Heuristic based on mathematical programming for a lot-sizing and scheduling problem in the the mold-injection production

Yasmín Á. Ríos Solís[1,*], Omar J. Ibarra Rojas

Universidad Autónoma de Nuevo Léon, Mexico

Edgar Possani, Marta Cabo

ITAM, Mexico

Abstract

This paper deals with a manufacturing setting where sets of pieces are assembled to form different products. Each piece may be processed in a set of molds with different production rates. Molds must be mounted on machines with their corresponding installation setup times. The objective is to maximize the profit of the assembled products. The key point of our methodology is a two phase iterative approach where the lot-size of the products is determined together with the mold-machine assignments but allowing mold overlapping (a mold may be used in more than one machine at the same time). Then, for each period, we determine if there is a feasible schedule of the molds with no overlapping. If unsuccessful, we go back to the first stage to slightly restrict the number of machines that a mold can visit, repeating both stages until a feasible schedule is found. We show the advantages of this methodology on randomly generated instances and on data from real companies. Experimental results show that our mathematical programming based heuristic converges to a feasible solution in only few iterations and is capable of finding high quality solutions in reasonable times.

Keywords: Lot-sizing, Scheduling, Mold-injection, Integer Linear Programming, Heuristic

1. Introduction

Plastic injection systems use molds mounted on machines for shaping raw plastic into pieces that are then assembled to form products. The injection molded plastic industry is booming since metals in various industrial applications are being replaced by light weight plastic pieces. Moreover, plastic injection offers flexible designing and low costs for mass production. Mold-injection manufacturing is used in various industries including automotive, consumer goods and toys, among others.

The planning process that the mold-injection manufacturing companies face is as follows (see Figure 1). The lot-size of a set of products must be determined by taking into account that each product requires a set of pieces to be assembled. Some pieces may be needed by more than one product, while others

*Corresponding author

Email addresses: yasmin.riossls@uanl.edu.mx (Yasmín Á. Ríos Solís), omar.ibarrarj@uanl.edu.mx (Omar J. Ibarra Rojas), epossani@itam.mx (Edgar Possani), marta.cabo@itam.mx (Marta Cabo)

are exclusive of a single one as illustrated by the product-pieces edges of Figure 1. Each piece has a set of molds that can produce it (piece-mold edges). Each mold may be different due to its technical specifications; of particular relevance is its number of cavities, that is, the number of pieces that are made at each injection cycle. As schemed in Figure 1, a mold can produce several types of pieces but only one type at the time, that is, first mold can produce a certain number of cycles of robot heads then another number of cycles of robot arms, but it cannot produce robot heads and arms in the same cycle. Finally, each mold can be mounted on a certain number of machines (mold-machines edges) to produce the plastic pieces; as molds, each machine has its own production rates.

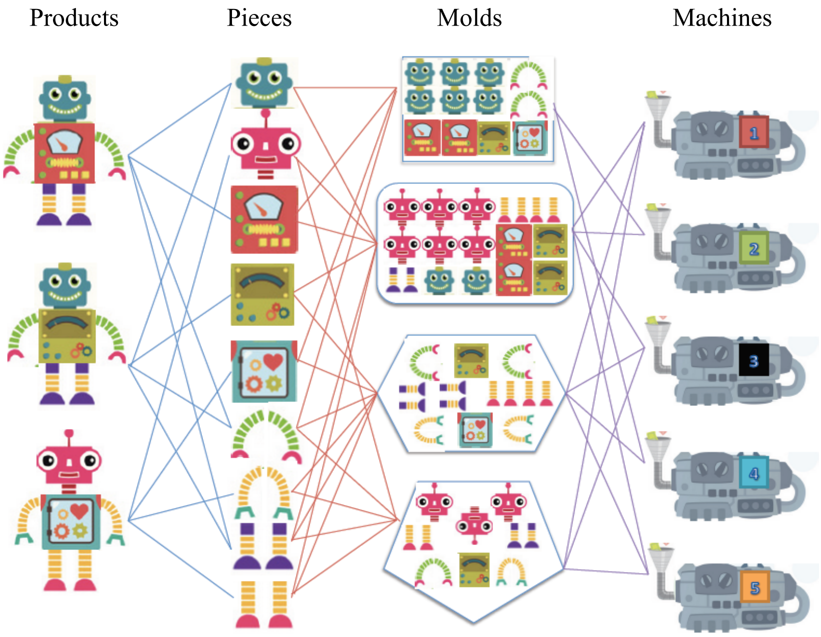

Figure 1: Mold-injection production scheme

Once a set of pieces has been produced on a machine, there are two options: to continue processing other pieces using the same mold, or to remove the mold from this machine to install a new one. Molds are heavy, and often need to be moved by cranes, therefore a mold change is an operation that requires a *mold-setup time*. In the Gantt chart of Figure 2 a feasible production plan for a two-period PPMM problem is presented. Mold 1 is installed on Machine 1, and pieces 1 and 6 are produced in it (we neglect the relatively small setup time between two different types of pieces). Then a mold change occurs, and Mold 2 is installed, with a mold-setup time. Note that Mold 1 is also used on Machine 2, but only after it has been released from Machine 1. Pieces of the same type may be produced simultaneously in two different machines using different molds. For example, pieces of type 2 are produced using Mold 4 on Machine 2, and at the same time using Mold 3 on Machine 3. The companies' objective is to determine the lot-size of products to maximize profit for several periods, considering constraints for assignments of pieces to molds and molds to machines.

Notice that it is possible to continue the production run from the previous period into the current

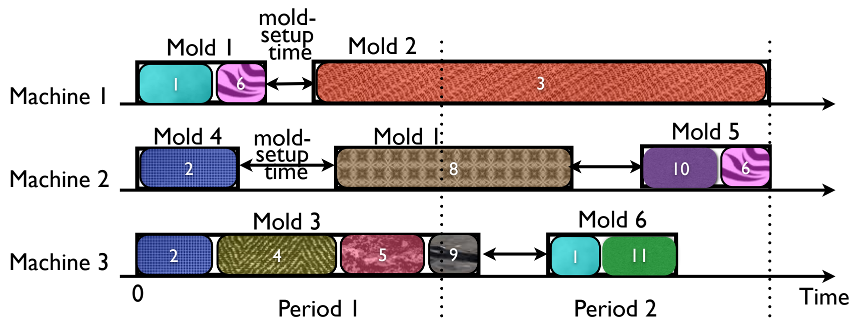

Figure 2: Gantt chart of the mold-injection production planning.

period without the need for an additional setup, that is, we face with a setup carry-over case. Demand for products is deterministic and time-varying but it may be not satisfied at all in one or more periods (lost sales case). We consider an inventory cost of holding pieces (not products) but we do not consider setup costs since we have noticed that companies cannot quantify them.

Therefore, the main planning problem is to determine the optimal lot-size of products by making an accurate production assignment of the pieces to molds and molds to machines in order to maximize the total value of the manufactured the products minus the inventory costs of the pieces. We refer to this problem as the Product-Piece-Mold-Machine (PPMM) problem.

The PPMM problem can be modeled by a quadratically constrained integer linear programming that does not yield feasible solutions after one hour computation. Thus, we propose a two phase iterative heuristic where we iteratively solve two integer linear programms of subproblems of PPMM but it yields high quality solutions as it is shown in the experimental results (Section 4).

