Explicit Modelling of Multi-Period Setup Times in Proportional Lot-Sizing Problem with Constant Capacity

Waldemar Kaczmarczyk (✉ waldek@agh.edu.pl)
AGH University of Science and Technology  https://orcid.org/0000-0002-9248-3665

Research Article

Keywords: Production, Inventory, Lot-sizing and scheduling, Mixed-integer programming, Heuristics

Posted Date: December 1st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1086310/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
Explicit Modelling of Multi-Period Setup Times in Proportional Lot-Sizing Problem with Constant Capacity

Waldemar Kaczmarczyk

Faculty of Management, AGH University of Science and Technology.

Corresponding author(s). E-mail(s): waldek@agh.edu.pl;

Abstract

The planning horizon of small bucket models is often divided into many fictitious micro-periods, with non-zero demand only in the last micro-period of each real (macro-)period. On the one hand, such models ensure schedules with short cycle times and low work-in-process inventory in multilevel systems; on the other, they make setup times that are longer than a single period more likely. This paper presents a new mixed-integer programming model for the case with setup operations that overlap multiple periods. The new model assumes that the capacity is constant in the whole planning horizon and explicitly determines the entire schedule of each changeover. Moreover, a two-level MIP heuristic is presented that uses model-specific cuts to fix a priori some minor decisions. The results of the computational experiments show that the new model and MIP heuristic require a substantially smaller computational effort from a standard MIP solver than the known models.

Keywords Production · Inventory · Lot-sizing and scheduling · Mixed-integer programming · Heuristics

MSC Classification: 90B30, 90C11
Explicit Modelling of Multi-Period Setup Times

1 Introduction

This paper addresses \textit{mixed-integer programming} (MIP) models of lot-sizing and scheduling problems for several products with deterministic dynamic demand on a single machine with a limited capacity. The models discussed are based on a model for the \textit{proportional lot-sizing and scheduling problem} (PLSP) proposed by Haase (1994) (see also Drexl and Haase (1995)). The PLSP is the most flexible small bucket model, as it allows for the processing of two products during a single period (one before and another one after the setup operation). Depending on the industry, the time bucket (period) may be a day or shift – sometimes even shorter (i.e., an integer fraction of a shift).

The standard PLSP assumes that each setup operation is executed within a single period; as a consequence, even the longest setup time must be shorter than the shortest period. This assumption has two significant disadvantages.

Firstly, long periods make it impossible to split production lots into many small transfer lots and, therefore, force long lead times and a high work-in-process inventory in a multilevel environment (Kaczmarczyk, 2020); i.e., in systems with dependent demand (e.g., Haase, 1994, pp. 15-18) or multi-stage flow lines (e.g., Stadtler and Sahling, 2013). In the general lot-sizing and scheduling problem (GLSP) (Fleischmann and Meyr, 1997) small buckets of variable length may be easily extended to enclose long set-up operations but do not allow for coordination of production between consecutive stages. Large bucket models like the capacitated lot-sizing problem (CLSP) force even longer lead times.

Secondly, long periods make it difficult to plan small lots, as they do not preserve a high utilisation of the period capacity. If period boundaries limit the execution of setup operations, then some good solutions may be cut off from the solution space. Suerie (2006) calculated the additional cost for a set of
random instances. For setup times equal to 40% of the period length, the total cost increment was 0.14-0.49%, depending on machine utilisation. For setup times equal to 80% of the period length, the cost increment was even 5-8.37%.

Therefore, to ensure a high-quality solution, small bucket models are often applied for a **micro-period calendar**; i.e., real periods (**macro-periods**; e.g., weeks) are split into several short fictitious **micro-periods** (e.g., days, shifts, or fractions of shifts) with non-zero demand only at the end of the last micro-period of each macro-period (e.g., Karmarkar and Schrage, 1985; Fleischmann, 1990; Suerie, 2006). In such models, setup times longer than a single period become more likely.

Only a few lot-sizing and scheduling models allow for multi-period setup operations. Cattrysse et al. (1993) and Blocher et al. (1999) adapted the discrete lot-sizing and scheduling problem (DLSP) to a case with setup times equal to multiples of a period. Haase (1994) was the first to propose a PLSP model formulation that allows for setup operations to overlap multiple periods. His model determines a setup schedule implicitly: the additional variable accumulates the time assigned to the setup operation during consecutive periods until it becomes equal to the setup time. Suerie (2006) proved that this model is not correct and proposed two new models (**POST1** and **POST2**) to refine Haase’s idea. **POST1** uses a binary variable to identify the periods during which the setup operation is executed but not finished. **POST2** is based on the formulation of the PLSP that explicitly calculates the division of each period between products.

In short-term planning, the length (capacity) of a single period may be very short; e.g., a fraction of a shift, primarily when real periods are split into fictitious micro-periods. In such a case, the whole calendar may consist of uniform periods (i.e., the capacity of the periods is constant during the whole
planning horizon). To satisfy such assumption duration of each maintenance operation and overtime must be integer multiple of period length. For such a case, Kaczmarczyk (2009) proposed three model formulations that explicitly determine the end of each setup operation for the known start time\(^1\).

Although multi-period setup operations have only been considered in a few research papers, their importance may grow in the future. Increases in the computational capability of enterprise information systems allow for more-detailed planning, among other planning based on a more detailed calendar. Thus, setup times may become longer in the future compared to the time buckets used in planning (Suerie, 2005b, 2006). A broad survey on lot-sizing and scheduling models can be found in Jans and Degraeve (2008) and Copil et al (2017).

The main objective of this paper is to find model formulations that will make it possible to solve practical examples with standard MIP methods. To achieve this goal, it is necessary to define the problem using variables that clearly describe crucial alternatives as well as to make the linear relaxation of the MIP problem as tight as possible. An effective solution of a variety of lot-sizing problems depends crucially on the development of tight formulations for the particular problem features occurring in practice (Belvaux and Wolsey, 2001; Pochet and Wolsey, 2006).

In this paper, I present the reformulation of a known explicit model for constant capacity presented in Kaczmarczyk (2009) by using a modified definition of two binary variables and the related constraints. The original intention was to simplify the model formulation (to adapt it to a more complex problem); however, the experiments show that this small change has a surprisingly

\(^1\)Unfortunately, this article was published in a little-known journal and remains unknown to the international Operations Research community.
high impact on the computational effort. Secondly, I propose an MIP heuristic, which uses additional cuts to make the model considerably easier to solve with a negligible increment of total cost in the optimal solution.

The next section presents the new model, including its parameters and variables, valid inequalities, and heuristic. Section 3 describes the computational experiments, tools, data sets, and results. Section 4 gives a summary.

2 Models for uniform periods

The list below presents the notation: firstly, the basic parameters that define the PLSP; next, the derivative parameters necessary for the explicit description of the changeovers in the model; and finally, the continuous and binary variables.

