Multi-piece mould design based on a mixed-integer programming method

Yong Chen* and Stephen Stoyan

Epstein Department of Industrial and Systems Engineering, University of Southern California, Los Angeles, CA 90089, USA

(Received 6 April 2011; final version received 12 February 2012)

Multi-piece mould design is a moulding technology that involves three-dimensional spatial construction of two or more mould pieces in a manner similar to assembling/dissembling a three-dimensional puzzle to build production parts. Using such a moulding technology, complex parts with intricate geometries can be made for limited run productions. Compared to traditional two-piece moulds and rapid prototyping, the multi-piece mould approach has many advantages with respect to part complexity and production speed, etc.; however, the technology has challenges in designing the actual multi-piece moulds. Previous methodologies address this problem primarily using heuristics. We present a multi-piece mould design (MPMD) framework that is based on a mixed-integer programming approach. The method constructs the MPMD by minimising the number of mould pieces that is required for a given Computer-Aided Design (CAD) model. The solution strategy for the formulated linear mixed-integer optimisation problem is presented. The algorithmic strategy for solving the resulting mixed-integer programming problem is also provided with examples that illustrate the effectiveness and efficiency of the approach.

Keywords: computer-aided mould design; multi-piece moulding; rapid tooling; mixed-integer programming

1. Introduction

Multi-piece mould design (MPMD) typically involves complex modelling using more than two mould pieces to construct production parts. The technique involves a three-part process, where the first step consists of hand-loading the mould pieces into a mould base that is mounted on an injection moulding machine. Next, part injection and cooling processes are undergone, where the mould pieces are accurately and securely clamped into a holding device. Finally, each mould piece is hand-removed from the mould base to release the injection moulded part as final product. Space puzzle modelling (Protoform GmbH 2011) is an example of multi-piece moulding technique based on metal tools. Figure 1 provides another example based on silicon tools, which also includes the related moulded parts produced by the process. As illustrated, each mould piece can have its own parting direction (PD) that allows the mould piece to be separated from the part. Compared to traditional two-piece moulds, multi-piece moulds can produce limited-run production parts with more complex geometries. Compared to rapid prototyping (RP) processes such as Stereolithography Apparatus (SLA), multi-piece moulding techniques can produce a small quantity of parts more efficiently than what is typically used in the RP processes. Furthermore, the fabricated parts can be made in an injection moulding material that may not be available to the RP processes. Hence, multi-piece moulding has become an important tooling technology in the era of mass custom production, in which limited-run production is increasingly becoming a common industrial practice (Wohlers 2004).

Despite the promises for limited-run production, there are inherent challenges in designing multi-piece moulds for a given Computer-Aided Design (CAD) model. Factors such as the part geometry, the selection of mould design variables, the number of different mould pieces, etc. need to be considered in producing the part. Generally, it is desired to minimise the number of mould pieces since fewer mould pieces reduce the cost associated with tooling and simplifies the operation of the mould, which may also reduce errors in part production. In this article, we consider the automation of the MPMD problem defined as follows:

MPMD problem: Given a solid part and a mould base, design the minimum number of mould pieces that can form the cavity of the part using the material injection process, and can be disassembled properly in the part ejection process.

1.1. Related work

The automation of mould design for injection moulding has been extensively studied. Some representative
work can be found in Hui (1996), Wong et al. (1998), Fu et al. (1999), Yin et al. (2001), Ye et al. (2004), McMains and Chen (2006). Most of the relevant research focuses on two-piece moulds, including the determination of parting directions, parting lines, parting surfaces, and undercut detecting. Among them, the selection of parting directions has received a lot of attention since it is an important step in the automatic mould design process. Recently, new programmable graphics hardware has accelerated algorithms that have been used to test the mouldability of parts and help redesign them (Kharderkar et al. 2006). Chen and Rosen (2002, 2003) first presented an MPMD method that allows three-dimensional mould decomposition. More recently, Gupta’s group presented a set of geometric algorithms for the automated design of multi-piece permanent moulds (Priyadarshi and Gupta 2004), sacrificial multi-piece moulds (Huang et al. 2003) and multi-stage moulding (Priyadarshi and Gupta 2009). Such work provides excellent foundation for MPMDs and future improvements.

