P1$ production in the first slot of $L1$ with the ending of $rma$ setup time shown by the first slot of $Tk1$. Moreover, the micro–period enables us to see when the refilling of $Tk1$ takes place in the second slot of $Tk1$. This tank refilling demands the interruption of $P1$ and the beginning of the second slot occupation of $P1$, after the end of $rma$ setup time in $Tk1$. Taking this into consideration, the model will use constraints and variables indexed by a micro–period that makes it possible to describe this kind of situation.
In short, we can say that the new approach used in the model proposed here is the integration and synchronization of two lot sizing and scheduling problems. This is done using variables and constraints indexed by slots and micro–periods. The slots provide us with the line and tank sequence of occupation. The micro–periods inform us of when this occupation will take place. Slots and micro–periods together enable us to use variables and constraints that integrate and synchronize the lot sizing and scheduling problems in the two levels of this problem.

Let us focus on the production lines first to see how this can be expressed formally. As a matter of fact, a particular product can be produced on some lines, but not on all in general. If $x_{jls}$ is a binary variable that indicates for what product $j$ a certain slot $s$ is reserved for, it is easy to express that some reservations will never occur given $L_j$, the production lines on which product $j$ can be produced:

$$x_{jls} = 0 \quad j = 1, \ldots, J \quad l \in \{1, \ldots, L\} \setminus L_j \quad s = 1, \ldots, T \cdot S$$

(2)

It is required to have a reservation for each single slot no matter whether or not that slot will indeed be used for producing something:

$$\sum_{j=1}^{J} x_{jls} = 1 \quad l = 1, \ldots, L \quad s = 1, \ldots, T \cdot S$$

(3)

If a product $j$ is scheduled for being produced in a certain slot, then there must be a reservation for that particular product. If $q_{jls}$ is the quantity to be produced and $p_{jl}$ is the amount of production time per unit, $C$, the available time per macro–period, is an upper bound on the time for production, which enable us to link the decision variables $q_{jls}$ and $x_{jls}$:

$$p_{jl} q_{jls} \leq C x_{jls} \quad j = 1, \ldots, J \quad l = 1, \ldots, L \quad s = 1, \ldots, T \cdot S$$

(4)
Depending on the reservations $x_{jls}$ being made, it is easy to figure out whether or not a setup from product $i$ to product $j$ must take place ($z_{ijls}$ indicates this). From this, it is possible to determine the setup time $\sigma_{ls}$ to be included just in front of a certain slot:

$$z_{ijls} \geq x_{jls} + x_{il,s-1} - 1$$  \hspace{1cm} i, j = 1, \ldots, J \quad l = 1, \ldots, L \quad s = 1, \ldots, T \cdot S \quad (5)$$

$$\sigma_{ls} = \sum_{i=1}^{J} \sum_{j=1}^{J} \alpha_{ij} z_{ijls}$$  \hspace{1cm} l = 1, \ldots, L \quad s = 1, \ldots, T \cdot S \quad (6)$$

For the first slot of a macro-period $t$, some setup time $\omega_{lt}$ may be at the end of the previous macro-period. If so, the end of the last lot in that previous macro-period must leave sufficient time for this setup time at the end of the previous macro-period:

$$\omega_{lt} \leq \sigma_{l,(t-1)S+1}$$  \hspace{1cm} l = 1, \ldots, L \quad t = 1, \ldots, T \quad (7)$$

$$t \cdot C - \sum_{\tau=(t-1)T+1}^{tT} C_{m}^{\tau} x_{l,t,s}^{E} \geq \omega_{lt+1}$$  \hspace{1cm} l = 1, \ldots, L \quad t = 1, \ldots, T - 1 \quad (8)$$

The time capacity per macro-period is constrained to be $C$ time units. Therefore, for each production line, the total sum of production times and setup times in a certain macro-period must not exceed that limit:

$$\sum_{s=(t-1)S+1}^{tS} (\sigma_{ls} + \sum_{j=1}^{J} p_{jl} q_{jls}) \leq C + \omega_{lt}$$  \hspace{1cm} l = 1, \ldots, L \quad t = 1, \ldots, T \quad (9)$$

The inventory balance states that at the end of a macro-period, the number of products on stock equals which we had on stock at the beginning of the period, plus what we have produced, minus the demand. As stated above, we would like to guarantee a feasible solution by allowing shortages. This is modeled by allowing $q_{j}^{0}$ units being “produced” in the first period without using capacity.

$$I_{j1} = I_{j0} + q_{j}^{0} + \sum_{l=1}^{L} \sum_{s=1}^{S} q_{jls} - d_{j1}$$  \hspace{1cm} j = 1, \ldots, J \quad (10)$$

$$I_{jt} = I_{j,t-1} + \sum_{l=1}^{L} \sum_{s=(t-1)S+1}^{tS} q_{jls} - d_{jt}$$  \hspace{1cm} j = 1, \ldots, J \quad t = 2, \ldots, T \quad (11)$$

Two consecutive slots must be scheduled in such a way that precedence constraints among these slots are respected. In other words, the finishing time of one slot, minus the finishing time of the previous slot, must be at least as large as the production time of the lot in the one slot, plus the required setup time in advance of that lot:

$$\sum_{\tau=(t-1)T+1}^{tT} C_{m}^{\tau} x_{l,t,s}^{E} \geq \sigma_{ls} + \sum_{j=1}^{J} p_{jl} q_{jls}$$  \hspace{1cm} l = 1, \ldots, L \quad t = 1, \ldots, T \quad s = (t-1)S + 2, \ldots, t \cdot S \quad (12)$$
To have a well-defined schedule, every slot must have a unique finishing time and a unique starting time for the lot scheduled in that slot:

\[
\sum_{\tau=(t-1)T^m+1}^{t-T^m} C^m x^E_{l,(t-1)S+1,\tau} \\
\geq (t-1)C + \sigma_{l,(t-1)S+1} - \omega_l + \sum_{j=1}^{J} p_j q_{j,l(t-1)S+1} \\
l = 1, \ldots, L \\
t = 1, \ldots, T \\
(13)
\]