Parameters:

\( \mathcal{T} = \{1, \ldots, T_{\text{max}}\} \) – set of periods;
\( \mathcal{N} = \{1, \ldots, N_{\text{max}}\} \) – set of products,
\( d_{jt} \) – demand of product \( j \) during period \( t \);
\( p_j \) – processing time of product \( j \);
\( I_{j0} \) – initial inventory of product \( j \);
\( h_{jt} \) – unit holding cost of product \( j \) during period \( t \);
\( sc_j \) – setup cost of product \( j \);
\( st_j \) – setup time of product \( j \);
\( C \) – capacity (length) of single period;

Derivative parameters:

\( Q_j = \lfloor st_j/C \rfloor \) is (integer) quotient of setup time of product \( j \) and capacity;
\( \mathcal{N}^0 = \{j \in \mathcal{N} : Q_j = 0\} \) – products with set-up times shorter than period;
\( \mathcal{N}^1 = \{j \in \mathcal{N} : Q_j \geq 1\} \) – products with set-up times not shorter than period;
Explicit Modelling of Multi-Period Setup Times . . .

\[ R_j = st_j - Q_j C \]

is the remainder of setup time of product \( j \) divided by capacity; however, if setup time is integer multiply of capacity (\( Q_j \geq 1 \) and \( st_j = Q_j C \)), then set \( Q_j = \lfloor st_j / C \rfloor - 1 \) and \( R_j = C \);

\[ \mathcal{P}_j = \{-Q_j, \ldots, 0\} \cup \mathcal{T} \] – set of periods necessary to describe the unfinished setup started in the past.

Continuous variables:

\( I_{jt} \) – inventory of product \( j \) at end of period \( t \);
\( x_{jt} \) – production volume of product \( j \) during period \( t \);
\( s_{jt} \) – time used to set up machine for product \( j \) during period \( t \) (but only when \( t \) is last period overlapped by setup operation);
\( s'_{jt} \) – time used to setup machine during any period \( t \) overlapped by setup operation,
\( b_{jt} \) – time during period \( t \) reserved for product \( j \) before changeover;
\( a_{jt} \) – time during period \( t \) reserved for product \( j \) after changeover;

Binary variables:

\( z'_{jt} = 1 \) if, during period \( t \), machine starts up to process setup operation for product \( j \); 0 otherwise;
\( v'_{jt} = 1 \) if, during period \( t \), \( z'_{jt} = 1 \) and time reserved for product \( j \) (to process setup operation) is longer than remainder of setup time (\( a_{jt} \geq R_j \)); 0 otherwise;
\( z_{jt} = 1 \) if, during period \( t \), machine starts up to produce product \( j \) (i.e., setup operation for this product is finished); 0 otherwise;
\( v_{jt} = 1 \) if, during period \( t \), \( z_{jt} = 1 \) and time used to process setup operation is longer than remainder of setup time (\( s_{jt} \geq R_j \)); 0 otherwise;
\( y_{jt} = 1 \) if, at end of period \( t \), machine is set up to process product \( j \) (i.e., machine is reserved [ready] for that product); 0 otherwise;
\( w_{jt} = 1 \) if, during period \( t \), processing of product \( j \) is switched off; 0 otherwise;
Explicit Modelling of Multi-Period Setup Times

$u_{jt} = 1$ if, during period $t$, setup operation of product $j$ is processed but not finished (i.e., it must be *continued* during next period); 0 otherwise.

In the models described in this paper, all variables with period indices $t \leq 0$ or $t \geq T_{\text{max}} + 1$ (i.e., from the past or future planning horizon) are set to be equal to zero. There are only two exceptions; firstly, inventory variables $I_{j0}$ may represent non-zero initial inventories; and secondly, variables with $t \leq 0$ may define the initial state of the machine, i.e., determine which product may be produced at the beginning of the first period or describe the state of a setup operation started in the past. Variable $s'_{jt}$ is used only to describe schedules in the illustrative examples and not in the considered models. Variables $z'_{jt}$ and $v'_{jt}$ are used here to describe examples and introduce new valid inequalities for the model proposed by Kaczmarczyk (2009).

The basis for further considerations constitutes an explicit formulation of the PLSP model (denoted here as PLSP/E). Kimms and Drexl (1998) developed this for cases with multiple resources; i.e., for products that require several resources at the same time. It uses variables $b_{jt}$ and $a_{jt}$, which explicitly describe the division of capacity among consecutive lots (i.e., the time before and after beginning each setup operation).

Kaczmarczyk (2009) used this explicit formulation to develop a new model for cases with setup operations overlapping multiple periods (denoted as PLSP/E-MS/CF). Its basic idea is illustrated by the example in Table 1.

The setup time for product $j$ is $st_j = 350$, and the capacity is $C = 100$; i.e., the (integer) quotient of setup time and capacity is $Q_j = 3$, and the remainder of that division is $R_j = 50$. There are two cases for the setup execution. If reserved capacity $a_{jf}$ in first overlapped period $f$ is longer or equal to the remainder of setup time $R_j$ (see Case a), then the setup operation overlaps the $Q + 1$ periods; otherwise, it overlaps the $Q + 2$ periods (see Case b). In
Table 1 Scenarios for setup operation started during period $f$ (Kaczmarczyk, 2009)

|   | $t$ | $f$ | $f + 1$ | $f + 2$ | $f + 3$ | $f + 4$ | $f + 5$ |
|---|-----|-----|---------|---------|---------|---------|---------|
| a) for $a_{jt} \geq R_j$; i.e., $v_{jt}' = 1$ | $a_{jt}$ | 70  | 100  | 100  | 80  | 20  | 100  |
|    | $s_{jt}$ | 70  | 100  | 100  | 80  | 20  | 100  |
|    | $x_{jt}$ | 20  | 100  | 100  | 20  | 80  | 100  |
|    | $z_{jt}'$ | 1   |       |       |       |       |       |
|    | $v_{jt}'$ | 1   |       |       |       |       |       |
|    | $u_{jt}$ | 1   | 1   | 1   | 1   | 1   | 1   |
|    | $y_{jt}$ | 1   | 1   | 1   | 1   | 1   |
| b) for $a_{jt} < R_j$; i.e., $v_{jt}' = 0$ | $a_{jt}$ | 30  | 100  | 100  | 100  | 20  | 80  |
|    | $s_{jt}$ | 30  | 100  | 100  | 100  | 20  | 80  |
|    | $x_{jt}$ | 100 | 100  | 20   |     |     |     |
|    | $z_{jt}'$ | 1   |       |       |       |       |       |
|    | $v_{jt}'$ | 0   |       |       |       |       |       |
|    | $u_{jt}$ | 1   | 1   | 1   | 1   | 1   | 1   |
|    | $y_{jt}$ | 1   | 1   | 1   | 1   | 1   |

...this model, the value of variable $a_{jf}$ is used to determine the whole setup schedule explicitly in later periods. Variable $z_{jt}'$ points to the beginning of the setup, and variable $v_{jt}'$ identifies the case. To set the complete schedule, one must determine the last period $l$ overlapped by the setup operation as well as the capacity $s_{jt}$ used to process it during this period. During time interval $[f, l - 1]$, production is prevented by binary variable $u_{jt}$, and during period $l$, the capacity available for production is decreased by $s_{jt}$.