One key problem with current approaches is that the parting directions of multi-piece moulds are only identified by using heuristics. Hence, no optimality was considered for a given arbitrary geometry. For example, in designing multi-piece permanent moulds (Priyadarshi and Gupta 2004), a set of candidate parting directions are selected from: (1) principal axis directions, (2) planar face normals and (3) cylindrical/conical face axes. Based on them, a solution scheme is developed to identify the number of required mould pieces. However, such methods may have difficulties in finding solutions for complex geometries. An example of a complex injection moulding parts based on a plastic component of a paper-handling device is given in Protoform GmbH (2005). Based on designer’s knowledge, the MPMD developed by Protoform GmbH (2005) consists of more than 80 mould pieces. Such complex MPMDs based on heuristic provide opportunity and motivation for an optimisation-based approach. Our research contributes to the automation of MPMD, which is a critical component of tooling technologies used in limited-run productions and supports future developments in the field.

In this article, we present the framework and methods to solve the MPMD using a mixed-integer programming approach. For a given production part, the approach will minimise the number of mould pieces required to make the part and provide guidelines on where to separate the mould pieces apart. Based on the results, additional measures can then be used in improving the identified mould design. Our design framework provides a unique and solid foundation in solving MPMD for a given CAD model. In addition, by formulating the mould design into a mixed-integer program, our approach can be efficiently and robustly implemented using commercially available optimisation solvers.

1.2. Problem formulation

Before discussing our approach, a more detailed formulation of the aforementioned MPMD Problem is presented as follows:

MPMD problem: Given a polygonal model ($P$) and a mould base ($\Psi$), design a $n$-piece mould $M = \{m_1, m_2, \ldots, m_n\}$ to:

Minimise: number of mould pieces ($n$).

Subject to:

1. Each $m_i \in M$ is distinct as a connected solid;
2. Each $m_i \in M$ has a parting direction $d_i$ such that $m_i$ can be translated along $d_i$ without interfering the injection moulded part. The global interference between mould pieces in the disassembly process is important but has not been considered in the article;
3. $M = \bigcup_{i=1}^{n} m_i$ satisfies $M = \Psi - P$

Figure 1. An example of multi-piece moulds (Test 6 in Section 6).
The MPMD problem is similar to the ones considered in Chen and Rosen (2002), and Priyadarshi and Gupta (2004).

2. Multi-piece mould design approach

For a part to be mouldable, every face on the part needs to be accessible from at least one direction. Usually such a direction can be easily found for each individual face; however, the core challenge in mould design is to find the minimum number of directions that are common to all of the faces in the part.

2.1. Mould design based on visibility of faces

Chen et al. (1993) formulated demouldability as a visibility problem and presented a set of important computational techniques such as visibility and Gaussian maps (V-maps and G-maps). For a planar surface $F_i$, its V-map is a hemisphere centred on the unit outward normal. By calculating the intersections of all V-maps of region faces, allowable draw ranges can be computed. Several approaches and algorithms based on spherical polygons have been presented for different applications (Woo 1994, Dhaliwal et al. 2003). The implementations of such algorithms require significant efforts. The computation geometry approach is also well-known challenging to be implemented robustly. In addition, it takes considerable computational time to compute the exact V-map intersections on spherical surfaces.

Instead, Chen and Rosen (2002) presented an alternative approach based on a linear program for evaluating the parting directions of a set of surfaces. Suppose a connected region $R$ consists of face $F_i$ (with unit face normal $N_i$ and area $A_i$, $1 \leq i \leq n$). If a common direction $d$ ($d_x$, $d_y$, $d_z$) exists such that every face $F_i$ can be accessible from such a direction, $d$ is a candidate parting direction of the region. In addition, the ease of ejection can be used as the criterion to choose an optimal parting direction from a feasible range. For a face $F_i$, its ease of ejection can be determined by the draft angle and the area in shear contact with the related part face during the mould-opening operation (i.e. $A_i(N_i \cdot d)$). Other heuristics can also be used, e.g. max ($\min(A_i(N_i \cdot d))$). Hence an optimisation problem for determining a parting direction of region $R$ can be formulated as follows:

Maximise: $\sum_{i=1}^{n} A_i(N_i \cdot d)$

Subject to: $N_{xi}d_x + N_{yi}d_y + N_{zi}d_z \geq 0$ for face $F_i$

$A_i = \pi(d_x^2 + d_y^2 + d_z^2) = 1$ (sphere constraint).