While \(x_{jls}\) indicates whether or not a certain slot is reserved to produce a particular product, \(u_{ls}\) indicates whether or not that slot is indeed used to schedule a lot. If the slot is not used, then the lot size must be zero. On the other hand, if a real lot is indicated, then at least a small amount \(\epsilon\) must be produced:

\[
\sum_{j=1}^{J} p_j q_{jls} \leq C u_{ls} \\
l = 1, \ldots, L \\
s = 1, \ldots, T \cdot S \\
(16)
\]

\[
\epsilon u_{ls} \leq \sum_{j=1}^{J} q_{jls} \\
l = 1, \ldots, L \\
s = 1, \ldots, T \cdot S \\
(17)
\]

In case a slot is not used, no time should be allocated for that slot, which means that the starting and ending time of the slot are the same, because a lot of size zero is scheduled:

\[
\sum_{\tau=(t-1)T^m+1}^{t-T^m} x^E_{l,s\tau} = 1 \\
l = 1, \ldots, L \\
t = 1, \ldots, T \\
(14)
\]

\[
\sum_{\tau=(t-1)T^m+1}^{t-T^m} x^B_{l,s\tau} = 1 \\
l = 1, \ldots, L \\
t = 1, \ldots, T \\
(15)
\]

\[
\sum_{\tau=(t-1)T^m+1}^{t-T^m} \tau x^E_{l,s\tau} - \sum_{\tau=(t-1)T^m+1}^{t-T^m} \tau x^B_{l,s\tau} \leq T^m u_{ls} \\
l = 1, \ldots, L \\
t = 1, \ldots, T \\
s = (t-1)S + 1, \ldots, t \cdot S \\
(18)
\]

\[
\sum_{\tau=(t-1)T^m+1}^{t-T^m} \tau x^B_{l,s\tau} - \sum_{\tau=(t-1)T^m+1}^{t-T^m} \tau x^E_{l,s\tau} \leq u_{ls} \\
l = 1, \ldots, L \\
t = 1, \ldots, T \\
s = (t-1)S + 1, \ldots, t \cdot S \\
(19)
\]

To avoid some redundancy in the model, whenever a slot is not used, it should start as soon as the previous slot ends:

\[
\sum_{\tau=(t-1)T^m+1}^{t-T^m} \tau x^B_{l,s\tau} - \sum_{\tau=(t-1)T^m+1}^{t-T^m} \tau x^E_{l,s-1,\tau} \leq T^m u_{ls} \\
l = 1, \ldots, L \\
t = 1, \ldots, T \\
s = (t-1)S + 2, \ldots, t \cdot S \\
(20)
\]

\[
\sum_{\tau=(t-1)T^m+1}^{t-T^m} \tau x^B_{l,(t-1)S+1\tau} - ((t-1)T^m + 1) \\
\leq T^m u_{l,(t-1)S+1} \\
l = 1, \ldots, L \\
t = 1, \ldots, T \\
(21)
\]
Another way to avoid redundancy is to enforce that, whenever a slot is used in a macro-period, every other previous slot in the same macro-period must also be used. As a consequence, possibly empty slots in a macro-period are the last ones in that macro-period:

\[ u_{ls} \geq u_{l,s+1} \]

The start and the end of a slot must be chosen in such a way that the start indicates the first micro-period in which the lot is produced, if there is a real lot for that slot, and the end indicates the last:

\[ \epsilon u_{ls} + \sum_{\tau = (t-1)T_{m} + 1}^{T_{m}} C_{m}^{\tau} x_{ls}^{E} \]

\[ - \sum_{\tau = (t-1)T_{m} + 1}^{T_{m}} C_{m}^{\tau} x_{ls}^{B} \leq \sum_{j=1}^{J} p_{jl} q_{jls} \]

\[ \sum_{\tau = (t-1)T_{m} + 1}^{T_{m}} C_{m}^{\tau} x_{ls}^{E} \]

\[ - \sum_{\tau = (t-1)T_{m} + 1}^{T_{m}} C_{m}^{\tau} x_{ls}^{B} \geq \sum_{j=1}^{J} p_{jl} q_{jls} - C_{m}^{l} \]

To link the production lines and the tanks, we first figure out which amount of a soft drink being bottled on a line comes from which tank:

\[ q_{jls} = \sum_{k=1}^{L} \sum_{\tau' = (t-1)T_{m} + 1}^{tT_{m}} q_{kjl\tau'} \]

Clearly, the soft drink taken from a tank must be taken during the time window in which the soft drink is filled into bottles:

\[ \sum_{k=1}^{L} \sum_{j=1}^{J} p_{jl} q_{kjl\tau'} \leq \sum_{\tau' = \tau}^{tT_{m}} C_{m}^{\tau} x_{ls}^{E} \]

\[ \sum_{k=1}^{L} \sum_{j=1}^{J} p_{jl} q_{kjl\tau'} \leq \sum_{\tau' = (t-1)T_{m} + 1}^{T_{m}} C_{m}^{\tau} x_{ls}^{B} \]

In the first micro-period of a lot, some time \( \delta_{ls} \) is reserved for setup or idle time, while the remaining time is production time:

\[ \delta_{ls} = \left( \frac{1}{tT_{m} - \sum_{\tau = (t-1)T_{m} + 1}^{T_{m}} \tau x_{ls}^{E} - \sum_{\tau = (t-1)T_{m} + 1}^{T_{m}} \tau x_{ls}^{B} + 1} \right) C_{m}^{l} \]
\[ - \sum_{j=1}^{J} p_{jl}q_{jkl} \]
\[ \sum_{k=1}^{L} \sum_{j=1}^{J} p_{jl}q_{kjls} \leq C^m - \delta_{ls} + (1 - x_{ls}^B)C^m \]
\[ l = 1, \ldots, L \]
\[ t = 1, \ldots, T \]
\[ s = (t-1)S + 1, \ldots, t \cdot S \]
\[ \tau = (t-1)T^m + 1, \ldots, t \cdot T^m \]