As shown in Figure 3, our methodology consists in first determining the lot-size of the products along with the amount of time that each piece-mold-machine combination is used in order to maximize the product profit without the mold overlapping constraint, so a mold may be used in more than one machine at a time. Moreover, this relaxation named as the LS-MM model, is enhanced by some *mold constraints* that allow us to avoid a large number of infeasible solutions, that is, to avoid solutions with mold overlapping. In the second stage, an integer linear programming model determines the schedule for the partial solution given by LS-MM in each planning period such that a measure of work in process is optimized (SCHED stage). Straddling is allowed by a mold between two periods on at most one machine (see molds 3 and 8 in Figure 2). If a feasible schedule with no mold overlapping is found for all periods, it would be a feasible solution of the PPMM problem. Otherwise, for the overlapping molds a restriction is included in the LS-MM stage to slightly perturb the current solution. We iterate through these two stages until a feasible solution is found. With this procedure, which can be seen as a *heuristic cutting plane* algorithm, only few iterations are usually needed to reach a high quality solution of the PPMM problem in reasonable time.

There are three excellent reviews concerning lot-sizing and scheduling problems (Drexl & Kimms, 1997;

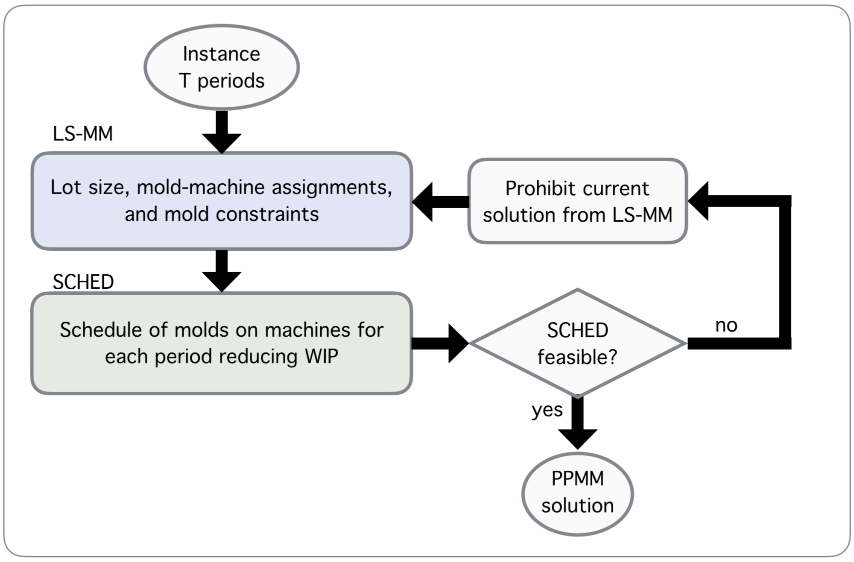

Figure 3: PPMM problem decomposition approach

Quadt & Kuhn, 2008), and Karimi et al. (2003). In none of them, the use of auxiliary equipment (as the molds in the PPMM problem) is mentioned. According to these reviews, the PPMM problem is a bi-level capacitated lot-sizing and scheduling problem that aims at determining the level of production complying with given capacity restrictions and such that demand for all products is maximized without backlogging and at the same time, determining the optimal timing of the molds on the machines to produce plastic pieces that form the final products. Moreover, many studies deal with problems considering a single machine (Stadtler, 2011; Haase & Kimms, 2000) thus, no mold overlapping considerations are needed.

There are few studies considering a complete integration of the lot-sizing and the scheduling stages as we do in this research. Kim et al. (2010) deal with a lot-sizing and scheduling problem that minimizes the sum of production cost, setup cost, inventory cost, and incorporate the constraints of setup carry-over and overlapping as well as demand splitting. They develop a mixed integer programming and decompose the problem by dividing time periods into a number of microtime periods. Meyr & Mann (2013) study the problem of determining the sizes and schedules of production lots on parallel, heterogeneous production lines with respect to scarce capacity, sequence-dependent setup times and deterministic, dynamic demand of multiple products. They propose a heuristic that iteratively decomposes the multi-line problem into a series of single-line problems. Fandel & Stammen-Hegene (2006) deal with the multi-level lot sizing and scheduling problem for job-shop production in capacitated, dynamic, and deterministic cases. The focus of this last study is on a big bucket model rather than in the resolution methodology. None of the just mentioned articles use of auxiliary equipment as we do. Moreover, our heuristic decomposition strategy is different since it is based on a relaxed solution which is then made feasible.

With respect to the use of auxiliary equipment, Chen & Wu (2006) deal with scheduling jobs on unrelated parallel machines where a setup for dies is incurred if there is a switch from processing one type of job to another type. The authors propose a heuristic based on threshold-accepting methods, tabu lists, and improvement procedures to minimize total tardiness. Lin et al. (2002a) study a production scheduling problem for making plastic molds of hi-fi models where the objective is to minimize the total machine makespan in the presence of due dates, variable lot size, multiple machine types, sequence dependent, machine dependent setup times, and inventory limits. Goal programming and load balancing are applied to select the set of machine types and assign mold types to machines, resulting in a set of single-machine scheduling problems. In Lin et al. (2002b), the auxiliary equipment is restricted to be installed in no more than one machine. Nevertheless, in the optimal solution, a mold may be installed in more than one machine even if this behavior is not intuitive.

Studies related with our proposed methodology are the following. Ouerfelli et al. (2009) study a periodical planning problem to determine the produced quantity and scheduling of machines. They present an integrated model that takes into consideration the scheduling constraints in the lot-sizing model and then solve it by a Benders' decomposition. In James & Almada-Lobo (2011) the authors combine metaheuristics and mixed integer programming to find high quality solutions to parallel-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times and costs. Finally, Beraldi et al. (2008) develop a rolling-horizon and fix-and-relax heuristics for the identical parallel machine lot-sizing and scheduling problem with sequence-dependent setup costs. In our case, the mathematical integer linear programming decomposition that we propose is used as a heuristic methodology.

In the injection molding production,Nagarur et al. (1997) consider a production planning and scheduling model for injection molding of PVC pipe fittings where the objective is to minimize the total costs of production, inventory, and shortages. Contrary to our integral methodology, a three-stage sequential procedure is used to solve their problem. Dastidar & Nagi (2005) address a production scheduling problem in an injection molding facility where the objective is to meet customer demands while minimizing the total inventory holding costs, backlogging costs and setup costs. They present a mathematical formulation of the problem and a two-phase workcenter-based decomposition scheme. Our mathematical formulation and decomposition scheme drastically differs from the one they present since ours considers more realistic characteristics. Finally, Ibarra-Rojas et al. (2011) propose a similar problem to the one we tackle, however, they focus on maximizing the production of pieces rather than products assembled by pieces and they focus only on feasibility in the scheduling problem. Our model is more realistic as it takes into account that companies usually sell products rather than spare pieces. Hence, even though they are set in a similar environment, their mathematical model and solution approach are not easily adapted to our case. Moreover, our model for the scheduling stage considers an objective function based on the assembly line and it allows idle times between the molds which is a characteristic that is essential for real cases and that had been overlooked by previous studies.

The rest of this article is structured as follows. In Section 2 we present the integer linear programming of the LS-MM stage while in Section 3 we introduce the SCHED stage. In Section 4 we present the experimental results that validate our methodology with generated data based on those from real companies. Finally, in Section 4.3 we compare the results obtained with the decomposition approach with those obtained by the companies. We summarize our conclusions in Section 5.