The basic idea of the new model proposed in this paper is very similar, “symmetrical”. Variables $z_{jt}'$ and $v_{jt}'$ are here replaced by variables $z_{jt}$ and $v_{jt}$. Variable $z_{jt}$ points to the end of the setup (and not to the begin) and separates time intervals, during which production is prevented and enabled.

Suerie (2006) used such variable in his models. An example of a schedule is presented in Table 2.

The starting point is the capacity $s_{jt}$ used to set up the machine during the last period $l$ overlapped by the setup operation. When $s_{jt} \geq R_j$ (see Case a), then the setup overlaps the $Q + 1$ periods; otherwise, it overlaps the $Q + 2$ periods (see Case b). To complete the schedule of a setup operation, one must determine the first overlapped period $f$ and the capacity $a_{jf}$ used to process it during this period.
The new explicit formulation of the PLSP model with setup operations overlapping multiple periods (denoted as PLSP/E-MS/P) is presented below:

\[
\begin{align*}
\min & \quad \sum_{t \in T} \sum_{j \in N} (s_{cj} z_{jt} + h_{jt} I_{jt}), \\
I_{jt} + x_{jt} - d_{jt} & = I_{jt}, \quad j \in N, \ t \in T, \\
Cy_{jt} + b_{jt} & \geq p_{jt} x_{jt} + s_{jt}, \quad j \in N^1, \ t \in T, \\
C(y_{jt} - v_{jt}) + a_{jt} + b_{jt} & \geq p_{jt} x_{jt} + s_{jt}, \quad j \in N^0, \ t \in T, \\
C(1 - z_{jt} + v_{jt}) & \geq p_{jt} x_{jt} + s_{jt}, \quad j \in N^0, \ t \in T, \\
\sum_{j \in N} (b_{jt} + a_{jt}) & = \sum_{j \in N} C w_{jt}, \quad t \in T, \\
C w_{jt} & \geq b_{jt}, \quad j \in N, \ t \in T, \\
C(v_{jt} + Q_{jt} + z_{jt} + Q_{jt} + 1)
& - v_{jt} + Q_{jt} + 1 \geq a_{jt}, \quad j \in N, \ t \in P_{jt}, \\
\sum_{j \in N} (y_{jt} + u_{jt}) & = 1, \quad t \in T, \\
z_{jt} - w_{jt} & = y_{jt} - y_{jt-1}, \quad j \in N, \ t \in T.
\end{align*}
\]
Explicit Modelling of Multi-Period Setup Times...

\[ R_j v_{jt} \leq s_{jt}, \quad j \in \mathcal{N}, \ t \in \mathcal{T}, \quad (1.11) \]

\[ s_{jt} \leq C z_{jt}, \quad j \in \mathcal{N}, \ t \in \mathcal{T}, \quad (1.12) \]

\[ z_{j,t+1} - v_{j,t+1} \leq u_{jt}, \quad j \in \mathcal{N}^0, \ t \in \mathcal{P}_j, \quad (1.13) \]

\[ R_j(z_{jt} + v_{jt} - 1) \leq a_{jt}, \quad j \in \mathcal{N}^0, \ t \in \mathcal{P}_j, \quad (1.14) \]

\[ R_j(z_{j,t+1} - v_{j,t+1}) \leq a_{jt} + s_{j,t+1}, \quad j \in \mathcal{N}^0, \ t \in \mathcal{P}_j, \quad (1.15) \]

\[ \sum_{r=1}^{Q_j+1} z_{j,t+r} - v_{j,t+Q_j+1} \leq u_{jt}, \quad j \in \mathcal{N}^1, \ t \in \mathcal{P}_j \quad (1.16) \]

\[ (R_j + C)(z_{j,t+Q_j} + v_{j,t+Q_j} - 1) \leq a_{jt} + s_{j,t+Q_j}, \quad j \in \mathcal{N}^1, \ t \in \mathcal{P}_j, \quad (1.17) \]

\[ R_j(z_{j,t+Q_j+1} - v_{j,t+Q_j+1}) \leq a_{jt} + s_{j,t+Q_j+1}, \quad j \in \mathcal{N}^1, \ t \in \mathcal{P}_j, \quad (1.18) \]

\[ I_{jt} \geq 0, \quad j \in \mathcal{N}, \ t \in \mathcal{T}, \quad (1.19) \]

\[ x_{jt}, a_{jt}, b_{jt}, s_{jt} \in [0, C], \quad j \in \mathcal{N}, \ t \in \mathcal{T}, \quad (1.20) \]

\[ w_{jt} \in [0, 1], \quad j \in \mathcal{N}, \ t \in \mathcal{T}, \quad (1.21) \]

\[ z_{jt}, v_{jt}, y_{jt}, u_{jt} \in \{0, 1\}, \quad j \in \mathcal{N}, \ t \in \mathcal{T}. \quad (1.22) \]

Objective (1.1) is intended to minimise the total setup and inventory holding costs. In models with fictitious micro-periods only the last micro-period within each macro-period has non-zero unit holding cost \( h_{jt} \). Equalities (1.2) ensure the balance of production, inventory, and demand.

Constraints (1.3)-(1.5) limit the production volume. In constraints (1.3) and (1.4), setup state variable \( y_{jt} \) determines the capacity assigned to product \( j \) during a period when no setup operation is started or finished. If the machine ends processing product \( j \) during period \( t \), then variable \( b_{jt} \) describes the capacity reserved for \( j \) (before the setup operation), while all other variables on the left side are equal to zero. Variable \( s_{jt} \) (on the right side of the inequality) reserves the necessary setup time.
For product $j$ with $st_j \leq C$ (or $Q_j = 0$), it is possible to execute the whole setup operation within the same period and start up the machine to produce $j$ at the end of that period. In constraint (1.4), variable $a_{jt}$ then describes the capacity reserved for $j$ (after the setup operation), and finish variable $v_{jt}$ cancels state variable $y_{jt}$.

Variable $a_{jt}$ is necessary in (1.4) only in the case when the whole setup is completed during period $t$. If the setup only starts in $t$ and must be finished in $t + 1$, $y_{jt} = b_{jt} = s_{jt} = 0$ and constraint (1.4) reduces to $a_{jt} \geq p_j x_{jt}$; i.e., a positive production value is enabled (though it should not). Constraint (1.5) prohibits such production.