Now, we can consider the tanks. Similar to the production lines, not every tank can be used to store every soft drink. What is allowed and what is not, is defined first:

\[ \bar{x}_{jks} = 0 \]
\[ j = 1, \ldots, J \]
\[ l \in \{1, \ldots, \bar{L}\} \backslash \bar{L}_j \]
\[ s = 1, \ldots, T \cdot \bar{S} \]

At any one given time, only one soft drink can be in a particular tank. We use more or less the same idea as for production lines and consider reservations of slots:

\[ \sum_{j=1}^{J} \bar{x}_{jks} = 1 \]
\[ k = 1, \ldots, \bar{L} \]
\[ s = 1, \ldots, T \cdot \bar{S} \]

The amount that can be filled into a tank has an upper limit, but for technical reasons, we also have to respect minimum lot sizes whenever something is filled into a tank:

\[ \bar{q}_{jks} \leq \bar{Q}_k \bar{x}_{jks} \]
\[ j = 1, \ldots, \bar{J} \]
\[ k = 1, \ldots, \bar{L} \]
\[ s = 1, \ldots, T \cdot \bar{S} \]

\[ \bar{q}_{jks} \leq \bar{Q}_k \bar{u}_{ks} \]
\[ j = 1, \ldots, \bar{J} \]
\[ k = 1, \ldots, \bar{L} \]
\[ s = 1, \ldots, T \cdot \bar{S} \]

\[ \sum_{j=1}^{J} \bar{q}_{jks} \geq \bar{Q}_k \bar{u}_{ks} \]
\[ k = 1, \ldots, \bar{L} \]
\[ s = 1, \ldots, T \cdot \bar{S} \]

Given the reservations for the slots, it is easy to figure out when sequence-dependent setups occur. In contrast to the production lines, the setup state is not preserved in the tanks, that is, a tank must be setup again (cleaned and filled), even if the same soft drink has been in the tank just before. It must be taken care that, in the model, setups occur only if the tank is really used, because changing the reservation for a tank does not necessarily mean that a setup takes place:

\[ \bar{z}_{ijks} \geq \bar{x}_{jks} + \bar{x}_{ik,s-1} - 2 + \bar{u}_{ks} \]
\[ i, j = 1, \ldots, \bar{J} \]
\[ k = 1, \ldots, \bar{L} \]
\[ s = 1, \ldots, T \cdot \bar{S} \]

\[ \bar{x}_{jks} - \bar{x}_{j,k,s-1} \leq \bar{u}_{ks} \]
\[ j = 1, \ldots, \bar{J} \]
\[ k = 1, \ldots, \bar{L} \]
\[ s = 1, \ldots, T \cdot \bar{S} \]

\[ \bar{\sigma}_{ks} = \sum_{i=1}^{J} \sum_{j=1}^{J} \bar{\sigma}_{ijk} \bar{z}_{ijks} \]
\[ k = 1, \ldots, \bar{L} \]
\[ s = 1, \ldots, T \cdot \bar{S} \]
\[ \bar{\omega}_{kt} \leq \bar{\sigma}_{k,(t-1)S+1} \]
\[ \sum_{s=(t-1)S+1}^{tS} \bar{\sigma}_{ks} \leq C + \bar{\omega}_{kt} \]
\[ t = 1, \ldots, T \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T \]

The amount that is filled into a tank must be scheduled, so that the micro–period in which the soft drink comes into the tank is known:

\[ \bar{q}_{jks} = \sum_{\tau=(t-1)T^{m}+1}^{tT^{m}} \bar{q}_{jks\tau} \]
\[ j = 1, \ldots, \bar{J} \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T \]
\[ s = (t-1)\bar{S} + 1, \ldots, t \cdot \bar{S} \]

The inventory balance for a tank is simple. What is in the tank at the end of a micro–period is what was in the tank at the beginning of that micro–period, plus what comes into the tank in that micro–period, minus what is taken from that tank for being bottled. Later on we will see that tanks can only be refilled if they are empty. To coordinate the production lines and the tanks, it is required that what is taken out of a tank must have been filled into the tank at least one micro–period ahead:

\[ \bar{I}_{jkt} = \bar{I}_{jk,\tau-1} + \sum_{s=(t-1)S+1}^{tS} \bar{q}_{jks\tau} \]
\[ j = 1, \ldots, \bar{J} \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T \]
\[ \tau = (t-1)T^{m} + 1, \ldots, t \cdot T^{m} \]

Slots must be scheduled in such a way that the setup time that corresponds to a slot fits into the time window between the end of the previous slot and the end of the slot under consideration. As for the production lines, the setup for the first slot in a macro–period may overlap the macro–period border to the previous macro–period:

\[ t \cdot C - \sum_{\tau=(t-1)T^{m}+1}^{tT^{m}} C^{m}_{r} \bar{x}_{k,t,S,\tau} \geq \bar{\omega}_{k,t+1} \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T - 1 \]

\[ \sum_{\tau=(t-1)T^{m}+1}^{tT^{m}} C^{m}_{r} \bar{x}_{k,\tau} \]
\[ t = 1, \ldots, T \]
\[ s = (t-1)\bar{S} + 2, \ldots, t \cdot \bar{S} \]

\[ \sum_{\tau=(t-1)T^{m}+1}^{tT^{m}} C^{m}_{r} \bar{x}_{k,(t-1)S+1,\tau} \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T \]
The end time of a scheduled slot must be unique:

\[ \sum_{\tau = (t-1)T_{m+1}}^{tT_m} \bar{x}_{k\tau}^E = 1 \quad k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T \]
\[ s = (t - 1)\bar{S} + 1, \ldots, t \cdot \bar{S} \]  

(46)

The soft drink that is filled into a tank is available for bottling just when the setup of the tank is completed:

\[ \sum_{j=1}^{J} \bar{q}_{j\kappa \tau} \leq \bar{Q}_{\kappa \tau} \bar{x}_{k\tau}^E \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T \]
\[ s = (t - 1)\bar{S} + 1, \ldots, t \cdot \bar{S} \]
\[ \tau = (t - 1)T_{m+1} + 1, \ldots, t \cdot T_{m+1} \]  