The sphere constraint makes the optimisation problem nonlinear and difficult to solve. A unit sphere related to the sphere constraint can be approximated by a set of linear surfaces with acceptable errors. Suppose the equations of a planar surface $M_i$ are $M_{xi}d_x + M_{yi}d_y + M_{zi}d_z = \mu_i$ with face normal $(M_{xi}, M_{yi}, M_{zi})$ toward the inside. Hence a linear problem can be formulated for evaluating a parting direction of the region.

Parting Direction Linear Problem (PDLP) Problem:

Maximise: $\sum_{i=1}^{n} A_i(N_i \cdot d)$

Subject to: $N_{xi}d_x + N_{yi}d_y + N_{zi}d_z \geq 0$ for face $F_i$

$M_{xi}d_x + M_{yi}d_y + M_{zi}d_z \geq \mu_i$ for face $M_i$. (2.3)

A linear program problem in three dimensions can be solved in $O(n)$ time ($n$ is the number of constraints) and on linear storage. Therefore, the running time to solve the PDLP Problem is very fast, typically in milliseconds for thousands of faces based on a commercial solver (e.g. LINGO system – www.lindo.com). More importantly, the computation process of finding a parting direction for a set of faces becomes much easier and more robust.

Motivated by the linear programming approach, we further develop a linear mixed-integer program for determining the minimum number of parting directions for all the faces of a given part. The optimisation formulation is discussed in Section 3. A simple example (Test 1 in Section 6) is shown in Figure 2 to illustrate the objective in the PDLP Problem, which is also used in the mixed-integer program. For a simple cube with a size of $5 \times 2 \times 1$, there is an infinite number of parting directions that can be used in its mould design. However, considering the ease of ejection, the two directions, $d_1$ and $d_2$ as shown in the figure, are the most desired. In such directions, the projection area $\sum_{i=1}^{n} A_i(N_i \cdot d_1 + N_i \cdot d_2)$ is also the maximum.

Figure 2. A simple example to illustrate the optimisation objective in PDLP.
2.2. Basic elements for multi-piece mould design

In the PDLP Problem, a single face $F_i$ is used as the basic element in evaluating a feasible parting direction. However, such a face is not a good element in identifying the minimum number of parting directions in MPMDs. This is because the accessible directions to a face may be blocked by neighbouring faces (e.g. two neighbouring faces of a concave edge). However, such blockage is difficult to be incorporated if faces are directly used as the basic element.

As shown in Figure 2, any pair of directions can be used in a mould design if the given part contains only convex edges. Based on such an idea, Chen et al. (1993) computed a set of 'pockets' based on the convex hull of an object to capture all non-convex regions. Accordingly, the visibility map of each pocket can be computed and the parting directions that maximise the number of completely visible pockets can be determined.

For a test part (refer to Table 3 – Test 4 in Section 6 and Figure 9), an example of pockets is given in Figure 3. The four pockets of the part are shown in different colours. It can be seen that a pocket is a moulding feature which contains multiple concave and convex edges. As discussed in Chen and Rosen (2002), pockets give us less design freedom with respect to using different combinations of elements; hence they are also not suitable for identifying the minimum number of parting directions in the MPMD.

As discussed in Chen and Rosen (2002), concave edges (i.e. the dihedral angle of two neighbouring faces is larger than 180°) can capture the blockage of accessible directions between neighbouring faces. Based on such edge classification, two types of elements can be defined as follows:

**Definition 2.1.** A concave region is a set of connected faces such that: (1) all internal edges of the region are concave or flat edges and (2) all boundary edges of the region are convex or flat edges.

**Definition 2.2.** All other faces $F$ are convex faces with only convex or flat edges.

As an example, the concave regions for the test part in Figure 3 are shown in Figure 4. There are a total of 12 concave regions that are shown in different colours; in addition, there are 68 convex faces that are drawn in wireframe in Figure 4. Hence, when compared to pockets, their combinations for identifying the minimum number of parting directions have been significantly increased.

Similar to Chen and Rosen (2002), concave regions and convex faces are used as the basic elements in our
MPMD method. A similar strategy has also been adopted in their generation. However, different from Chen and Rosen (2002), that is based on a 3D geometric modeller (ACIS from Spatial – www.spatial.com), our current method utilises polygonal meshes (i.e. the input model is defined in triangulated surfaces), which is the de facto format in RP and manufacturing. Hence the approach in computing concave regions and convex faces is accordingly modified as follows by adding a concept of Convex-Edge-Vertex:

(1) Classify all edges as concave, convex and flat.
(2) Assign two faces to the same region if they share a concave or flat edge.
(3) Identify all the internal convex edges of each region and mark the vertices of such edges as Convex-Edge-Vertex. Notice: we do not want a region to have internal convex edges (i.e. all the internal edges should be concave as shown in Definition 2.2).
(4) Regenerate regions by assigning two faces to the same concave region if they share a concave edge, or a flat edge if such an edge has no vertex that is marked as Convex-Edge-Vertex.