Constraints (1.6)-(1.8) explicitly model the distribution of capacity among two products during each period in which a setup operation starts. Constraints (1.6) ensure that the whole period capacity is distributed among the products. Constraints (1.7) and (1.8) allow non-zero values of $b_{jt}$ and $a_{jt}$ only if the machine switches off ($w_{jt} = 1$) or starts up the processing of product $j$, respectively. The setup operation may start in some period $t$ in two cases; if it ends during period $t + Q_j$ (for $v_{j,t+Q_j} = 1$), or it ends during period $t + Q_j + 1$ (for $z_{j,t+Q_j+1} = 1$ and $v_{j,t+Q_j+1} = 0$).

The original formulation of the explicit PLSP model (Kimms and Drexl, 1998) is slightly different. Among others, the original $b_{jt}$ and $a_{jt}$ variables describe the production volumes and not the reserved capacity.

Constraints (1.9)-(1.10) set the values of the binary variables describing the state of the machine. Constraint (1.9) ensures the unique state of the machine at the end of each period. Constraint (1.10) forces start-up variables $z_{jt}$ and switch-off variables $w_{jt}$ to take a value of 1 after each change of the machine state. The case with $z_{jt} = w_{jt} = 1$ and $y_{jt} = y_{j,t-1} = 0$ is considered in Subsection 2.2.
Explicit Modelling of Multi-Period Setup Times . . .

Constraints (1.11)-(1.18) model the execution of the setup operations. During the last period overlapped by setup operation ($z_{jt} = 1$), constraint (1.11) ensures that finish variable $v_{jt}$ takes a value of 1 only if $s_{jt} \geq R_j$. In such a case, the setup operation overlaps only $Q_j + 1$ periods (Example a in Table 2). For the remainder $R_j = 0$, this constraint would be satisfied for any values of both of the $v_{jt}$ and $s_{jt}$ variables. To avoid this ambiguity, the remainder $R_j$ is always greater than zero by definition; i.e., if the setup time is an integer multiple of period capacity $C$, then $R_j = C$ and $Q_j = st_j/C - 1$.

Constraint (1.12) ensures that there is no setup time when $z_{jt} = 0$. Otherwise, it could be “profitable” to set $v_{jt} = 1$, as it would ensure some (infeasible) capacity for production according to (1.8).

Constraints (1.13)-(1.18) schedule the setup operation if the (whole) setup time is shorter than the period capacity; i.e., for $st_j \leq C$ (or $Q_j = 0$). In such a case, the setup operation may overlap two periods at most. If the setup operation is executed within a single period, (1.11) ensures the proper value of $s_{jt}$, and (1.14) reserves the appropriate capacity $a_{jt}$. If the setup operation overlaps two periods, then constraint (1.13) sets continuation variable $u_{jt}$ during the first period and (1.15) reserves capacity $a_{jt}$ for a given $s_{j,t+1}$.

Constraints (1.16)-(1.18) schedule the setup operation if the setup times are longer than one period; i.e., for $st_j > C$ (or $Q_j \geq 1$). Constraint (1.16) ensures that continuation variables $u_{jt}$ take a value of 1 during all periods fully utilised by the setup operation (during the $Q_j$ or $Q_j + 1$ periods, in Table 2 periods $l - 4, \ldots, l - 1$). According to (1.9), this is equivalent to $y_{jt} = 0$ and prevents production. Constraint (1.17) determines the time reserved for setup in period $t$ when it ends during period $t + Q_j$, and Constraint (1.18) when it ends during period $t + Q_j + 1$ (compare this with (1.8)).
If the initial state of the machine is not fixed (i.e., \( z_{j0} = y_{j0} = u_{j0} = 0 \) for each \( j \in \mathcal{N} \)), then the schedule must start with some setup operation. In such case, the following constraints ensure a feasible beginning of the schedule:

\[
\sum_{j \in \mathcal{N}} v_{j,1+Q_j} = 1, \quad (2.1)
\]

\[
v_{j,1+Q_j} \leq z_{j,1+Q_j}, \quad j \in \mathcal{N}, \quad (2.2)
\]

\[
b_{j1} = 0, \quad j \in \mathcal{N}. \quad (2.3)
\]

Constraints (2.1) and (2.2) force the start-up of any setup operation during the first period as well as its quick execution. Constraints (2.3) prohibit earlier production.

### 2.1 Valid inequalities

The valid inequalities presented below (denoted as ILB) determine the minimal inventory necessary at the end of period \( t - 1 \) if, during some time interval \([t, t + \Delta]\), the machine is never setup to produce product \( j \) (e.g., Barany et al., 1984; Belvaux and Wolsey, 2001; Pochet and Wolsey, 2006, pp. 217-220). They are added a priori to the model here.

\[
I_{j,t-1} \geq \sum_{\tau=t}^{t+\Delta} d_{j\tau} \left[ 1 - y_{j,t-1} - \sum_{k=t}^{\tau} z_{jk} \right], \quad (3)
\]

\[j \in \mathcal{N}, \ t \in \mathcal{T}, \ \Delta \in [0, T_{\text{max}} - t].\]

When real periods (macro-periods) are split into micro-periods with non-zero demand only at the end of the last micro-period of each macro-period, then inventory lower bound (3) should be replaced by constraints (4) (denoted here as ILBW), which are defined only for macroperiods (Kaczmarczyk, 2020).
Explicit Modelling of Multi-Period Setup Times…

Additional parameters in cases with macro- and micro-periods:

\( \mathcal{W} = \{1, \ldots, W_{\text{max}}\} \) – set of macro-periods; \( W_{\text{max}} \) – number of macro-periods;

\( \mathcal{T}_w = F(w), \ldots, L(w) \) – set of periods (micro-periods) in macro-period \( w \);

where \( F(w) \) and \( L(w) \) are the first and last period in macro-period \( w \),

\( Y_{jw} = 1 \) if machine is setup to process product \( j \) during macro-period \( w \);

0 otherwise.

Macro-period based inventory lower bound for the new model:

\[
Y_{jw} \in \{0, 1\}, \quad j \in \mathcal{N}, \ w \in \mathcal{W}; \quad (4.1)
\]

\[
Y_{jw} = y_{j,L(w-1)} + \sum_{t \in \mathcal{T}_w} z_{jt}, \quad j \in \mathcal{N}, \ w \in \mathcal{W}; \quad (4.2)
\]

\[
I_{j,L(w-1)} \geq \sum_{\tau=w}^{w+\Delta} d_{j\tau} \left[ 1 - \sum_{k=w}^{\tau} Y_{jk} \right], \quad j \in \mathcal{N}, \ w \in \mathcal{W}, \quad (4.3)
\]

Each constraint (4.3) may be converted into an equivalent (3) by substituting \( Y_{j\tau} \) according to (4.2). The number of inequalities (4.3) is smaller than inequalities (3), because (4.3) start and end only at the end of each macro-period and not in each micro-period (as with (3)).