(47)

The scheduled starting time for a slot must be unique as well:

\[ \sum_{\tau = (t-1)T_{m+1}}^{tT_m} \bar{x}_{k\tau}^B = 1 \quad k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T \]
\[ s = (t - 1)\bar{S} + 2, \ldots, t \cdot \bar{S} \]  

(48)

\[ \sum_{\tau = (t-2)T_{m+1}}^{tT_m} \bar{x}_{k,(t-1)S+1,\tau}^B = 1 \quad k = 1, \ldots, \bar{L} \]
\[ t = 2, \ldots, T \]  

(49)

\[ \sum_{\tau = 1}^{T_m} \bar{x}_{k1\tau}^B = 1 \quad k = 1, \ldots, \bar{L} \]  

(50)

The time window in which a slot is scheduled must be positive if and only if that slot is used to set up the tank:

\[ \sum_{\tau = (t-1)T_{m+1}}^{tT_m} \tau \bar{x}_{k\tau}^E - \sum_{\tau = (t-1)T_{m+1}}^{tT_m} \tau \bar{x}_{k\tau}^B \leq T_{m}^m \bar{u}_{k} \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T \]
\[ s = (t - 1)\bar{S} + 2, \ldots, t \cdot \bar{S} \]  

(51)

\[ - \sum_{\tau = (t-1)T_{m+1}}^{tT_m} \tau \bar{x}_{k,(t-1)S+1,\tau}^E \leq T_{m}^m \bar{u}_{k,(t-1)S+1} \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 2, \ldots, T \]  

(52)

\[ \sum_{\tau = 1}^{T_m} \tau \bar{x}_{k1\tau}^E \leq T_{m}^m \bar{u}_{k1} \]
\[ k = 1, \ldots, \bar{L} \]  

(53)

\[ \sum_{\tau = (t-1)T_{m+1}}^{tT_m} \tau \bar{x}_{k\tau}^B - \sum_{\tau = (t-1)T_{m+1}}^{tT_m} \tau \bar{x}_{k\tau}^E \leq \bar{u}_{k} \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 1, \ldots, T \]
\[ s = (t - 1)\bar{S} + 2, \ldots, t \cdot \bar{S} \]  

(54)

\[ \sum_{\tau = (t-2)T_{m+1}}^{tT_m} \tau \bar{x}_{k,(t-1)S+1,\tau}^B \leq \bar{u}_{k,(t-1)S+1} \]
\[ k = 1, \ldots, \bar{L} \]
\[ t = 2, \ldots, T \]  

(55)
\[
\sum_{\tau=1}^{T_m} \tau \bar{x}_{k1 \tau}^B - \sum_{\tau=1}^{T_m} \tau \bar{x}_{k1 \tau}^E \leq \bar{u}_{k1} \quad k = 1, \ldots, \bar{L}
\]

Redundancy can be avoided if empty slots are scheduled right after the end of previous slots:

\[
\begin{align*}
\sum_{\tau=(t-1)T^m+1}^{tT^m} \tau \bar{x}_{k\tau}^B & - \sum_{\tau=(t-1)T^m+1}^{tT^m} \tau \bar{x}_{k\tau}^E \\
& \leq T^m \bar{u}_{k}\tau & k = 1, \ldots, \bar{L} \\
& t = 1, \ldots, T \\
& s = (t-1)S + 2, \ldots, t \cdot S
\end{align*}
\]

\[
\sum_{\tau=(t-2)T^m+1}^{tT^m} \tau \bar{x}_{k(t-1)S+1 \tau}^B = \tau \bar{x}_{k(t-1)S+1 \tau}^E \quad k = 1, \ldots, \bar{L} \\
& t = 2, \ldots, T
\]

\[
\sum_{\tau=1}^{T_m} \tau \bar{x}_{k1 \tau}^B - 1 \leq T^m \bar{u}_{k1} \quad k = 1, \ldots, \bar{L}
\]

More redundancy can be avoided if we enforce that within a macro–period the unused slots are the last ones:

\[
\bar{u}_{ks} \geq \bar{u}_{k,s+1} \quad k = 1, \ldots, \bar{L} \\
& t = 1, \ldots, T \\
& s = (t-1)\bar{S} + 1, \ldots, t \cdot \bar{S} - 1
\]

Without loss of generality, we can assume that the time needed to set up a tank is an integer multiple of the time of a micro–period. Hence, the start and end of a slot must be scheduled so that this window equals exactly the time needed to set the tank up:

\[
\begin{align*}
C^m \bar{u}_{ks} & + \sum_{\tau=(t-1)T^m+1}^{tT^m} C^m \tau \bar{x}_{k\tau}^E \\
& - \sum_{\tau=(t-1)T^m+1}^{tT^m} C^m \tau \bar{x}_{k\tau}^B = \bar{\sigma}_{ks} & k = 1, \ldots, \bar{L} \\
& t = 1, \ldots, T \\
& s = (t-1)\bar{S} + 2, \ldots, t \cdot \bar{S}
\end{align*}
\]

\[
\begin{align*}
C^m \bar{u}_{k,(t-1)S+1} & + \sum_{\tau=(t-1)T^m+1}^{tT^m} C^m \tau \bar{x}_{k(t-1)S+1 \tau}^E \\
& - \sum_{\tau=(t-2)T^m+1}^{tT^m} C^m \tau \bar{x}_{k(t-1)S+1 \tau}^B = \bar{\sigma}_{k(t-1)S+1} & k = 1, \ldots, \bar{L} \\
& t = 2, \ldots, T
\end{align*}
\]