2.3. Approach overview

Based on the identified basic elements (a set of concave regions and convex faces), the essence of the MPMD is to generate a different combination of them, and accordingly identify a design with the best performance (i.e. the minimum number of mould pieces and the easiness of ejection if the same number of mould pieces is achieved). Similar to previous work on MPMD (Chen and Rosen 2002, Priyadarshi and Gupta 2004), we focus on the local interference between neighbouring mould pieces. The global interference, which is computationally complex in the space puzzle moulding, will not be considered.

There have been various approaches to solve such combination problem between elements (e.g. the well-known knapsack problem). General solution methods involve: (i) searching all possible combinations of variables, (ii) using heuristics such as greedy heuristic or (iii) optimisation methods such as integer and dynamic programming. Each solution method has advantages and disadvantages. Searching all combinations, for example, has costs with respect to solution time, especially with large-scale problems where validating a solution may take years. Heuristics typically generate fast solutions, but they do not guarantee generality and optimality. Optimisation methods can guarantee optimality and the related techniques can converge faster than searching the whole solution space; however, they generally perform slower than heuristics. When models involve a large number of integer variables (such as the ones in the next section), the issues with the various solution methods described above become worse. Although optimisation methods have trade-offs, depending on the complexity of the problem, current state-of-the-art solvers such as CPLEX (from ILOG – www.ilog.com) can solve large-scale problems in reasonable time.

As discussed in Section 1, the previous MPMD methods are based on heuristics. They are effective but may fail for geometries that have not been considered when generating such heuristics. In this article, we formulate the MPMD Problem into a linear mixed-integer program and solve it based on optimisation methods. As will be shown in Section 6, by using CPLEX and a decomposition algorithm for the MPMD Problem, such models can be solved relatively fast and generate satisfactory solutions.

An overview of our method that is based on an example (Test 3 in Section 6) is shown in Figure 5.

(1) A given part to be injection moulded is shown in Figure 5(a). The part has 64 triangles. They can be classified into three concave regions (drawn in different colours) and 40 convex faces (drawn in wireframe) as shown in Figure 5(b).
(2) Based on such elements, a mixed-integer program is formulated and implemented using CPLEX and an ad-hoc algorithm. Accordingly, two parting directions, \(d_1 = (0, 0, 1)\) and \(d_2 = (0, -1, 0)\), are identified by the optimisation solver.
(3) The solution provides a lower bound on the solution (i.e. \(n = 2\)). Based on them, all the concave regions and convex faces can be combined into two-mould piece regions as shown in two different colours in Figure 5(c). Since multiple solutions may exist, a best one may be identified based on moulding design knowledge.
(4) Based on the generated mould piece regions and related parting directions, parting lines and parting surfaces can be identified. Accordingly, two mould pieces, \(M_1\) and \(M_2\), can be constructed, as illustrated in Figure 5(d). One may observe that there is a one to one correspondence between the mould pieces and the mould piece regions. In addition, the two mould pieces can be assembled and disassembled properly in the related parting directions. Additional locking features can be added in the mould pieces. In comparison, the MPMD based on heuristics (Chen and Rosen...
2002) for the same example is shown in Figure 5(e). Three mould pieces instead of two are constructed, which is less efficient.

In this article, we focus on the process of generating mould piece regions from concave regions and convex faces. The remainder of the article is organised as follows. Section 3 presents the optimisation formulation for the MPMD Problem. Section 4 discusses the draft angle and the face connectivity of mould piece regions based on the identified part directions. The further improvements of the mould design are discussed in Section 5. The experimental results are discussed in Section 6 and finally, conclusions and future work directions are provided in Section 7.