In the old model (Kaczmarczyk, 2009), start-up variables \( z'_{jt} \) do not point to the first period during which production is allowed; therefore, constraint (4.2) must be replaced by the following:

\[
Y_{jw} = y_{j,L(w-1)} + \sum_{t \in \mathcal{T}_w \setminus \{L(w)\}} z'_{j,t-(Q_j+1)} - v'_{j,F(w)-(Q_j+1)} + v'_{j,L(w)-Q_j}, \quad (5)
\]

\( j \in \mathcal{N}, \ w \in \mathcal{W}, \)
The period index of variables $z'_{jt}$ and $v'_{jt}$ is shifted back by $Q_j + 1$ micro-periods, as they indicate the beginning of a setup operation. Any setup operation started during period $F(w) - (Q_j + 1)$ will be finished during macro-period $w$, when $v'_{j,F(w) - (Q_j + 1)} = 0$. Any setup operation started during period $L(w) - Q_j$ will be finished during macro-period $w$, when $v'_{j,L(w) - Q_j} = 1$.

### 2.2 Two setups during same period

In the considered model, two setup operations and a small production lot $j$ may be processed during the same single period $t$, although $y_{jt,t-1} = 0$ and $y_{jt} = 0$. This is possible because, in (1.4) and (1.3), the capacity is reserved not only by variable $y_{jt}$ but also by $b_{jt}$, which depends on $w_{jt}$ and not on $y_{jt}$. The old explicit model PLSP/E-MS/CF (Kaczmarczyk, 2009) contains constraint $w_{jt} \leq y_{j,t-1}$, which prohibits such schedules.

Let us see an example in Table 3. The setup operation for product $j$ has $Q_j = 0$ and $R_j = 40$. It starts during Period 1 with $a_{j1} = 30$ and finishes during Period 2 with $s_{j2} = 10$. During Period 2, the execution of the lot for product $j$ lasts $p_j x_{j2} = 60$. During the same Period 2, the setup operation for another product $k$ begins with $a_{k2} = 30$. This is possible for the following values of the binary variables: $z_{j2} = w_{j2} = u_{j1} = u_{k2} = 1$, while $y_{j1} = y_{j2} = 0$. This solution satisfies all constraints of Model (1).

Such solutions are likely only for products with sporadic demand that require rare and small lots; i.e., they cannot occur very often. Therefore, their impact on the optimal objective value should not be significant.

| $t$ | $z_{jt}$ | $v_{jt}$ | $u_{jt}$ | $y_{jt}$ | $w_{jt}$ | $a_{jt}$ | $b_{jt}$ | $s_{jt}$ | $p_j x_{j2}$ | $z_{kt}$ | $u_{kt}$ | $a_{kt}$ |
|-----|---------|---------|---------|---------|---------|---------|---------|---------|---------------|---------|---------|---------|
| 1   |         |         |         | 1       | 0       | 30      |         |         |               |         |         |         |
| 2   | 1       | 0       | 0       | 0       | 1       | 70      | 10      | 60      |               |         |         | 1       |
| 3   |         |         |         |         |         |         |         |         |               | 1       | 0       |         |
Explicit Modelling of Multi-Period Setup Times ...

2.3 MIP heuristics

The new (and old) explicit model creates an opportunity to develop simple MIP heuristics. Start-up variables $z_{jt}$ (or $z'_{jt}$) determine the most important decisions of when to switch the machine from one product to another. Variables $v_{jt}$ (or $v'_{jt}$) make only some fine-tuning. More precisely, they determine whether the setup operation overlaps one additional period or not. However, the binary variables leave some free decision space for continuous variables; i.e., the time devoted to the setup in the first or last overlapped period may be set equal to zero or to the total capacity of that period. When micro-periods get shorter, this tuning becomes smaller (i.e., less important). Therefore, I have developed three MIP heuristics that use heuristic cuts to set the values of $v_{jt}$ (or $v'_{jt}$). All three heuristics consist of two steps:

**Step 1.** Optimise the new (or old) explicit model with variables $v_{jt}$ (or $v'_{jt}$) fixed with the help of some heuristic cut.

**Step 2.** Optimise the explicit model again, now without the heuristic cuts but with start-up variables $z_{jt}$ fixed according to the solution obtained in **Step 1**.

The cut used in the first heuristic (denoted as $H_1$) selects values of variables so that each setup operation overlaps the smallest possible number of periods:

$$v_{jt} \geq z_{jt}, \quad j \in \mathcal{N}, \ t \in \{Q_j + 2, \ldots, T_{\text{max}}\}.$$  \hspace{1cm} (6)

The cut used in the second heuristic (denoted as $H_0$) selects values of variables so that each setup operation overlaps the greatest possible number of periods:

$$v_{jt} = 0, \quad j \in \mathcal{N}, \ t \in \{Q_j + 2, \ldots, T_{\text{max}}\}.$$  \hspace{1cm} (7)
The cut used in the third heuristic (denoted as HR) is an attempt to assign more likely values to variables $v_{jt}$ (or $v'_{jt}$). Assuming that each completion time of a setup operation during the last overlapped period has the same probability, if $R_j$ is smaller than $0.5 \, C$, then it is more likely that the setup operation will overlap only $Q_j + 1$ periods rather than $Q_j + 2$ periods. HR is implemented as the following constraints:

\begin{align}
  v_{jt} &\geq z_{jt}, \quad j \in \mathcal{N}: R_j \leq 0.5 \, C, \quad t \in \{Q_j + 2, \ldots, T_{\text{max}}\}; \tag{8.1} \\
  v_{jt} &= 0, \quad j \in \mathcal{N}: R_j > 0.5 \, C, \quad t \in \{Q_j + 2, \ldots, T_{\text{max}}\}. \tag{8.2}
\end{align}

At the beginning of the schedule, it is often difficult to satisfy the initial demand. Therefore, heuristics cuts are not applied during the $Q_j + 1$ initial periods to allow for the free optimisation of the first start-up variables.

### 3 Results of experiments

To assess the properties of the new explicit model formulation, I have executed the following experiments for two different data sets on three formulations of the PLSP model:

- **POST1** – implicit model proposed by Suerie (2006),  
- **E-MS/CF** – old explicit model made by Kaczmarczyk (2009),  
- **E-MS/P** – new explicit model.

The POST2 model was omitted because it is harder to solve when set-up times get longer than period length (Suerie, 2006; Kaczmarczyk, 2009). The two explicit models were also solved with the help of three heuristics:

- **H1** – described by (6),  
- **H0** – described by (7),  
- **HR** – described by (8).
All instances were solved with the help of the CPLEX 12.8.0 solver on an Intel Core i9-7900X processor with 3.3 GHz clock speed, 16 GB RAM, and 10 cores. All computations were performed with the standard-setting of the solver.