\[
\begin{align*}
C^m \bar{u}_{k1} & + \sum_{\tau=1}^{T_m} C^m \tau \bar{x}_{k1 \tau}^E \\
& - \sum_{\tau=1}^{T_m} C^m \tau \bar{x}_{k1 \tau}^B = \bar{\sigma}_{k1} - \bar{\omega}_{k1} & k = 1, \ldots, \bar{L}
\end{align*}
\]
Finally, a tank can only be set up again if it is empty:

\[
\sum_{j=1}^{J} I_{jk,\tau-1} \leq Q_k((1 - \bar{x}_{ks\tau}) + (1 - \bar{u}_{ks})) \\
\sum_{j=1}^{J} \bar{I}_{jk,\tau-1} \leq Q_k((1 - \bar{x}_{ks\tau}) + (1 - \bar{u}_{ks}))
\]

\begin{align}
J &= \{2, 3, 4\} \text{ and } \bar{J} = \{1, 2\} \\
L &= \{2, 3, 4\} \text{ and } \bar{L} = \{2, 3\} \\
J &= \{2, 3, 4\} \text{ and } \bar{J} = \{1, 2\} \\
S &= \{2\} \text{ and } \bar{S} = \{2\} \\
T &= \{1, 2, 3, 4\}, T^m = 5 \text{ and } C^m = 1h.
\end{align}

<table>
<thead>
<tr>
<th>L</th>
<th>\bar{L}</th>
<th>J</th>
<th>\bar{J}</th>
<th>S</th>
<th>\bar{S}</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1: Parameter combination for each set of instances

Computational Results

The lack of similar models in the literature led us to create a set of instances for the SITLSP. The parameters adopted are:

- \( L = \{2, 3, 4\} \text{ and } \bar{L} = \{2, 3\} \).
- \( J = \{2, 3, 4\} \text{ and } \bar{J} = \{1, 2\} \).
- \( S = \{2\} \text{ and } \bar{S} = \{2\} \).
- \( T = \{1, 2, 3, 4\}, T^m = 5 \text{ and } C^m = 1h. \)
- \( minDem = 500u \text{ and } maxDem = 10000u, \ u: \text{ product unity}. \)
- \( h_j = \bar{h}_j = 1 (\$/u). \)
- \( v_{jl} = \bar{v}_{jk} = 1 (\$/u). \)
- \( \bar{Q}_k = 5000l \text{ and } Q_k = 1000l, \ l: \text{ liter of raw material}. \)

A certain combination of parameters \( L, \bar{L}, J, \bar{J}, S \) and \( \bar{S} \) defines a set of instances. These sets are considered of small-to-moderate sizes if compared to the industrial instances found in soft drink companies. Table 1 presents the values used in each combination. Observe that there are 9 possible combinations for each macro–period \( T = 1, 2, 3, 4. \) For
each one of these sets, 10 replications are randomly generated, resulting in $4.9 \times 10 = 360$ instances in total. The other parameters of the model are generated from a uniform distribution in the interval $[a, b]$, as shown in Table 2.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Ranges</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma'_{ij}$</td>
<td>$U[0.5, 1]$</td>
<td>line setup time</td>
</tr>
<tr>
<td>$\sigma'_{ijk}$</td>
<td>$U[1, 2]$</td>
<td>tank setup time</td>
</tr>
<tr>
<td>$p_{ji}$</td>
<td>$U[1000, 2000]$</td>
<td>process time</td>
</tr>
<tr>
<td>$r_{ji}$</td>
<td>$U[0.3, 3]$</td>
<td>conversion factor</td>
</tr>
</tbody>
</table>

Table 2: Parameter ranges

The computational tests were run on a Pentium IV microcomputer with 2.8 GHz. Initially, the instances were coded using the modeling language GAMS IDE (version 2.0.10.0) and solved by the solver CPLEX (version 7). The GAMS/CPLEX ran over each instance once during the time limit of 1 hour. Two kinds of problem solutions were returned: the optimal solution or the best feasible solution achieved up to the time limit.

Table 3 presents the number of constraints and variables (binaries and continuous) for each set of instances $L/L/J/J$ with $T = 1$.

<table>
<thead>
<tr>
<th>$L/L/J/J$</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Var.</td>
<td>Continuous Var</td>
</tr>
<tr>
<td>2 / 2 / 2 / 1</td>
<td>100</td>
</tr>
<tr>
<td>2 / 2 / 3 / 2</td>
<td>136</td>
</tr>
<tr>
<td>2 / 2 / 4 / 2</td>
<td>142</td>
</tr>
<tr>
<td>3 / 2 / 2 / 1</td>
<td>150</td>
</tr>
<tr>
<td>3 / 2 / 3 / 2</td>
<td>204</td>
</tr>
<tr>
<td>3 / 2 / 4 / 2</td>
<td>213</td>
</tr>
<tr>
<td>4 / 2 / 2 / 1</td>
<td>176</td>
</tr>
<tr>
<td>4 / 2 / 3 / 2</td>
<td>246</td>
</tr>
<tr>
<td>4 / 2 / 4 / 2</td>
<td>258</td>
</tr>
</tbody>
</table>

Table 3: Model statistics for the set of instances with $T = 1$

The following deviation value was used to compare the solutions:

$$Dev(\%) = 100 \frac{(Z - \bar{Z})}{\bar{Z}}$$

where $Z$ is the final solution value returned by GAMS/CPLEX and $\bar{Z}$ is the lower bound value also determined by GAMS/CPLEX. Table 4 shows the GAMS/CPLEX results for the set of instances with $T = 1$.