3. Optimisation formulation for computing parting directions

Before getting into the intricate details of the problem, we first outline the decision variables and parameters associated with the model. To begin we define:

- $n$: the total number of parting directions of mould pieces;
- $p$: the total number of concave regions;
- $m$: the total number of approximated spherical surfaces (refer to sphere constraint in PDLP problem); and their respective sets as:

$$
\phi := \{ i : i \in [1, n] \} : \\
\psi := \{ k : k \in [1, p] \} : \text{the set of concave regions;} \\
\Theta := \{ \ell : \ell \in [1, m] \} : \text{the set of spherical surfaces.}
$$

In addition to the number of concave regions, there also exist a number of surfaces associated with each region, which do not necessarily have the same number of surfaces. Thus, we define the set of surfaces associated with each concave region and introduce the set of convex surfaces as:

- $\beta_k$: the set of surfaces in each concave region $k = 1, \ldots, p$;
- $\beta_c$: the set of convex surfaces.

Figure 5. An overview of our method.
The decision variables involved in the optimisation problem are as follows:

\[ d_i = [d_i^1, d_i^2, d_i^3] \]  \text{ the vector of parting directions for mould pieces } i = 1, \ldots, n;  \\
\text{ where the solution } n \text{ gives the minimum number of parting directions } (d_i \neq 0) \text{ to Equations (3.1)–(3.11). Then, the second optimisation problem (II) is:  \\
\text{Maximise: } \sum_{i=1}^{n} \sum_{k=1}^{p} A(k, \beta_k)|N(k, \beta_k)d_i|  \\
+ \sum_{i=1}^{n} A(\beta_c)|N(\beta_c)d_i|  \\
\text{Subject to: } (3.2) – (3.11).  \\
\text{The constraints are the same for both problems since they define the boundaries of the mould to be created. Constraints Equations (3.2)–(3.3) and (3.4)–(3.5) ensure that at least one parting direction vector } d_i \text{ satisfies the desired inequality and the rest can be turned ‘off’ via the binary variables } g_i(k) \text{ and } g_i(\beta_c); \text{ respectively. There are issues, however, with both Equations (3.1)–(3.11) and (3.12)–(3.13) in that the objective functions and constraints are nonlinear and also involve binary variables } g_i(k) \text{ and } g_i(\beta_c). \text{ However, the NonLinear Mixed-Integer Program (NLMIP) of Equations (3.1)–(3.11) and Equations (3.12)–(3.13) can be converted to an equivalent linear program with the introduction of a decomposition strategy and a few additional variables and constraints. The linear transformation allows the problem to be more tractable and easier to solve.}  \\
\text{The design described in the earlier sections requires the solution to two optimisation problems. Given a set of constraints that define the mould we want to create, the first problem (I) entails finding the minimum number of vector parting directions } d_i \text{ necessary to design the mould. After we know the minimum number of parting directions, the second problem (II) involves maximising the surface area covered in forming the mould pieces. The first optimisation problem (I) is the following:}  \\
\text{Minimise: } n  \\
\text{Subject to: } g_i(k)(N(k, \beta_k)d_i) \quad \forall i \in \phi, k \in \psi  \\
\sum_{i=1}^{n} g_i(k) \geq 1 \quad \forall k \in \psi  \\
g_i(\beta_c)(N(\beta_c)d_i) \geq 0 \quad \forall i \in \phi  \\
\sum_{i=1}^{n} g_i(\beta_c) \geq 1  \\
\text{where } M_i \geq u_i \quad \forall i \in \phi, \ell \in \theta  \\
|d_i| \geq L \quad \forall i \in \phi  \\
|d_i| \leq U \quad \forall i \in \phi  \\
d_i \in \mathbb{R} \quad \forall i \in \phi  \\
g_i(k) \in \{0, 1\} \quad \forall i \in \phi, k \in \psi  \\
g_i(\beta_c) \in \{0, 1\} \quad \forall i \in \phi,
solution, thus subproblem (1) begins with \( d_1 \) and \( d_2 \). Then we stop after the first instance when the subproblem obtains a feasible solution. The objective of problem (II) aims to maximise the surface area associated with the absolute value of the direction vector and the corresponding surface region. This has a linearly equivalent set of equations by introducing the following:

Maximise:

\[
\sum_{k=1}^{n} \frac{1}{A(k, \beta_k)} \left( \sum_{i=1}^{n} z_i(k) \right) \tag{3.14}
\]

Subject to:

\[
z_i(k) \leq N(k, \beta_k) d_i \quad \forall i \in \phi, k \in \psi \tag{3.15}
\]

\[
z_i(k) \leq -N(k, \beta_k) d_i \quad \forall i \in \phi, k \in \psi \tag{3.16}
\]

This is also done for \(|N(\beta_d) d_i|\) where \( z_i \) is used, which is shown in Equations (3.22), (3.25) and (3.26) below.