2. Lot-sizing and mold-machine assignments (LS-MM)

The phase I of our solution approach consist of solving the LS-MM optimization problem with a commercial solver. We describe the mathematical formulation of LS-MM in the following. Let I denote the set of products, and $I(j)$ the subset of products that need pieces of type j to be assembled. Let J be the set of all pieces from which products are made, $J(i)$ the subset of pieces needed to assemble product i, and $J(k)$ the subset of pieces that can be produced on mold k. Likewise, let K be the set of molds where pieces may be processed, and $K(j)$ the subset of molds where piece of type j may be produced. Finally, let L be the set of machines on which the molds are mounted, and $L(k)$ the set of machines where mold k may be mounted.

Let T denote the set of periods considered in the planning horizon. For each period $t \in T$, a product $i \in I$ has a profit of p_i^t per item and a demand of d_i^t. The number of pieces of type j needed to assemble one unit of i is denoted by a_{ij}. Moreover, a piece of type j can be manufactured using any mold $k \in K(j)$ with a number of cavities c_{jk}. For each period $t \in T$, machine l has an available production time f_l^t. The time it takes to manufacture a batch of piece j with mold $k \in K(j)$ installed on machine $l \in L(k)$ is denoted by b_{jkl}, this includes possible change of color or plastic type, as they are negligible in the mold injection setting. Installation and removal time, i.e., setup time of mold k on machine $l \in L(k)$, is denoted by s_{kl}. A piece of type j which is not used during period t is held as inventory for the next period and it yields an inventory cost of h_j^t.

Since LS-MM determines the lot-size and the piece-mold-machine assignments that are used in each planning period, we define the following decision variables.

- x_{jkl}^t: number of batches of piece j produced with mold k on machine l in period t.
- w_i^t: quantity of product i to be assembled in period t (lot-size of product i).
- r_j^t: number of pieces of type j not used at the end of planning period t. These pieces will be held as inventory for the next period (we consider $r_j^0 = 0$).
- y_{kl}^t: binary variable taking the value of 1 if mold k is used on machine l in period t, and 0 otherwise.
- z_{kl}^t: binary variable taking the value of 1 if mold k is straddling between period t and $t+1$ on machine l, and 0 otherwise.

We now state the MILP for the LS-MM as follows.

$$\max \sum_{t \in T} \left(\sum_{i \in I} p_i^t w_i^t - \sum_{j \in J} h_j^t r_j^t \right) \tag{1}$$

subject to:

$$w_i^t \leq d_i^t \qquad t \in T,\ i \in I \tag{2}$$

$$\sum_{j \in J(k)} x_{jkl}^t \leq M_{kl}^t y_{kl}^t \qquad t \in T,\ k \in K,\ l \in L(k) \tag{3}$$

$$\sum_{l \in L(k)} z_{kl}^t \leq 1 \qquad t \in T,\ k \in K \tag{4}$$

$$r_j^t + \sum_{i \in I(j)} a_{ij} w_i^t = r_i^{t-1} + \sum_{k \in K(j)} \sum_{l \in L(k)} c_{jk} x_{jkl}^t \qquad t \in T,\ j \in J \tag{5}$$

$$\sum_{k \in K(l)} \left(s_{kl}(y_{kl}^t - z_{kl}^{t-1}) + \sum_{j \in J(k)} b_{jkl} x_{jkl}^t \right) \leq f_l^t \qquad t \in T,\ l \in L \tag{6}$$

$$r_j^t = z_{kl}^t = 0 \qquad t = |T|,\ j \in J,\ k \in K,\ l \in L(k) \tag{7}$$

$$w_i^t,\ x_{jkl}^t,\ r_j^t \in \mathbb{Z}^+ \qquad t \in T,\ i \in I,\ j \in J,\ k \in K(j),\ l \in L(k) \tag{8}$$

$$z_{kl}^t,\ y_{kl}^t \in \{0,1\} \qquad t \in T,\ k \in K,\ l \in L(k) \tag{9}$$

Objective function (1) maximizes the profit, that is, the income of selling the assembled products minus the costs of holding pieces in inventory for the next periods. Constraints (2) limit the lot-size of the products by their demand in that period. However, pieces of type $j \in J(i)$ can be manufactured to be held as inventory for next periods. Constraints (3) force y_{kl}^t to be 1 when at least one batch is produced with mold k and machine $l \in L(k)$, where M_{kl}^t represents an upper bound on the number of batches that can be processed for each combination of mold k, machine l, and period t. Those constants can be set as the machine time f_l^t in period t divided by the smallest processing time among all pieces, i.e., $M_{kl}^t = \frac{f_l^t}{\min\limits_{j \in J(k)} \{b_{jkl}\}}$.

Since a mold can visit several machines, we must guarantee that a mold k can be straddling between periods t and $t+1$ on only one machine, this is represented by constraints (4). Inequalities (5) ensures—for a given piece type j—that the inventory at the end of period t plus the number of pieces of that type used for assembling products is equal to the inventory of the previous period $t-1$ plus the total production of that piece on period t. Constraints (6) guarantee that for each period t, the time required to install and remove all molds assigned to machine l plus the manufacturing time for all pieces on that machine cannot exceed f_l^t. Finally, constraints (7) state that there is no inventory nor straddling mold in the last planning period $t = |T|$.

This problem belongs to the NP-hard complexity class since the structure of constraints (6) is similar the ones of the multi-dimensional knapsack problem (Martello & Toth, 1990; Chekuri & Khanna, 2005) that is NP-hard. As it can be seen, the LS-MM does not consider the schedule of molds on machines thus, mold overlapping is possible to occur. However, we define in the following sections two families of inequalities in the LS-MM that enhance the behavior of our solution approach by decreasing the number of iterations due to infeasibility in the SCHED phase and reducing the space of feasible solutions for the LS-MM.

2.1. Limiting the usage of a mold by the maximum machine time

A feasible solution in the SCHED phase implies that there is no mold overlapping thus, an upper bound on the maximum time that mold $k \in K$ can be used is the maximum available machine time $\max\limits_{l \in L(k)} \{f_l^t\}$ among all machines compatibles with mold k. By using the latter, we define the following necessary but not sufficient condition to avoid mold overlapping in the LS-MM.

$$\sum_{l \in L(k)} \left(s_{kl}(y_{kl}^t - z_{kl}^{t-1}) + \sum_{j \in J(k)} b_{jkl} x_{jkl}^t \right) \leq \max_{l \in L(k)} \{f_l^t\} \qquad k \in K,\ t \in T \tag{10}$$

Indeed, inequalities (10) were succesfully implemented by Ibarra-Rojas et al. (2011) to reduce the number of iterations in a similar sequential approach for a related optimization problem in planning of plastic manufacturing systems.