Each combination $L/L/J/J$ represents a set containing 10 instances. The second column shows the number of optimal solutions found among the 10 instances evaluated. The average (Avg), maximum (Max), minimum (Min) values and the standard deviation (Std) of Dev and CPU are also listed in the table. Note in the table that the runtime limit of 3600 seconds was reached for certain instances (see column Max), however, GAMS/CPLEX was able to solve most of the problem instances optimally with $T = 1$ (77 out of 90 instances) within this time limit.
Table 4: GAMS/CPLEX results for set of instances with $T = 1$

| $L/J/J$ | Optimal Solutions | Dev(%) | CPU(s) |  |
|---------|------------------|--------|--------|
|         |                  | Avg    | Max    | Min    | Std  | Avg   | Max   | Min   | Std  |
| 2 / 2 / 1 | 10               | 0.00   | 0.00   | 0.00   | 0.00 | 2.08  | 10.22 | 0.09  | 3.14 |
| 2 / 2 / 1 | 10               | 0.00   | 0.00   | 0.00   | 0.00 | 0.96  | 4.06  | 0.09  | 1.25 |
| 2 / 2 / 1 | 10               | 0.00   | 0.00   | 0.00   | 0.00 | 0.77  | 2.11  | 0.22  | 0.61 |
| 2 / 2 / 1 | 10               | 0.00   | 0.00   | 0.00   | 0.00 | 573.32 | 2323.84 | 2.53  | 839.98 |
| 2 / 2 / 1 | 2                | 1.13   | 3.23   | 0.00   | 1.11 | 3068.64 | 3600  | 10.73 | 1203.42 |
| 2 / 2 / 1 | 8                | 0.55   | 3.03   | 0.00   | 1.17 | 989.39 | 3600  | 0.78  | 1474.24 |
| 2 / 2 / 1 | 7                | 0.80   | 3.05   | 0.00   | 1.29 | 1356.72 | 3600  | 18.39 | 1621.32 |

Table 5: Model statistics for the set of instances with $T = 2$

<table>
<thead>
<tr>
<th>$L/J/J$</th>
<th>Binary Var.</th>
<th>Continuous Var.</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 / 2 / 1</td>
<td>210</td>
<td>533</td>
<td>575</td>
</tr>
<tr>
<td>2 / 2 / 1</td>
<td>282</td>
<td>1004</td>
<td>919</td>
</tr>
<tr>
<td>2 / 2 / 1</td>
<td>294</td>
<td>1235</td>
<td>1039</td>
</tr>
<tr>
<td>3 / 2 / 1</td>
<td>315</td>
<td>916</td>
<td>862</td>
</tr>
<tr>
<td>3 / 2 / 1</td>
<td>423</td>
<td>1771</td>
<td>1368</td>
</tr>
<tr>
<td>3 / 2 / 1</td>
<td>441</td>
<td>2206</td>
<td>1552</td>
</tr>
<tr>
<td>4 / 2 / 1</td>
<td>367</td>
<td>1122</td>
<td>1013</td>
</tr>
<tr>
<td>4 / 2 / 1</td>
<td>507</td>
<td>2211</td>
<td>1641</td>
</tr>
<tr>
<td>4 / 2 / 1</td>
<td>531</td>
<td>2790</td>
<td>1865</td>
</tr>
</tbody>
</table>

Table 6: GAMS/CPLEX results for the set of instances with $T = 2$

| $L/J/J$ | Optimal Solutions | Dev(%) | CPU(s) |  |
|---------|------------------|--------|--------|
|         |                  | Avg    | Max    | Min    | Std  | Avg   | Max   | Min   | Std  |
| 2 / 2 / 1 | 10               | 0.00   | 0.00   | 0.00   | 0.00 | 28.27 | 107.97 | 1.02  | 34.30 |
| 2 / 2 / 1 | 6                | 1.50   | 7.42   | 0.00   | 2.51 | 1703.08 | 3600  | 17.09 | 1697.02 |
| 2 / 2 / 1 | 6                | 1.02   | 4.49   | 0.00   | 1.54 | 1545.06 | 3600  | 6.31  | 1775.13 |
| 3 / 2 / 2 | 1                | 0.82   | 1.71   | 0.00   | 0.52 | 3365.86 | 3600  | 1258.02 | 740.62 |
| 3 / 2 / 2 | 0                | 3.15   | 11.75  | 0.21   | 3.66 | 3600  | 3600  | 3600  | 0.0  |
| 3 / 2 / 2 | 1                | 2.21   | 7.59   | 0.00   | 2.21 | 3260.23 | 3600  | 201.67 | 1074.67 |
| 4 / 2 / 2 | 0                | 1.23   | 2.41   | 0.27   | 0.78 | 3600  | 3600  | 3600  | 0.0  |
| 4 / 2 / 2 | 1                | 1.07   | 3.48   | 0.00   | 1.01 | 3254.55 | 3600  | 144.70 | 1092.69 |
| 4 / 2 / 2 | 0                | 4.23   | 17.45  | 0.62   | 4.99 | 3600  | 3600  | 3600  | 0.0  |

Table 6: GAMS/CPLEX results for the set of instances with $T = 2$

Table 5 shows the model statistics for the set of instances with $T = 2$. Table 6 presents the corresponding GAMS/CPLEX results obtained for $T = 2$. The GAMS/CPLEX starts having difficulties in finding an optimal solution within the time limit for this set - note that only 25 out of 90 instances were optimally solved. Moreover, the required average CPU runtimes largely increase. Most of the solutions obtained consist of the best feasible
solutions found by the branch & cut algorithm up to the time limit. This is expected in some sense because of the increase in the number of constraints and variables of the model for each instance (table 5).

This behavior becomes worse for the set of instances with \( T = 3, 4 \), whose results are presented in tables 7 and 8. The GAMS/Cplex was unable to find optimal solutions in most of these problem instances within 1 hour. The increase in the deviation values show us that the quality of the best feasible solution obtained becomes worse than in the previous sets of instances. Tables 9 and 10 show the model statistics for the sets of instances with \( T = 3, 4 \).

The complexity of the model proposed here, associated with the high number of variables and constraints in the sets of instances with \( T = 3, 4 \), explains the hardness found by the branch & cut algorithm to determine optimal solutions during the time limit of 1 hour. Figure 3 uses the set of instances 4/2/4/2 as an example to illustrate the number of variables and constraints when \( T = 1, 2, 3, 4 \). Observe that the number of binary variables reaches something near 1000 in the last macro–period. The number of continuous variables starts around 1000 when \( T = 1 \) and it is greater than 5000 when \( T = 4 \). The number of constraints is near 1000 for \( T = 1 \) and almost 4000 for \( T = 4 \).