The nonlinear constraints in the problem can be addressed in a similar fashion, which make the problem more tractable and less complex. The nonlinear constraints in Equations (3.2) and (3.4) can be converted into linear constraints by using the following equation:

\[
\Lambda(1 - g_i(k)) + N(k, \beta_k) d_i \geq 0 \quad \forall i \in \phi, k \in \psi \tag{3.17}
\]

where \( \Lambda \) is a large scalar value. Here, when \( g_i(k) \) is equal to zero then this constraint is essentially turned ‘off’ since the large \( \Lambda \) value will satisfy the inequality for any \( d_i \). When \( g_i(k) \) is equal to one then \( N(k, \beta_k) d_i \geq 0 \) must be satisfied, which is the desired function of constraint Equation (3.2). This is again repeated for \( g_i(\beta_d) N(\beta_d) d_i \), as is shown below in Equation (3.29).

Finally, the absolute value upper and lower bound constraints of Equations (3.7) and (3.8) can be linearly expressed by using the same idea in Equations (3.14)–(3.16). We first have to set the coordinates of each parting direction to be an absolute value and then ensure their product is greater than or equal to the lower bound and less than or equal to the upper bound. Given \( d_i = [d_i^1, d_i^2, d_i^3] \), to obtain the following property: 

\[
L \leq |d_i^1 + d_i^2 + d_i^3| \leq U,
\]

the constraints below will have to be added to the problem:

\[
\lambda_i^1 \leq d_i^1 \quad \forall c = \{x, y, z\}, i \in \phi \tag{3.18}
\]

\[
\lambda_i^1 \leq -d_i^1 \quad \forall c = \{x, y, z\}, i \in \phi \tag{3.19}
\]

\[
-(\lambda_i^1 + \lambda_i^2 + \lambda_i^3) \geq L \quad \forall i \in \phi \tag{3.20}
\]

\[
-(\lambda_i^1 + \lambda_i^2 + \lambda_i^3) \leq U \quad \forall i \in \phi, \tag{3.21}
\]

where \( \lambda_i^1, \forall c = \{x, y, z\}, i \in \phi \) is maximised in the objective function, as shown in Equation (3.22) below.

Thus, the original problem of solving problem (I) and (II) separately can now be done in one equivalent linear problem. This equates to solving the following linear Mixed-Integer Program (MIP):

Maximise:

\[
\sum_{k=1}^{p} \frac{1}{A(k, \beta_k)} \left( \sum_{i=1}^{n} z_i(k) \right)
+ \frac{1}{A(\beta_c)} \left( \sum_{i=1}^{n} z_i(\beta_c) \right)
+ \sum_{i=1}^{n} (\lambda_i^1 + \lambda_i^2 + \lambda_i^3) \tag{3.22}
\]

Subject to:

\[
z_i(k) \leq N(k, \beta_k) d_i \quad \forall i \in \phi, k \in \psi \tag{3.23}
\]

\[
z_i(k) \leq -N(k, \beta_k) d_i \quad \forall i \in \phi, k \in \psi \tag{3.24}
\]

\[
z_i(\beta_c) \leq N(\beta_c) d_i \quad \forall i \in \phi \tag{3.25}
\]

\[
z_i(\beta_c) \leq -N(\beta_c) d_i \quad \forall i \in \phi \tag{3.26}
\]

\[
A(1 - g_i(k)) + N(k, \beta_k) d_i \geq 0 \quad \forall i \in \phi, k \in \psi \tag{3.27}
\]

\[
\sum_{i=1}^{n} g_i(k) \geq 1 \quad \forall k \in \psi \tag{3.28}
\]

\[
\sum_{i=1}^{n} g_i(\beta_c) \geq 1 \tag{3.29}
\]

\[
M_i d_i \geq u_i \quad \forall i \in \phi, \ell \in \theta \tag{3.31}
\]

\[
\lambda_i^1 \leq d_i^1 \quad \forall c = \{x, y, z\}, i \in \phi \tag{3.32}
\]

\[
\lambda_i^1 \leq -d_i^1 \quad \forall c = \{x, y, z\}, i \in \phi \tag{3.33}
\]

\[
-(\lambda_i^1 + \lambda_i^2 + \lambda_i^3) \geq L \quad \forall i \in \phi \tag{3.34}
\]