2.2. Mold straddling in two consecutive periods imply that no other mold is installed on the machine

We assume that a mold can only be installed on a machine once per planning period. The latter assumption makes sense in plastic injection processes since installing a mold twice implies large setup times. However, we consider the multiple-period case where it is possible that a mold k could be straddling on machine $l \in L(k)$ in consecutive planning periods $t-1$ and t. In this case, if mold k is also straddling on the same machine on periods t and $t+1$, it is unlikely that in the optimal solution, a different mold k' would be installed on machine l, since this will lead to two large setup times for mold k on l. Then, we add the following inequalities to the LS-MM in order to reduce the space of feasibe solutions.

$$\sum_{k' \in K(l): k' \neq k} y_{k'l}^t \leq |K(l)|(2 - z_{kl}^t - z_{kl}^{t-1}) \qquad t \in T,\ k \in K,\ l \in L(k). \tag{11}$$

Notice that if variables $z_{kl}^t = z_{kl}^{t-1} = 1$ then mold k is straddling between periods $t-1$ and t, and t and $t+1$ on the same machine thus, the right side of the inequality is zero and it would be impossible to assign other mold k' to machine l in planning period t. On the basis of the above, our enhanced mathematical formulation for the LS-MM is defined by

$$\max \sum_{t \in T} \left(\sum_{i \in I} p_i^t w_i^t - \sum_{j \in J} h_j^t r_j^t \right)$$

subject to (2) – (11).

Experimental results show that the commercial solver of CPLEX 12.6 is capable of finding high quality solutions for the LS-MM MILP in short execution times (even for large instances). Moreover, our proposed constraints (10) and (11) are useful since the obtained solutions of LS-MM are feasible in the SCHED stage or only a few iterations of our solution approach are needed.

3. Mold scheduling on parallel machines (SCHED)

In this section we propose a mathematical formulation for the optimization problem (SCHED) that determines a schedule based on the mold-machine assignment obtained by solving the LS-MM. This schedule determines the processing order of the molds in the machines in each planning period t. Our formulation is an enhancement of the one proposed by Ibarra-Rojas et al. (2011) since we consideri idle times, multiple planning periods, and an objective function to optimize instead of focus only on feasibility.

Given a solution of LS-MM obtained in Phase I of our proposed sequential methodology we know the values of variables y^t_{kl}, z^t_{kl}, r^t_j, and x^t_{jkl}. Then, we define parameter τ^t_{kl} as the total time (installation and processing time) that mold k is on machine l in period t as follows.

$$\tau^t_{kl} := s_{kl}(y^t_{kl} - z^{t-1}_{kl}) + \sum_{j \in J(k)} b_{jkl} x^t_{jkl} \qquad t \in T,\ k \in K,\ l \in L(k). \tag{12}$$

For a given period, we represent a schedule as a serie of events for each machine or mold, where an event is defined as the use of one resource in combination with other type of resource (machine or mold). To consider the events on a single machine l in period t, we define parameter $\alpha^t_l := \sum_{k \in K(l)} y^t_{kl}$ which calculates the number of molds installed on machine l in period t (i.e., events of machine l). Alternatively, we may focus on the machines visited by a given mold, thus defining parameter $\beta^t_k := \sum_{l \in L(k)} y^t_{kl}$ as the number of events for mold k in period t. Moreover, we define the following precedence and event-based decision variables:

- $ma^t_{lk'k}$: binary variable taking the value of 1 if machine l process mold k' before processing mold k during period t, 0 otherwise.
- u^t_{knl}: binary variable that takes the value of 1 if mold k is the n-th event on machine l in period t, 0 otherwise.
- $mo^t_{kl'l}$: binary variable taking the value of 1 if mold k is installed on machine l' before being installed on machine l during period t, 0 otherwise.
- v^t_{lnk}: binary variable that takes the value of 1 if machine l is the n-th visited machine of mold k in period t, 0 otherwise.

Notice that, for $1 < t < |T|$, $\alpha^t_{k1l} = \beta^t_{l1k} = 1$ if mold k is straddling from period $t-1$ on machine l, since that mold is already installed on machine l and will be the first one to be processed. Hence the following constraints guarantee a feasible schedule free of mold overlapping among all planning periods.

$$\sum_{k \in K(l)} u^t_{knl} = 1 \qquad t \in T,\ l \in L,\ n = 1, \ldots, \alpha^t_l \tag{13}$$

$$\sum_{k' \neq k} ma^t_{k'kl} = \sum_{n=1}^{\alpha^t_l} (nu^t_{knl}) - y^t_{kl} \qquad t \in T,\ l \in L, k \in K(l) \tag{14}$$

$$\sum_{l \in L(k)} v^t_{lnk} = 1 \qquad t \in T,\ k \in K, n = 1, \ldots, \beta^t_k \tag{15}$$

$$\sum_{k' \neq k} mo^t_{kl'l} = \sum_{n=1}^{\beta^t_k} (nv^t_{lnk}) - y^t_{kl} \qquad t \in T,\ k \in K,\ l \in L(k) \tag{16}$$

$$mo^t_{kl'l} + mo^t_{kll'} \leq 1 \qquad t \in T,\ k \in K,\ l,\ l' \neq l \in L(k) \tag{17}$$

$$ma^t_{lk'k} + ma^t_{lkk'} \leq 1 \qquad t \in T,\ l \in L,\ k,\ k' \neq k \in K(l) \tag{18}$$

$$\sum_{k' \neq k} \left(\tau^t_{k'l'} ma^t_{k'kl}\right) + \tau^t_{kl'} \leq \sum_{k' \neq k} \tau^t_{k'l} ma^t_{k'kl} + \mathcal{M}^t(1 - mo^t_{kl'l}) \qquad t \in T,\ k \in K, l \in L(k) \tag{19}$$

Constrains (13) guarantee that for each machine $l \in L$, only one mold k (with $y^t_{kl} = 1$) is the n-th event. Inequalities (14) indicate that if mold k is assigned to machine l ($y^t_{kl} = 1$) and k is the $n-$th event of machine l ($u^t_{knl} = 1$), there must be $n-1$ molds that precede k on machine l, that is, $\sum_{k' \neq k} ma^t_{k'kl} = n-1$. In a similar way, constraints (15) and (16) define the events for each mold $k \in K$. Inequalities (17) and (18) preserve the order relation for events. Constraints (19) avoid mold overlapping where $\mathcal{M}^t$ is a big M parameter. Notice that the left hand side of (19) calculates the completion time of mold k on machine l' while the first term on the right hand side calculates the starting time of mold k on machine l thus, completion time of mold k on machine l' is less or equal than starting time of mold k in machine l if mold k is processed on machine l' before machine l, i.e., $mo^t_{kl'l} = 0$. Finally, we define $\mathcal{M}^t = \max_{l \in L}\{f^t_l\}$ as the maximum machine time available in period t.

The latter constraints (13)–(19) are a generalization of the ones proposed by Ibarra-Rojas et al. (2011), in order to considering multiple planning periods. Now, we introduce our enhancements to define an objective function to be optimized as well as a more flexible scheduling problem.

3.1. Consideration of the assembly line

As we mention in the problem statment, setup times of processing different pieces with the same molds are negligible in practice. For example, precedence for pieces produced within the same mold can be defined due to the color of the pieces in order to avoid cleaning time of cavitites in the mold. Then, we assume a given precedence of pieces, denoted as $j' \prec j$, if pieces of type j' should be manufactured before pieces of type j in the same mold.