<table>
<thead>
<tr>
<th>( L/L/J/J )</th>
<th>Optimal Solutions</th>
<th>Dev(%)</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Avg</td>
<td>Max</td>
</tr>
<tr>
<td>2 / 2 / 2 / 1</td>
<td>2</td>
<td>1.23</td>
<td>2.69</td>
</tr>
<tr>
<td>2 / 2 / 3 / 2</td>
<td>3</td>
<td>2.37</td>
<td>6.65</td>
</tr>
<tr>
<td>2 / 2 / 4 / 2</td>
<td>3</td>
<td>6.07</td>
<td>20.09</td>
</tr>
<tr>
<td>3 / 2 / 2 / 1</td>
<td>0</td>
<td>1.49</td>
<td>2.89</td>
</tr>
<tr>
<td>3 / 2 / 3 / 2</td>
<td>0</td>
<td>5.79</td>
<td>14.81</td>
</tr>
<tr>
<td>3 / 2 / 4 / 2</td>
<td>0</td>
<td>4.70</td>
<td>13.22</td>
</tr>
<tr>
<td>4 / 2 / 2 / 1</td>
<td>1</td>
<td>1.54</td>
<td>3.69</td>
</tr>
<tr>
<td>4 / 2 / 3 / 2</td>
<td>0</td>
<td>9.41</td>
<td>29.44</td>
</tr>
<tr>
<td>4 / 2 / 4 / 2</td>
<td>0</td>
<td>8.23</td>
<td>14.08</td>
</tr>
</tbody>
</table>

Table 7: GAMS/Cplex results for the set of instances with \( T = 3 \)

<table>
<thead>
<tr>
<th>( L/L/J/J )</th>
<th>Optimal Solutions</th>
<th>Dev(%)</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Avg</td>
<td>Max</td>
</tr>
<tr>
<td>2/2/2/1</td>
<td>1</td>
<td>1.50</td>
<td>3.34</td>
</tr>
<tr>
<td>2/2/3/2</td>
<td>1</td>
<td>5.36</td>
<td>11.75</td>
</tr>
<tr>
<td>2/2/4/2</td>
<td>0</td>
<td>4.29</td>
<td>15.84</td>
</tr>
<tr>
<td>3/2/2/1</td>
<td>0</td>
<td>1.71</td>
<td>2.92</td>
</tr>
<tr>
<td>3/2/3/2</td>
<td>0</td>
<td>8.25</td>
<td>22.32</td>
</tr>
<tr>
<td>3/2/4/2</td>
<td>0</td>
<td>8.54</td>
<td>15.75</td>
</tr>
<tr>
<td>4/2/2/1</td>
<td>0</td>
<td>1.71</td>
<td>3.12</td>
</tr>
<tr>
<td>4/2/3/2</td>
<td>0</td>
<td>8.79</td>
<td>19.21</td>
</tr>
<tr>
<td>4/2/4/2</td>
<td>0</td>
<td>11.02</td>
<td>20.78</td>
</tr>
</tbody>
</table>

Table 8: GAMS/Cplex results for the set of instances with \( T = 4 \)
Conclusion

In this paper we present a mathematical model able to describe an industrial problem found in some industrial settings, such as in soft drink companies. It is a two level production problem where, in each level, a lot sizing and scheduling problem with parallel machines, capacity constraints and sequence-dependent setup costs and times has to be solved. The solutions in each level are integrated because the decisions made in one level have consequences for the other. These solutions are also synchronized because
the scheduling in both levels have to be compatible throughout the time periods of the planning horizon. This problem was called a Synchronized and Integrated Two-Levels Lot Sizing and Scheduling Problem (SITLSP) here.

The first contribution of this work was to propose a new model to represent the synchronization and integration features in our problem. The constraints established are not new, but they are assembled the first time to describe lots and micro-periods simultaneously in a two levels integrated approach.

The model was evaluated here using instances considered of small-to-moderate sizes if compared to the industrial instances of soft drink companies. Our goal was to find solutions within 1 hour of computer runtime in each instance. The software GAMS/CPLEX was used to obtain optimal solutions or the best feasible solution at least. This exact solution approach was able to return the optimal solution only in small size instances \((T = 1, 2)\). In most part of the instances, the best feasible solution found was returned at the end of the time limit. It is worth mentioning that the CPLEX version used in the experiments (CPLEX 7.0) is out-of-date and the newest version (CPLEX 10.0) could result in better performances.

The mixed-integer linear formulation proposed is large and complex because of its vast set of parameters, variables and constraints. This became hard for the exact solution approach to find optimal solutions when the parameters of the model increases. An interesting perspective for future research is to develop effective heuristic methods to deal with this problem. Following this line of research, the difficulties found in determining exact solutions led us to develop a genetic algorithm to deal with such a problem. The results obtained using this approach are being compiled and they will be reported in a forthcoming paper.

Acknowledgement

We are indebted to Flávio A. M. Veronese from Companhia de Bebidas Ipiranga, Ribeirão Preto, Brazil, for his kind collaboration. This research was partially supported by FAPESP (grant no. 00/02609–2) and CNPq (grant no. 303956/2003-8).