Two characteristics are used to assess the quality of the solutions. The objective gap [%] is the mean relative difference between the best objective value \( f \) obtained for the considered model and the best objective value \( f^* \) obtained for any model and number of micro-periods, i.e., \( \text{objective gap} [%] = (f - f^*)/f^* \times 100\% \). The MIP gap [%] is the mean relative difference between best objective value \( f \) for any feasible solution to the best lower bound \( \text{LB} \) obtained with the given model formulation; i.e., \( \text{MIP gap} [%] = (f - \text{LB})/f \times 100\% \).

The computational effort may be evaluated with help of following characteristics. Time [s] is the computation time, and CV is its coefficient of variability; i.e., the ratio of the standard deviation to the mean. LP gap is the relative difference between the best objective value \( f^* \) of any MIP model and objective value \( f^R \) of the LP relaxation of the considered model (obtained without any MIP heuristics or cuts generated by the solver), i.e., \( \text{LP gap} [%] = (f^* - f^R)/f^* \times 100\% \). Nodes is the number of iterations, Iter. is the number of iterations, and It. time [\( \mu \text{s} \)] is the mean time per single iteration.

### 3.1 Results for Suerie’s data set

Suerie (2006) prepared instances with 15 or 30 periods. The smaller instances can be now solved in a few seconds; therefore, they are not considered here. In 40 instances with 30 periods, there are 3 products. The ratios of the setup times and period capacity \( st/C \) are 0.4, 0.8, or 1.2. The cycle times calculated with the EOQ formula for average demand are within a range of [6.6, 8.1] periods.
Suerie distinguished instances with \textit{loose} utilisation (with demand at about 48\% of the capacity) and \textit{tight} utilisation (with demand at about 60\% of the capacity). Preparing the data, Suerie ignored the impact of setup times on utilisation. Therefore for short setup times, it is low, and for long setup times, it is high. In optimal solutions, the utilisation is within a range of 50-90\%.

In the instances prepared by Suerie (2006), the initial state of the machines is not fixed, so the new model was extended with constraints (2). The demand and unit holding cost is non-zero during all periods; therefore, all models were also extended with \textit{ILB} valid inequalities; i.e., the classic inventory lower bounds (3). For all models and all instances, the solver determined an optimal solution (without a time limit), i.e., the \textit{MIP} gap is zero. The average results for all models and heuristics are presented in Table 4.

The objective gap for the optimal solutions of \textit{E-MS/CF} is 0.002\% on average, as this model does not allow for the processing of two setups during the same period as described in Section 2.2.

| Cuts | Model | Gap [%] | Time [s] | Nodes | Iter. | It.time |
|------|-------|---------|----------|-------|-------|---------|
|      | post1 | –       | –        | 21    | 214.0 | 1.8     | 154.1   | 8123  | 18    |
| "   | E-MS/CF | 0.002   | –        | 44    | 36.6  | 1.0     | 18.1    | 1645  | 23    |
| "   | E-MS/P  | –       | –        | 24    | 11.6  | 1.0     | 7.0     | 619   | 25    |
| H0  | E-MS/CF | 6.023-2.831$^1$ | –  | 7    | 1.9   | 0.2     | 0.2     | 33    | 106   |
| "   | E-MS/P  | 6.546-0.976$^1$ | –  | 8    | 2.4   | 0.2     | 0.5     | 52    | 71    |
| H1  | E-MS/CF | 0.702-0.393$^1$ | –  | 30   | 5.2   | 0.6     | 3.7     | 290   | 23    |
| "   | E-MS/P  | 0.717-0.528$^1$ | –  | 19   | 3.1   | 0.9     | 1.8     | 157   | 46    |
| HR  | E-MS/CF | 0.229-0.183$^1$ | –  | 23   | 3.3   | 0.3     | 1.7     | 138   | 48    |
| "   | E-MS/P  | 0.203-0.068$^1$ | –  | 16   | 2.4   | 0.3     | 0.6     | 63    | 66    |

$^1$ "-" replaces zero.

Results after the first and second step of the heuristic.

The average optimisation time with explicit models is much shorter than with the \textit{POST1} model, by even 95\% for \textit{E-MS/P}. The number of nodes is 95\%
smaller, and the number of iterations is 92% smaller. The time for the new explicit model E-MS/P is 68% shorter than that of the old E-MS/CF. The LP gap of the new explicit model is a little worse than for the POST1 model but substantially better than for the old explicit model. So it can (partially) explain only the difference between the two explicit models.

The coefficient of variability for the POST1 model is much higher than for other models because only for this model solver needed more than five minutes to complete the search; for seven tight instances — four of them required even more than ten minutes, and one 29 minutes. A high CV value means that more time should be reserved for the calculations; e.g., the 95th percentile of the optimisation time for POST1 is equal to 831 seconds (assuming the normal distribution for simplicity). For E-MS/P, this is equal to 30 seconds, and for the HR heuristic with E-MS/P, this is only 4 seconds.

Kaczmarczyk (2009) solved the same instances with CPLEX 12.0, 1.66 MHz, 1 GB RAM, and 1 core. The average time for POST1 was 253 s; for E-MS/CF, it was 203 s. This means that the explicit model was solved 20% faster in 2009; ten years later, that difference has grown to 83%. It is now more than four times larger! This result is a bit counterintuitive; new solvers are much better than old ones (generating better cuts, among other things) and, therefore, should be less dependent on the model formulation. However, it turns out to be the opposite — the new solver makes better use of a tight explicit model.

HR is the best heuristic for both explicit models. Firstly, H0 and H1 determine solutions with higher objective gaps. Secondly, H0 failed to find solutions for five instances (with low workloads and setup times longer than the period length), and H1 needs more time, nodes, and iterations than HR. Therefore, H0 and H1 are omitted in further considerations.
Explicit Modelling of Multi-Period Setup Times

HR ensures excellent solution quality; for the E-MS/P model, the mean objective gap is equal to 0.068%, and the maximal is 0.87%. Its computation time is very short – 99% shorter than for the POST1 model and 79% shorter than for the exact E-MS/P; the CV is also very small. Heuristic HR provides better solutions in a shorter time with the new E-MS/P explicit model than with the old E-MS/CF model.

For each heuristic, the model in the second step of was always solved as a linear relaxation (without branching), and its computation time was negligible (less than 0.2 seconds).

Table 5 presents the impact of utilisation and setup length on computation time. When utilisation is high or the setup times are long, the benefits of the explicit model are greater. The new E-MS/P model required 67-98% less time than POST1 and 62-80% less time than the old E-MS/CF model on average.

| Model       | Utilisation | \(st_j/C\) |
|-------------|-------------|-------------|
|             | Loose       | Tight       | 0.4 | 0.8 | 1.2 | mix |
| POST1       | 64          | 364         | 16  | 534 | 183 | 123 |
| E-MS/CF     | 28          | 46          | 27  | 72  | 16  | 32  |
| E-MS/P      | 9           | 14          | 5   | 25  | 3   | 12  |

3.2 Results for data set with micro-periods

The second data set imitates the division of real periods into many short fictitious micro-periods, with non-zero demand only in the last micro-period of each macro-period. In such a case, it is more likely that a single setup operation will overlap multiple periods. The second data set enables extensive experiments thanks to its greater diversity of setup times and the high number of micro-periods.
In the second data set, there are eight macro-periods divided into 5, 10, 15, or 20 micro-periods. Week capacity is always 1200. In the basic calendar with five micro-periods, capacity $C$ is 240. There are five products with uniform processing times $p_j = 1$. Unit holding cost is $h_{jt} = 1$ in the last micro-period within each macro-period, and $h_{jt} = 0$ in all other micro-periods, i.e., the number of micro-periods does not have any impact on the inventory cost accounting. The setup of the machine during period zero is fixed ($\sum_{j \in N} y_{j0} = 1$); i.e., at the beginning of the first period, the machine is ready to produce some randomly chosen product.