Notice that for a given period t, $\sum_{k' \in K(l): k' \neq k} \tau^t_{k'l} ma^t_{k'kl}$ represents the earliest time in which mold k may be installed on machine l; $s_{kl}(y^t_{kl} - z^{t-1}_{kl})$ represents the installation time of mold k on machine l; and finally, $\sum_{j' \in J(k): j' \prec j} x^t_{j'kl} b_{j'kl}$ calculates the total processing time of all the pieces j' that precede piece

j in mold k mounted on machine l. Hence, we define auxiliary variables q^t_{jkl} that represent the starting time of production of pieces j with mold k installed on machine l whithin period t as follows.

$$q^t_{jkl} := \sum_{k' \in K(l): k' \neq k} \left(\tau^t_{k'l} ma^t_{k'kl}\right) \;+\; s_{kl}(y^t_{kl} - z^{t-1}_{kl}) \;+ \sum_{j' \in J(k): j' \prec j} x^t_{j'kl} b_{j'kl} \qquad t \in T,\; j \in J,$$
$$k \in K(j),\; l \in L(k) \tag{20}$$

Finally, we define variables o^t_i to represent the maximum difference between the starting times of processing pieces needed to assemble product i on period t which must satisfy the following inequalities.

$$q^t_{j'k'l'} - q^t_{jkl} \leq o^t_i \qquad\qquad i \in I, t \in T, (j,k,l) \neq (j', k', l') \tag{21}$$

We are interested in minimizing the weighted sum of such differences o_i in order to avoid large waiting times in the assembling stage thus, our mathematical formulation for the optimization problem SCHED is defined as follows (where parameter ω_i represents the preference of assembling products of type i).

$$\min \sum_{i \in I} \omega_i o^t_i \text{ subject to (19) - (21).}$$

Notice that starting times of molds used on machines are auxiliary variables defined using event-based variables ($ma^t_{k'kl}$ and $mo^t_{kl'l}$) thus, the schedule found through this event-based formulation does not consider idle time between the processing of two molds, which may lead to suboptimal solutions. Hence, we represent idle times by introducing a small number of dummy molds to be installed on machines with the procedure explained in the next subsection.

3.2. Allocation of idle time between molds installed on the same machine

As stated above, idle times are modeled as dummy molds with variable processing times. Since our scheduling problem is sensible to the number of molds to be scheduled in each machine, it is important to bound the number of dummy molds. The procedure presented in Algorithm 1 describes how dummy molds are created.

Algorithm 1 GenerateDummyMolds()

```
Calculate idle time IdleTime_l^t for each machine l at period t
for (each machine l at period t with IdleTime_l^t > 0) do
  while (IdleTime_l^t > MinLengthDummy) do
    Create a dummy mold with processing time IdleTime_l^t / 2 to be installed on machine l in period t
    Update the idle time of machine l at period t: IdleTime_l^t ← IdleTime_l^t / 2
  end while
  Create a dummy mold with processing time IdleTime_l^t to be installed on machine l in period t
end for
```

The only information required is the processing time of each mold. The first step is to determine if there is the need of dummy molds on machine l during period t. This is done by calculating the difference between the available time on the machine minus the total processing time of all the molds used on this machine in this period. Then, Algorithm 1 iteratively generates molds with decreasing processing time until a lower bound $MinLengthDummy$ of processing times of dummy molds is reached. This procedure is executed after we obtain a solution of the LS-MM and before solving the formulation of the SCHED problem.

3.3. Removing infeasibilities of SCHED from LS-MM

Once that straddling molds are identified in the solution obtained by solving the LS-MM, the SCHED problem is solved for each planning period t. In every iteration of our solution approach with an infeasible SCHED problem for some period t, we identify the mold k_t^* that visit the more machines for that period t, i.e., $k_t^* = \text{argmax}_{k \in K}\{\beta_k^t\}$ and we restrict its visits in the LS-MM by adding the following inequality.

$$\sum_{l \in L(k)} y_{k_t^* l}^t \leq |\beta_k^t| - 1 \tag{22}$$

Our main goal is cutting the infeasibilities in SCHED from the LS-MM feasible space thus, we iterate until we obtain a feasible solution of the SCHED problem for each planning period (as illustrated in Figure 3). However, we highlight that (22) is not necessarily a valid inequality of the integrated problem PPMM so we call this step as *heuristic cutting phase*.

4. Experimental evaluation of the heuristic based on MILPs for the PPMM problem

In this section we show how our heuristic approach for the PPMM problem behaves using randomly generated and real instances.

4.1. Randomly Generated Instances

We generated a set of 120 random instances to evaluate the proposed methodology. The parameters used to generate these instances is based on data from real plastic injection companies. Note that the instances from Ibarra-Rojas et al. (2011) do not consider products nor batches for pieces.

The number of pieces of type j needed by a product i, a_{ij}, is a randomly integer selected between [1,4]. The demand of a product i in a given period t, d_i^t, is an integer random number between $[100, 100000]$. The mold-setup times needed for installing and removing a mold k on machine l, s_{kl}, are integer uniformly distributed between [1,60] minutes. The number of cavities in mold k for piece j, c_{jk}, is a random integer between $[1, 20]$. We set the time taken for producing a batch of piece j with mold k on machine l, b_{jkl}, to be uniformly distributed between $[1, 200]$ seconds. The instances do not take into account maintenance times or plant shutdowns. We assume all machines work 24 hours a day, therefore the time available for each machine l is $f_l^t = 1440$ for all $t \in T$. Finally, the profit of a product p_i^t is also randomly distributed between $[1, 500]$ and is the same for all periods.

For each different setting (defined by number of products, pieces, molds, and machines), we generate 10 random instances. Our instance generator guarantees that for all instances, each product has at least one piece, each piece must belong to at least one product, each piece can be produced by at least one mold, each mold produces at least one piece, each mold can be mounted on at least one machine, and finally, on each machine at least one mold can be installed.

4.2. Results on generated instances based on real data

Our decomposition of the PPMM problem was executed on a MacPro with 3.5 GHz 6-Core Intel Xeon E5 with 32 GB of RAM. Formulations LS-MM and SCHED were solved with the integer linear solver of Gurobi 6.05 with the default settings except for the optimality gap which we set to 0 and the time limit of 60 minutes for the execution of each model, LS-MM and SCHED, independently of the number of iterations. Parameter $MinLengthDummy$, explained in Section 3.2, involved in the creation of dummy molds that represent idle times between molds is set to one hour.

Table 1 presents the results of solving the PPMM problem for instances with two periods. The demand of each product is homogenous for all periods. Each row represents the average over 10 instances. The first four columns are the description of the instance: number of Products, number of Pieces, number of Molds, number of Machines.The last three columns show the results of applying our decomposition method. Column "gap" indicates the average gap that is a measure of how far the best solution obtained is from the lower bound of the branch-and-algorithm algorithm[1], "T(s)" is the average total time in seconds of the decomposition algorithm to solve the instances while "Iterations" corresponds to the average number of iterations needed to find the optimal solution or the best solution within the time limit.

The results of Table 1 show that, on average, in less than 5.5 hours we can have a high quality production plan. The average number of iterations is no larger than six which validates the cutting plane decomposition methodology. The SCHED model is always solved to optimality in less than half a second for all instances. Although the LS-MM stage deals easily with instances of smaller size, as the size of the instances increases, the gap also increases since the time limit is often reached. The difficulty of the instances resides mainly of the number of jobs and secondly on the number of machines. Thus, we can solve real Indeed, the instance class (50,90,45,20) has a gap of 19% which shows the limit of our methodology.

We have observed that, at the end of the first iteration, many molds are installed in different machines at the same time. The fact that the gap of the SCHED stage is, on average, less than 3% and the number of iterations is no bigger than 6 shows that our speed-ups is efficient and not too restrictive. We also notice that for bigger instances, 200 pieces and 20 products, the 60 minute time limit may need to be increased if we want to improve the quality of the solutions.