A Symbols for Parameters and Decision Variables

A.1 General Parameters

\( T \) Number of macro–periods

\( C \) Capacity (in time units) within a macro–period

\( T^m \) Number of micro–periods per macro–period

\( C^m \) Capacity (in time units) within a micro–period \((C = T^m C^m)\)

\( \epsilon \) A very small positive, real–valued number

\( M \) A very large number
A.2 Parameters for the Production Lines

- $J$ Number of products
- $L$ Number of parallel lines
- $L_j$ Set of lines on which product $j$ can be produced
- $S$ Upper bound on the number of lots per macro-period, i.e. the number of slots
- $d_{jt}$ Demand for product $j$ at the end of macro-period $t$
- $s_{ijl}$ Sequence-dependent setup cost coefficient for a setup from product $i$ to $j$ on line $l$ ($s_{jjl} = 0$)
- $h_j$ Holding cost coefficient for having one unit of product $j$ in inventory at the end of a macro-period
- $v_{jl}$ Production cost coefficient for producing one unit of product $j$ on line $l$
- $p_{jl}$ Production time needed to produce one unit of product $j$ in line $l$ (assumption: $p_{jl} \leq C^m$)
- $x_{jl0} = 1$, if line $l$ is set up for product $j$ at the beginning of the planning horizon
- $\sigma'_{ijl}$ Setup time for setting line $l$ up from product $i$ to product $j$ ($\sigma'_{jjl} \leq C$; $\sigma'_{jjl} = 0$)
- $\omega_{l1}$ Setup time for the first slot on line $l$ in macro-period 1 that has already been performed before macro-period 1 starts (note: if $\omega_{l1} > 0$ then values $x_{jl1}$ are already known and these variables should be set accordingly)
- $I_{j0}$ Initial inventory of product $j$

A.3 Decision Variables for the Production Lines

- $z_{ijls} = 1$, if line $l$ is set up from product $i$ to $j$ at the beginning of slot $s$; 0, otherwise ($z_{ijls} \geq 0$ is sufficient)
- $I_{jt}$ Amount of product $j$ in inventory at the end of macro-period $t$
- $q_{jls}$ Number of products $j$ being produced on line $l$ in slot $s$
- $x_{jl} = 1$, if slot $s$ on line $l$ can be used to produce product $j$; 0, otherwise
- $u_{ls} = 1$, if a positive production amount is produced in slot $s$ in production line $l$ (0, otherwise)
- $\sigma_{ls}$ Setup time at line $l$ at the beginning of slot $s$
- $\omega_{lt}$ Setup time for the first slot on production line $l$ in macro-period $t$ that is scheduled at the end of macro-period $t - 1$
- $x_{lsr} = 1$, if lot $s$ (which belongs to a single macro-period $t$) on line $l$ ends in micro-period $\tau$
$x_{lts}^B = 1$, if lot $s$ (which belongs to a single macro-period $t$) on line $l$ begins in micro-period $\tau$

$\delta_{ls}$ Time in the first micro-period of a lot which is reserved for setup time and idle time

$q_{ij}^0$ Shortage of product $j$

### A.4 Parameters for the Tanks

- $\bar{J}$ Number of raw materials
- $\bar{L}$ Number of parallel tanks
- $\bar{L}_j$ Set of tanks in which raw material $j$ can be stored
- $\bar{S}$ Upper bound on the number of lots per macro-period, i.e. the number of slots
- $\bar{s}_{ijk}$ Sequence-dependent setup cost coefficient for a setup from raw material $i$ to $j$ in tank $l$
- $\bar{h}_j$ Holding cost coefficient for having one unit of raw material $j$ in inventory at the end of a macro-period
- $\bar{v}_{jk}$ Production cost coefficient for filling one unit of raw material $j$ in tank $k$
- $\bar{x}_{jk0} = 1$, if tank $k$ is set up for raw material $j$ at the beginning of the planning horizon
- $\bar{\sigma}_{ijk}'$ Setup time for setting tank $k$ up from raw material $i$ to raw material $j$ ($\bar{\sigma}_{ijk}' \leq C$) (w.l.o.g. $\bar{\sigma}_{ijk}'$ is an integer multiple of $C^m$)
- $\bar{\omega}_{k1}$ Setup time for the first slot on tank $k$ in macro-period 1 that has already been performed before macro-period 1 starts (note: if $\bar{\omega}_{k1} > 0$ then values $\bar{x}_{jk1}$ are already known and these variables should be set accordingly)
- $\bar{I}_{jk0}$ Initial inventory of raw material $j$ in tank $k$
- $\bar{Q}_{jk}$ Maximum amount to be filled in tank $k$
- $\bar{Q}_k$ Minimum amount to be filled in tank $k$

### A.5 Decision Variables for the Tanks

- $\bar{z}_{ijks} = 1$, if tank $k$ is set up from raw material $i$ to $j$ at the beginning of slot $s$; 0, otherwise ($\bar{z}_{ijks} \geq 0$ is sufficient)
- $\bar{I}_{jkr}$ Amount of raw material $j$ in tank $k$ at the end of micro-period $\tau$
- $\bar{q}_{jks}$ Amount of raw material $j$ being filled in tank $k$ in slot $s$
- $\bar{q}_{jks\tau}$ Amount of raw material $j$ being filled in tank $k$ in slot $s$ in micro-period $\tau$
- $\bar{x}_{jks} = 1$, if slot $s$ can be used to fill raw material $j$ in tank $k$; 0, otherwise
\( \bar{u}_{ks} = 1 \), if slot \( s \) is used to fill raw material in tank \( k \); 0, otherwise

\( \sigma_{ks} \) Setup time, i.e. the time for cleaning and filling the tank up, at tank \( k \) at the beginning of slot \( s \)

\( \bar{\omega}_{kt} \) Setup time, i.e. the time for cleaning and filling the tank up, for the first slot on tank \( k \) in macro-period \( t \) that is scheduled at the end of macro-period \( t - 1 \)

\( \bar{x}_{ks}\tau^E = 1 \), if lot \( s \) — a setup, i.e. cleaning and filling the tank up — (which belongs to a single macro-period \( t \)) at tank \( k \) ends in micro-period \( \tau \)

\( \bar{x}_{ks}\tau^B = 1 \), if lot \( s \) — a setup, i.e. cleaning and filling the tank up — (which belongs to a single macro-period \( t \)) at tank \( k \) begins in micro-period \( \tau \)

A.6 Parameters for Linking Production Lines and Tanks

\( r_{ji} \) Amount of raw material \( j \) to produce one unit of product \( i \)

A.7 Decision Variables for Linking Production Lines and Tanks

\( q_{kjl\tau} \) Amount of product \( j \), which is produced on line \( l \) in micro-period \( \tau \) and which belongs to lot \( s \), that uses raw material from tank \( k \)

References