There are three patterns of setup times: mixed, short, and long. In the first one, there is a mix of different setup times that are equal to 48, 96, 144, 192, and 240. For $C = 240$, these setup times constitute 0.2, 0.4, 0.6, 0.8, and 1.0 of $C$, respectively. The ratios of the setup times and period capacity $st/C$ for a greater number of micro-periods are presented in the Table 6. The setup-time quotients are within a range of $[0, 3]$; in each instance, the setup remainders are within a range of $[0.2, 1.0]$. Two other setup time patterns assume uniform setup times for all products equal to 96 or 192 (i.e., 0.4 or 0.8 of $C = 240$).

| $j$ | $st_j$ | 5  | 10 | 15 | 20 |
|-----|--------|----|----|----|----|
| 1   | 48     | 0.2| 0.4| 0.6| 0.8|
| 2   | 96     | 0.4| 0.8| 1.2| 1.6|
| 3   | 144    | 0.6| 1.2| 1.8| 2.4|
| 4   | 192    | 0.8| 1.6| 2.4| 3.2|
| 5   | 240    | 1.0| 2.0| 3.0| 4.0|

### Table 6 Ratios of setup times to capacity

| Micro-periods | 5  | 10 | 15 | 20 |
|---------------|----|----|----|----|
| $C$           | 240| 120| 80 | 60 |

For each setup time pattern, there is a different set of five demand scenarios generated with the following procedure. At first, average demand values $\overline{d}_j$
for all products $j$ were randomly drawn from a uniform distribution over an interval of $[10, 90]$ to create high- and low-demand products. Next, all macro-period demands $d_{jw}$ were randomly generated with the help of the gamma distribution, with a shape parameter equal to $\bar{d}_j$ and scale parameter equal to 1. Initial inventory values $I_{j0}$ were generated with the same distribution as the demand. Finally, the demand and initial inventory have been rescaled proportionally for all products in such a way as to keep the machine’s workload equal to 60%.

The setup time has been set so that the expected time devoted to the setup operations is about 20% of the total capacity. For the pattern with setup times $st_j = 96$, average cycle time $T_{\text{eoq}}$ (i.e., the time interval between lots of the same product) was assumed to be two macro-periods. If each of the five products has a setup operation every two macro-periods, then the average setup time per macro-period is $5 \times 0.5 \times 96 = 240$, which is equal to $0.2 \times 1200$. For $st_j = 192$, the cycle was assumed to be $T_{\text{eoq}} = 4$, which gives a total setup time of $5 \times 0.25 \times 193 = 240$. For the mixed pattern, this was $T_{\text{eoq}} = 3$, and the total setup time was $(48 + 96 + 144 + 192 + 240) \times 1/3 = 240$.

The setup cost was set to $sc_j = T_{\text{eoq}}^2 \left( \sum_{jt} d_{jt} - \sum_{j} I_{j0} \right)/W_{\text{max}}/2$, according to the EOQ formula. In this way, the total expected machine utilisation is 80% for each instance. In optimal solutions, the utilisation was within a range of 74.2-75.8% The design of the data set is based on experience with lot-sizing and scheduling in the electronics industry (Kaczmarczyk, 2011).

It should be noted that the value of the objective function does not depend on the number of fictitious micro-periods. Therefore, any solutions for all instances that differ only in their numbers of micro-periods can be directly compared with each other; i.e, their objective gaps are calculated for the same best value of $f^*$.
Problem instances in the second data set are small enough to be solved almost to optimality with all the considered models, to make it easier to compare their results; however, real cases may be larger. Firstly, the manufacturing system may be more complex, i.e., it may consist of a number of critical stages and in each stage may work several parallel machines. Secondly, the number of macro- and micro-periods may be higher.

A foundry that inspired this study has over a dozen moulding machines and four postprocessing stages with a much smaller number of stations. The planning horizon consists of about 20 working days with non-zero demand, and each of them is divided into three shifts. Each shift must be further split into fictitious micro-periods to enable small transfer lots because the storage capacity is roughly equal to the volume of a single hour production (it depends on the actual product mix). Such case would lead to 20 macro-periods, each divided into 24 micro-periods. Moreover one has to satisfy additional constraints, e.g., a mold has a limited service life, and its replacement is equivalent to a changeover (Suerie, 2005a).

Table 7 presents the average results for the data set with micro-periods. All of the computations were performed with a ten-minute time limit. Sol. is the number of instances for which the solver found some feasible solution within the time limit, and Opt. is the number of instances for which the solver completed the search; i.e., the MIP gap was equal to zero.

Firstly, the models with macro-period-based ILBW inventory lower bounds are easier to solve than with classic ILB micro-period-based bounds. The time-saving for POST1 is equal to 32%, for E-MS/CF – 37%, and for E-MS/P – 50%. The number of nodes is quite similar, and the number of iterations is even greater. However, the average time needed to execute a single iteration is more than two or three times less. For ILB, the MIP gap is higher, and there are
Table 7  Mean results for data set with micro-periods

| Cuts  | Model | Gap [%] | Sol. Opt. | Time [s] | Nodes | Iter. | It.time |
|-------|-------|---------|-----------|----------|--------|--------|---------|
|       |       | obj.    | MIP       | LP       | mean   | CV     | [10^3]  | [10^6]  | [μs]   |
| ILB   | POST1 | 1.66    | 2.79      | 11       | 58     | 32     | 339     | 1.3   | 21.7  | 5.8  | 66   |
|       | E-MS/CF | 2.83   | 3.54      | 41       | 55     | 27     | 374     | 1.5   | 14.3  | 4.3  | 108  |
|       | E-MS/P | 0.52    | 1.28      | 10       | 59     | 42     | 281     | 1.1   | 11.9  | 3.3  | 97   |
| ILBW  | POST1 | 0.87    | 1.60      | 22       | 60     | 46     | 228     | 0.9   | 22.5  | 9.1  | 29   |
|       | E-MS/CF | 1.07   | 2.45      | 27       | 60     | 48     | 234     | 1.0   | 21.8  | 7.9  | 26   |
|       | E-MS/P | 0.30    | 0.82      | 22       | 60     | 53     | 140     | 0.7   | 10.2  | 3.8  | 30   |
| ILBW, HR | E-MS/CF | 0.13-0.12 | 0.28    | 22       | 60     | 56     | 98      | 0.6   | 12.1  | 4.8  | 20   |
|       | E-MS/P | 0.07-0.04 | –        | 22       | 60     | 60     | 39      | 0.9   | 5.8   | 2.0  | 23   |

"–" replaces zero.