Many companies plan for more than two periods, however, it is a common practice to use a rolling horizon approach (Pochet & Wolsey, 2006). That is, focus on finding feasible plans for a small number of

[1]The gap is computed as (ObjBound-ObjVal)/ObjVal, where ObjBound and ObjVal are the MILP objective bound and incumbent solution objective, respectively, as obtained from the branch-and-bound algorithm (Gurobi Optimization 2015).

Table 1: Decomposition method for PPMM with $T = 2$, and equal demand for each product in all of the periods.

Products	Pieces	Molds	Machines	gap	T(s)	Iterations
5	5	5	5	0		
5	10	5	5	0	0.08	1.9
5	20	10	10	0	1.9	2.1
5	30	15	20	0	80.05	3.1
10	5	5	5	0		
10	10	10	5	0		
10	10	10	5	0		
10	20	10	5	0	1.27	2.8
10	30	15	10	0	13.79	3.5
10	40	20	20	0	28.69	2.9
20	20	20	5	0		
20	30	15	5	0	6.73	3.1
20	40	20	10	0	713.64	4.4
20	50	25	20	0.06	8162.31	3.5
50	50	50	5			
50	70	35	5	0.01	1048.91	2.1
50	80	40	10	0.04	3915.04	3.3
50	90	45	20	0.19	19176.45	5.93
70	50	50	5			
70	90	40	5			
70	100	45	10			
70	110	50	20			

periods, and use this solution as a starting point for the next set of periods, until completing the whole planning horizon. For example, considering a horizon of six periods, $T = 6$, one can set the variables to be integers for only two periods at a time. Thus, first solving the problem, using integer variables, only for periods 1 and 2, and continuous variables in the remaining periods. Then, one can keep the solution for the first period, and solve the problem with periods 2 and 3 with integer variables, and continuous variables for periods 4, 5 and 6. Thus obtaining a solution for period 2. Repeating this rolling horizon approach until variables for periods 5 and 6 are set to integer values, thus reaching a solution for the complete horizon. One may vary the number of periods to be fixed as integer variables, but it is common practice to set this window to two periods, as explained for this example.

For sake of comparison with Table 1, we run for $T = 6$, and only setting variables for periods 1 and 2 to be integer. That is, we do not continue constructing integer solutions for periods 3 to 6. However, we do consider two scenarios, one where the demand for each product is the same throughout the different periods, and a second one where we allow different demands. Table 2 only shows the results for $\delta_{PP} = 25\%$ and $\delta_{PM} = 15\%$, which correspond to the values observed for most of the real companies of our case study. The first three rows consider instances with equal demand for each one of the products for all the periods while the last three have different demand for each one of the products at each period. The total demand for each product, over the six periods, for both scenarios is the same.

Table 2: Decomposition method for PPMM with $T = 6$ and a rolling horizon of two periods.

	Prd	Pcs	Molds	Machs	δ_{PP} %	δ_{PM} %	LS-MM % Gap	SCHED % Gap	Time (mins)	Iter
Equal demand	5	50	30	5	25	15	1.85	1.06	6.13	2.2
	10	120	80	20	25	15	4.02	1.51	205.07	5.4
	20	200	120	25	25	15	24.44	14.28	210.80	2.7
Diff. demand	5	50	30	5	25	15	2.03	1.08	0.15	1.5
	10	120	80	20	25	15	4.15	1.00	243.05	6
	20	200	120	25	25	15	14.97*	2.49*	220.83*	3.5*

Considering four more periods, with continuous variables, does not increase considerably the time it takes to solve the first two periods. Surprisingly, for instances with 5 and 10 products, the average gap in both stages were reduced. This implies than adding four extra periods, even if it is with continuous variables, help the method to reach a feasible schedule faster. For the big instances with 20 product, this does not hold, as reaching the 3% gap will require more than the allotted hour. In fact for instances with variable demand, there was one instance where a feasible solution was not found (mark with an "*" to indicate that only 9 out of 10 instances were solved). When studying the impact of having different demands over the six periods, our model seems to be competitive, achieving feasible solutions in similar times, as when the demand is constant; also the quality of the solutions, in terms of the gaps, are not worse. We may then conclude, that our model is capable of handling several periods as well as variable demand.

4.3. Evaluation of the decomposition approach in real companies

We now present results of applying this methodology on data from four real companies. Companies usually plan one period at a time, hence we run our decomposition methodology for only one period. We have observed that the production planners of the companies under study do not focus on the final price of the product, but instead they prioritize between different products; so that the profit of a product is not based only on its monetary value but also on customer satisfaction and production strategies. The marketing and sales teams are the ones in charge of setting prices and priorities, and usually the production planning team is only given a list of priorities on which products to manufacture on a given period. Thus, a particular product may have different priorities on different periods.

The planners on these companies follow similar policies to make a production plan: first choose the product with the highest priority and less quantity of pieces; if possible, use the installed molds to produce the pieces of these products; otherwise, install the molds that produce these products at the fastest rate. The companies have provided us with data on their completed production, which sometimes differ from the initial production plan. These differences are due to changes on demand and priorities that happened while the manufacturing was taking place, as well as unexpected machine failures.

To compare the results obtained using our methodology, we implement the above policy as a heuristic using the data on the completed production. We name this heuristic *Planner*. When applying *Planner* we found that it gave better solutions than the completed production. This is easy to understand as the production planning team has to deal the unforeseen changes and complex settings: many products and a variety of piece-mold-machine assignments.

4.4. Company A

Company A has 70 products that they assemble with 50 different pieces. It has 50 injection molds and 8 machines. Products have on average 2.6 pieces. This company has a mold per piece type so there is a one to one relation between molds and pieces; finally, molds can be installed in average on 5.5 machines. Each product has a demand of 2000 units in that period.

In Figure 4 we have Gantt charts showing the schedule obtained after applying our decomposition approach, on the left, and the one given by the *Planner* heuristic, on the right. Each row represents a machine, and each rectangle a mold, inside the rectangle we have indicated the pieces that are made on this mold. The length of each rectangle represents the processing time to produce the pieces including installation and removal time.

Notice that there is no idle time between molds, on any machine on either schedule. The decomposition approach suggest the use of 12 molds, while the Planner heuristic uses 10. One may think that using fewer molds will lead to a better schedule, however, the solution found with the decomposition approach is a 112% better than the one obtained by *Planner*, and was obtained in less than 1 second. There are