\[
-(\lambda_i^1 + \lambda_i^2 + \lambda_i^3) \leq U \quad \forall i \in \phi \tag{3.35}
\]
\[ d_i \in \mathbb{R} \quad \forall i \in \phi \quad (3.36) \]
\[ g_i(k) \in \{0, 1\} \quad \forall i \in \phi, k \in \psi \quad (3.37) \]
\[ g_i(\beta_c) \in \{0, 1\} \quad \forall i \in \phi \quad (3.38) \]
\[ z_i(k) \in \mathbb{R} \quad \forall i \in \phi, k \in \psi \quad (3.39) \]
\[ z_i(\beta_c) \in \mathbb{R} \quad \forall i \in \phi \quad (3.40) \]
\[ \lambda_i \in \mathbb{R} \quad \forall c = \{x, y, z\}, i \in \phi \quad (3.41) \]

where Equations (3.22)–(3.41) is solved using an increasing number of parting directions \( d_i \) until the first instance that generates a solution, as described earlier and shown in Figure 6. The number of decision variables in the problem are \( 6n(p + 2) \), of which \( 3n(p + 1) \) are binary variables.

4. Draft angle and connectivity of multi-piece mould design

As discussed in Section 2.3, the MIP presented in Equations (3.22)–(3.41) and accompanying algorithm can be implemented using CPLEX. One benefit of our approach is that different draft angle requirements in the injection moulding process can be easily incorporated. Figure 7(a) shows various drafting types of a face \( F \) related to a parting direction \( d_i \). For a part to be injection moulded, its surfaces that are parallel to the paring direction are usually required to be drafted for an angle \( \gamma \) in order to ease the ejection of the part and reduce the damaging possibility of both parts and moulds. The minimal draft angle mainly depends on the moulding process and material. For some moulding processes such as urethane rubber moulding, a zero or even slightly negatively draft angle are acceptable due to the flexibility of the moulding material.

For a given minimal draft angle \( \gamma \), we can compute \( \tau = \sin(\gamma) \) and use it to replace 0 in Equations (3.27) and (3.29). Hence our optimisation formulation can be changed as:

\[ \Lambda(1 - g_i(k)) + N(k, \beta_c)d_i \geq \tau, \quad \forall i \in \phi, k \in \varphi, \text{ and} \]
\[ \Lambda(1 - g_i(\beta_c)) + N^c(\beta_c)d_i \geq \tau, \quad \forall i \in \phi. \]

Accordingly, the computed solution (if any) will satisfy the given draft angle requirements. In addition, any non-drafted or under-drafted surfaces can be identified if no solution is found for them. A simple test example (Test 2 in Section 6) is shown in Figure 7(b). Three draft angles (\( \gamma = 0, -5^\circ \) and \( 5^\circ \)) are applied to a cylindrical hole. The related CAD models are used in generating input to the MIP. For different \( \gamma \) and \( \tau \) values, the computed parting directions are shown in Table 1.

Notice, however, in our MIP formulation, only the demouldability requirement has been incorporated. We had difficulties in converting the face connectivity of a mould piece region into computable formulation; hence such connectivity has not been incorporated. Consequently, the optimal solutions given by solving the MIP are actually a lower bound on the MPMD. That is, without considering the connectivity of mould ...

![Figure 7. Different draft angle requirements.](image)

**Table 1.** The parting directions for different draft angle \( \gamma \) and \( \tau \).

<table>
<thead>
<tr>
<th>Draft angle ( \gamma )</th>
<th>Parting directions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (( \tau = 0.01 ))</td>
<td>No solution</td>
</tr>
<tr>
<td>( -5^\circ ) (( \tau = 0.0 ))</td>
<td>No solution</td>
</tr>
<tr>
<td>5(^\circ) (( \tau = 0.1 ))</td>
<td>No solution</td>
</tr>
<tr>
<td>(( \tau = 0.0 ))</td>
<td>( d_1 = (0, 1, 0), ) and ( d_2 = (0, -1, 0) )</td>
</tr>
<tr>
<td>(( \tau = -0.09 ))</td>
<td>( d_1 = (0, 1, 0), ) and ( d_2 = (0, -1, 0) )</td>
</tr>
<tr>
<td>(( \tau = 0.03 ))</td>
<td>( d_1 = (0, 1, 0), ) and ( d_2 = (0, -1, 0) )</td>
</tr>
</tbody>
</table>
piece regions, the minimum number of mould pieces is found to be \( n \) (e.g. \( n = 5 \)). Accordingly, it is impossible to find a better solution (i.e. \( n < 5 \)) after the connectivity of \( n \) mould piece regions has been considered.