1Results after the first and second step of the heuristic.

some unsolved instances; this suggests that the difference would be greater for a higher time limit.

Secondly, the new E-MS/P model (with ILBW valid inequalities) is better in almost every respect than the other models – the solution quality is higher, and the computational effort is lower (e.g., for E-MS/P, the mean time is 38 and 40% shorter than for POST1 and E-MS/CF, respectively); only the mean time per iteration is worse.

Unlike the exact models (without the HR cut), both of the explicit models with heuristic cuts are optimally solved for almost each data instance. In this way, the HR heuristic ensured solutions of a higher quality in a shorter amount of time than with the exact models. The maximal objective gap for HR with the E-MS/P model was only 0.74%, with E-MS/CF being 1.84%. Thanks to the HR cut, the computation time for E-MS/CF was 55% shorter, and for E-MS/P – even 72%. The HR heuristic with the E-MS/P model provided solutions with a mean objective value that was 1.62% better than the POST1 model in 83% less time.

Figure 1 presents the computation time with ILBW inequalities for various setup time patterns. In each case, the new explicit model requires less effort
than with the other models. The HR heuristic always ensures an additional and substantial time-saving.

![Graphs showing mean computation time for data set with micro-periods](image)

**Fig. 1** Mean computation time [s] for data set with micro-periods

### 4 Summary

In this paper, a new mixed-integer programming (MIP) model of the proportional lot-sizing problem (PLSP) with setup operations overlapping multiple periods is proposed for a fixed period length (capacity). It is a reformulation of the model proposed by Kaczmarczyk (2009), which explicitly determines a schedule for each setup operation; i.e., for the known ending of the setup operation, the constraints directly point to its beginning. In the competitive implicit model (Suerie, 2006), the constraints ensure that the time reserved for a setup operation (cumulated over consecutive periods) is long enough.

The new formulation uses a modified set of binary variables that separates the periods during which production is allowed; this simplifies the constraints
and makes the model tighter. Moreover, the MIP heuristic that uses additional cuts to fix the minor decisions for both explicit models (the new one and the old one) is proposed.

Computational experiments confirmed that the new explicit model is significantly easier to solve with standard solver than the implicit model (Suerie, 2006) and old explicit formulation (Kaczmarczyk, 2009); i.e., the branch and bound algorithm needs less time as well as fewer nodes and iterations to find optimal solutions. Besides, the proposed heuristic significantly accelerates optimisation at a negligible increment of the total cost.

The valid inequalities for micro-period calendar (Kaczmarczyk, 2020) and the new explicit model for setup operations overlapping multiple periods enable practical applications of models with many short micro-periods (especially in multi-level systems) with dependent demand and several production stages. This opportunity should be examined first.

Next, new models should be adapted for cases with parallel machines and limited capacity of additional resources; e.g., workers executing setup operations (Tempelmeier and Buschkühl, 2008), an approximation of zero lead times (Stadtler, 2011; Stadtler and Sahling, 2013), and limited space for a work-in-process inventory.

Acknowledgments

This work has been supported by the Polish National Science Center (NCN, Research Grant #DEC-2013/11/B/ST8/04458) and by the AGH University of Science and Technology (Grant #11.11.200.324).

Data deposition information: No datasets have been used.
References

Barany I, van Roy TJ, Wolsey LA (1984) Uncapacitated lot-sizing: The convex hull of solutions. Mathematical Programming Studies 22:32–43

Belvaux G, Wolsey LA (2001) Modelling practical lot-sizing problems as mixed-integer programs. Manage Sci 47(7):993–1007

Blocher JD, Chand S, Sengupta K (1999) The changeover scheduling problem with time and cost considerations: Analytical results and a forward algorithm. Oper Res 47:559–569

Cattrysse D, Salomon M, Kuik R, et al (1993) A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup times. Manage Sci 39:477–486

Copil K, Wörbelauer M, Meyr H, et al (2017) Simultaneous lotsizing and scheduling problems: a classification and review of models. OR Spectrum 39(1):1–64

Drexl A, Haase K (1995) Proportional lotsizing and scheduling. Int J Prod Econ 40:73–87

Fleischmann B (1990) The discrete lot-sizing and scheduling problem. Eur J Oper Res 44(3):337–348

Fleischmann B, Meyr H (1997) The general lotsizing and scheduling problem. OR Spektrum 19(1):11–21

Haase K (1994) Lotsizing and scheduling for production planning. No. 408 in Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin
Jans R, Degraeve Z (2008) Modeling industrial lot sizing problems: a review. Int J Prod Res 46(6):1619–1643

Kaczmarczyk W (2009) Modelling multi-period set-up times in the proportional lot-sizing problem. Decis Mak Manuf Serv 3(1–2):15–35. https://doi.org/10.7494/dmms.2009.3.2.15

Kaczmarczyk W (2011) Proportional lot-sizing and scheduling problem with identical parallel machines. Int J Prod Res 49(9):2605–2623

Kaczmarczyk W (2020) Valid inequalities for proportional lot-sizing and scheduling problem with fictitious microperiods. Int J Prod Econ 219:236–247

Karmarkar US, Schrage L (1985) The deterministic dynamic product cycling problem. Oper Res 33(2):326–345

Kimms A, Drexl A (1998) Proportional lot sizing and scheduling: Some extensions. Networks 32(2):85–101

Pochet Y, Wolsey LA (2006) Production planning by mixed integer programming. Series in Operations Research and Financial Engineering, Springer, New York

Stadtler H (2011) Multi-level single machine lot-sizing and scheduling with zero lead times. Eur J Oper Res 209(3):241–252

Stadtler H, Sahling F (2013) A lot-sizing and scheduling model for multi-stage flow lines with zero lead times. Eur J Oper Res 225(3):404–419

Suerie C (2005a) Campaign planning in time-indexed model formulations. Int J Prod Res 43(1):49–66
Explicit Modelling of Multi-Period Setup Times

Suerie C (2005b) Time Continuity in Discrete Time Models. No. 552 in Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Heidelberg

Suerie C (2006) Modeling of period overlapping setup times. Eur J Oper Res 174(2):874–886

Tempelmeier H, Buschkühl L (2008) Dynamic multi-machine lotsizing and sequencing with simultaneous scheduling of a common setup resource. Int J Prod Econ 113(1):401–412
Declarations

Funding (see Acknowledgments)

Conflicts of interest/Competing interests (Not applicable)

Availability of data and material (Data available on request in CSV format)

Code availability (Not available)