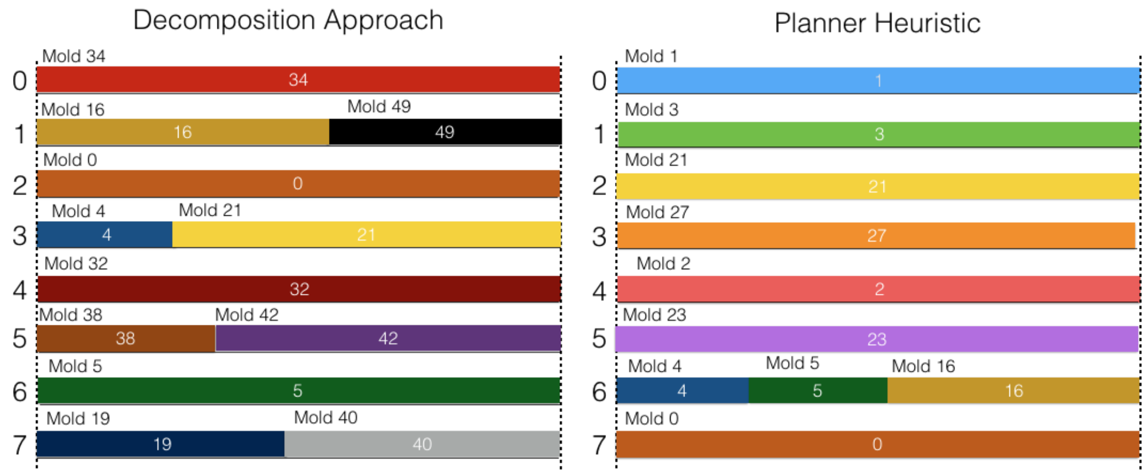

Figure 4: Gantt chart for the production plans for company A.

only five molds that are common in both solutions, three of them with the same processing time, and only one installed on the same machine. In the decomposition approach, there is a change of molds on half of the machines, while the Planner only consider a change of molds on one machine. Recall that machines may have different processing times for different molds, thus modify the solution obtained by the Planner heuristic to match the one obtained by the decomposition approach does not seem to be a straight forward task.

Table 3 shows the products and their lot size obtained in that period with both schedules. The first four columns correspond to the solution obtained by the decomposition approach, while the last four to the Planner heuristic. Columns show the type of product (Prd), the Lot-size, the value of the product (Priority), and the number of pieces needed to assemble that product (#Pcs per Prd). Note that the Planner always chooses products with higher priority and tries to produce as much as possible to satisfy the demand within the period. It seems that it also chooses products with fewer pieces. On the other hand, the solution obtained by the decomposition methodology, shows that there is a greater benefit in choosing a set of products that are not necessarily the ones with highest priorities or with less pieces.

Note that the decomposition approach schedule manufactures 15 different products, while the Planner heuristic produces only 7. The demand of 8 of the products manufactured by the decomposition approach is fully satisfied, while *Planner* only satisfies the demand for 1 product. There are only two products (3 and 20) that are common to both production plans. However, only the lot size of product 20 is the same on both solutions. The solution obtained by the decomposition approach does not only manufactures more products, but also with bigger lot sizes, and even though they have smaller priorities than those produced by the Planner heuristic, the total benefit is much greater. Hence, even though the planning heuristic seems to follow common sense, it does not come close to the optimal solution.

Table 3: Production plan in terms of products and the reported benefit for each approach

Decomposition Approach				Planner Heuristic			
Prd	Lot-size	Priority	# Pcs per Prd	Product	Lot-size	Priority	# Pcs per Prd
3	2000	166	2	0	1800	169	2
4	343	165	2	1	1600	168	2
15	2000	154	2	2	1080	167	2
18	1917	151	2	3	1780	166	2
20	2000	149	3	20	2000	149	3
31	2000	138	3	22	757	147	3
33	1999	136	3	26	619	143	3
39	2000	130	3				
41	1178	128	4				
43	2000	126	4				
56	567	113	2				
60	2000	109	2				
62	1999	107	2				
68	2000	101	1				
69	896	100	1				
Benefit		3282274				1546636	

4.5. Company B

Company B has 31 products that can be assembled with 37 pieces. There are 6 different molds, and 5 injection machines. This is a small company. Each product has an average of 2.4 pieces, each piece has an average of 1.2 molds that can produce it, and each mold can be on average installed on 2.8 machines. Each of the 31 products has a demand of 1000 units.

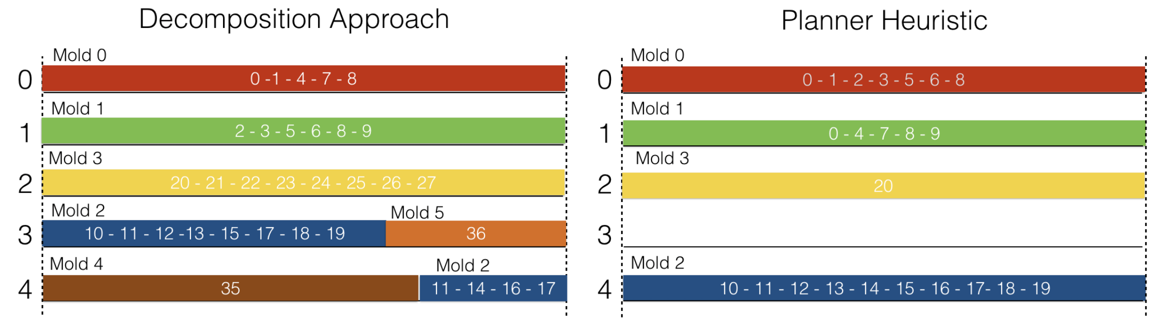

Figure 5: Gantt chart for the production plans of company B.

Figure 5 shows the Gantt chart for both schedules. The first thing we notice, is that machine 3 is not used in the Planner heuristic solution, due to the fact that this heuristic will not produce pieces if they do not complete a product. This highlights the importance of the correct placing of molds on machines. In practice, real planners would usually use these idle machines to schedule pieces on available molds that have a high probability to be used in the next period, increasing the inventory cost. However, we have observed that the Planner heuristic finds a better solution that the company's planning team.

The Planner heuristic uses only 4 out of the 6 available molds, whereas the decomposition approach uses all of them. In fact, mold 2 is moved from machine 3 to machine 4. This counterintuitive move increases the production rate of the pieces, and allows the production of more products. However, neither schedule produces all the required pieces. This may indicate that the demand for this period is higher than the capacity of the firm. It is also noticeable that although molds 0, 1 and 3 are placed in the same machines, in both schedules, they do not produce the same pieces.

Table 4: Production plan and lot size for each approach for Company B under the Decomposition Approach (D.A.) and the Planner Heuristic (P.H.)

Prd	0	1	2	3	4	5	6	7	8	9	10	11
Lot-size (D.A.)	1000	1000	1000	1000	1000	1000	1000	813	1000	1000	1000	1000
Lot-size (P.H.)	1000	1000	1000	1000	1000	1000	1000	1000	1000		1000	1000
Prd	12	13	14	16	17	18	19	20	21	22	23	
Lot-size (D.A.)	1000	1000	392	1000	1000	1000	1000	1000	1000	1000	510	
Lot-size (P.H.)	1000	1000	392									

Table 4 shows the production plan and lot size for company B. Notice that the Planner Heuristic produces 14 products, fulfilling the demand on 13 of them; while the decomposition approach manufactures 24 products, fulfilling the demand on 21 of them. Of the products that both plans have in common, the only difference is that the decomposition approach produces less units of product 7. This, with a better assignment of pieces to molds and machines, increases the final production by 62%.

4.6. Company C

Company C has 232 different products, 114 pieces, 57 molds, and 9 machines for a planning period of 120 hours. Each product has in average 1.06 pieces, each piece has an average of 1.3 molds that can produce it, and each mold can be on average installed on 6.7 machines. This is a large company where there is almost a one to one relation between molds and pieces. The difficulty for the planner lies on the amount of products and the flexibility on the mold-machine assignment.

For this company we have information for two non-consecutive planning periods. For one, the benefit of using the decomposition approach is of 86% while for the other it is of 170%. Since company C is larger than the others, the solution presented by the Planner heuristic is easily improved by the decomposition approach. Unlike the previous two companies, the decomposition schedule uses less molds: 31 and 15, instead of 44 and 34 respectively for each period. The reduced number of molds decreases the setup times and, with a better allocation of molds to machines, manages to increase the production. Moreover, the decomposition methodology finds the solution in 8 seconds while we have observed that the planning team may take up to one day to produce a feasible production plan.

4.7. Company D

Company D is the largest solvable setting in our case study. It has 388 products, 76 different pieces and 61 molds that can be mounted on 28 machines. Each product has, on average, 1.09 pieces; each piece can be made, on average, by 1.13 molds, while each mold can be mounted, on average, on 22.4 machines. Thus, the complexity of company D does not only reside in the large number of products but also in the large compatibility between molds and machines. For company D, the decomposition approach has increased the production by 19%, using 61 molds while the Planner heuristic uses 71 molds. Thus, even if the products have an almost one to one relation with the pieces and the pieces also have an almost one to one relation with the molds, the compatibility of the molds to the machines make the problem hard to solve for a real planner. The decomposition methodology yields its solution in less than 32 seconds.

These examples show the advantage of using the decomposition approach to solve the PPMM problem. This translates into higher production rates, bigger benefits for the companies, better satisfaction of the demand, and shorter times to find feasible schedules. Even though, we did not apply the methodology for more than one period, we have shown in Section 4 that it is easy to use this methodology to solve multi-period instances.

5. Conclusions

In this paper we introduce the Product-Piece-Mold-Machine (PPMM) problem which deals with the manufacturing of a set of pieces to assemble products. Each piece may be produced in a set of molds, and each mold, in turn, can be installed on a certain number of machines. The aim is to determine the multi period lot production size of each product, the piece-mold and the mold-machine assignments to maximize the benefits of manufacturing the products.

Unlike previous research, our model allows molds to be installed on different machines on the same period. Our results show that this counterintuitive assumption yields better schedules. Traditionally, research on this area has focused on the lot size of the pieces rather than the amount of completed products. To the best of our knowledge this is the first paper that considers as an objective the profit of the assembled products. We also consider work in process factors by introducing precedence constraints in the assembly line, and inventory costs from one period to another.

The key point of our approach is a decomposition of the PPMM problem. We first determine the lot-size of the products along with the mold-machine assignments. In this stage we drop the mold overlapping constraint and add additional constraints that avoid a large number of suboptimal solutions. The second stage of our methodology determines if there is a feasible scheduling of the molds on the machines for each period. If such schedule is found we have the optimal solution. Otherwise, we return to the first stage adding restrictions forcing certain molds to be installed on only one machine, and iterate through these stages until a feasible solution is found. Additionally, we introduce speed-ups that make the implementation of our decomposition approach of the PPMM problem viable in practice.

Experimental results on generated instances show the efficiency of our approach for different sizes of the instances of the problem when used with a rolling horizon structure. We were able to solve problems with a high compatibility between products to pieces and pieces to molds. Considering more than one period reduces the gap when dealing with instances with compatibilities similar to those found in real companies with less than 20 products. Allowing different demands on each period does not decrease substantially the quality of the solutions obtained. An important observation is that higher compatibility factors leads to longer computational times.

Several real instances are analyzed to show the performance of the decomposition approach. These instances have a very low compatibility factor between products to pieces and pieces to molds. Thus, our approach finds solutions in less than one minute. The schedules obtained with this methodology have a higher production, bigger benefits for the companies and better satisfaction of the demands. Hence, this approach can be used by the manufacturing planning team to find solutions in shorter times. This methodology could also facilitate the analysis of strategic decisions such as evaluating the advantage of adding an extra machine, or increasing the compatibility between pieces to molds and molds to machines.

Acknowledgments

We are very grateful to Logykopt (`http://www.logykopt.com/`), that has developed SCHEMA (a planning software for the plastic mold injection industry), for having shared with us their experience with real companies of plastic injection molding. E. Possani and M. Cabo would like to thank Associación Mexicana de Cultura A.C. for their support. Additionally, we wish to thank Miguel Morales for all the fruitful discussions about the single period version of the PPMM problem.

References

Beraldi, P., Ghiani, G., Grieco, A., & Guerriero, E. (2008). Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. *Computers & Operations Research*, *35*, 3644–3656.

Chekuri, C., & Khanna, S. (2005). A polynomial time approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing*, *35*, 713–728.

Chen, J.-F., & Wu, T.-H. (2006). Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints. *Omega*, *34*, 81–89.

Dastidar, S. G., & Nagi, R. (2005). Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs. *Computers & Operations Research*, *32*, 2987–3005.

Drexl, A., & Kimms, A. (1997). Lot sizing and schedulingsurvey and extensions. *European Journal of Operational Research*, *99*, 221–235.

Fandel, G., & Stammen-Hegene, C. (2006). Simultaneous lot sizing and scheduling for multi-product multi-level production. *International Journal of Production Economics*, *104*, 308–316.

Haase, K., & Kimms, A. (2000). Lot sizing and scheduling with sequence-dependent setup costs and times and efficient rescheduling opportunities. *International Journal of Production Economics*, *66*, 159–169.

Ibarra-Rojas, O., Ríos-Mercado, R., Rios-Solis, Y., & Saucedo-Espinosa, M. (2011). A decomposition approach for the piece–mold–machine manufacturing problem. *International Journal of Production Economics*, *134*, 255–261.

James, R. J., & Almada-Lobo, B. (2011). Single and parallel machine capacitated lotsizing and scheduling: New iterative mip-based neighborhood search heuristics. *Computers & Operations Research*, *38*, 1816–1825.

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

Kim, S.-i., Han, J., Lee, Y., & Park, E. (2010). Decomposition based heuristic algorithm for lot-sizing and scheduling problem treating time horizon as a continuum. *Computers & Operations Research*, *37*, 302–314.

Lin, C., Wong, C., & Yeung, Y. (2002a). Heuristic approaches for a scheduling problem in the plastic molding department of an audio company. *Journal of Heuristics*, *8*, 515–540.

Lin, C. K. Y., Wong, C. L., & Yeung, Y. C. (2002b). Heuristic approaches for a scheduling problem in the plastic molding department of an audio company. *Journal of Heuristics*, *8*, 515–540.

Martello, S., & Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc.

Meyr, H., & Mann, M. (2013). A decomposition approach for the general lotsizing and scheduling problem for parallel production lines. *European Journal of Operational Research*, *229*, 718–731.

Nagarur, N., Vrat, P., & Duongsuwan, W. (1997). Production planning and scheduling for injection moulding of pipe fittings a case study. *International journal of production economics*, *53*, 157–170.

Ouerfelli, H., Dammak, A., & Chtourou, E. K. (2009). Benders decomposition method for an integrated problem of production lot-sizing and scheduling. In *Computers & Industrial Engineering, 2009. CIE 2009. International Conference on* (pp. 323–327). IEEE.

Pochet, Y., & Wolsey, L. A. (2006). *Production planning by mixed integer programming*. Springer Science & Business Media.

Quadt, D., & Kuhn, H. (2008). Capacitated lot-sizing with extensions: a review. *4OR: A Quarterly Journal of Operations Research*, *6*, 61–83.

Stadtler, H. (2011). Multi-level single machine lot-sizing and scheduling with zero lead times. *European Journal of Operational Research*, *209*, 241–252.