Two examples of such connectivity problems are shown in Figure 8 with different solution strategies. (1) For a test part as shown in Figure 4, a minimum of three parting directions has been identified after solving the MIP (i.e. \( d_1, d_2 \) and \( d_3 \) as shown in Figure 8(a) for the mould piece regions \( R_1, R_2 \) and \( R_3 \), that will be constructed). Accordingly, three-mould piece regions \( (m_1, m_2, m_3) \) are required for them, respectively. However, for a concave region \( (CVR_1) \), its solution for the formulated problem in Section 3 is \( d_1 \) since it has the biggest projection areas in such a direction; nonetheless, \( CVR_1 \) is not connected to other regions that use the same parting direction (region \( R_1 \)). After checking its neighbours, \( CVR_1 \) is assigned to region \( R_2 \) which uses parting direction \( d_2 \). (2) A concave region \( (CVR_4^2) \) as shown in Figure 8(b) is not connected to other regions of \( CVR_4 \), which share the same parting direction \( d_4 \). Instead it connects to \( CVR_1 \) which use another parting direction \( d_1 \). Even though \( CVR_4^2 \) can be demoulded along parting direction \( d_4 \), the constructed mould piece \( M_4 \) will intersect the

Figure 8. Two examples of connectivity of mould piece regions.

Figure 9. An example of mould pieces by incorporating design heuristics.
constructed mould piece $M_1$ based on $CVR_1$. However, since $CVR_2$ cannot be demoulded along parting direction $d_1$, a new parting direction $d_5$ needs to be added for the region. Accordingly, a new mould piece $M_5$ is added, which is shown in Figure 10(d).

Based on a set of computed parting directions $d_i$, it is trivial to check the face connectivity between concave regions to identify the isolated regions (suppose $m$ of them is identified). Accordingly, we have an upper bound on the MPMD. That is, a solution must be able to be found by using $n + m$ mould pieces after considering both demouldability and face connectivity requirements. Hence a MPMD must be able to be found with a number of mould pieces that is between $[n, n + m]$.

5. Refining multi-piece mould design

Based on the computed parting directions, we can keep on refining the generated mould piece regions such that a mould design can be achieved that is closer to the lower bound $(n)$ instead of the upper bound $(n + m)$. This can be done based on various heuristics. For example, we can compute the projection areas of each concave region for the given parting directions (i.e. $\sum_{i=1}^{6} A_i (N_i \cdot d)$). An example of such results is shown in Table 2 for the test case in Figure 8(a). In the table, a mark ‘×’ is assigned to a region if the related parting direction cannot satisfy the demouldability of the region.

Hence, for $CVR_1$ that is identified as ‘isolated region’ in direction $d_1$, we can check the other directions $d_2$ and $d_3$. Since they are both demouldable and connected, we can reassign $CVR_1$ to another mould piece region instead of adding a new one. Notice, as shown in Table 2, there are several concave regions that have a unique assignment to a related parting direction. For example, $CVR_3$, $CVR_4$ and $CVR_8$ can only be assigned to $d_3$, $d_2$ and $d_1$, respectively. We call such concave regions as the core concave regions of the related mould piece regions. Their assignments will not be changed during the refining process while other concave regions and convex faces may be. In addition, we can also compute a connectivity table of all the changeable concave regions and convex faces based on the core concave regions.

In addition to demouldability and face connectivity, a wealth of mould design and manufacturing knowledge has been characterised into a set of heuristics. Some of the heuristics examples can be found in Hui (1996), Wong et al. (1998), Fu et al. (1999), Chen and Rosen (2002), Ye et al. (2004). These heuristics can also be considered in the process of

![Figure 10. A test example on arbitrary orientations.](image-url)

**Table 2.** The projection area of nine concave regions for Test 4.

<table>
<thead>
<tr>
<th>Parting directions</th>
<th>Concave region # (9 out of 12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_1 (0.0, -1)$</td>
<td>0 1 2 3 4 5 6 7 8</td>
</tr>
<tr>
<td>$d_2 (1.0,0)$</td>
<td>× 0.3 × × × 4 5 5 14</td>
</tr>
<tr>
<td>$d_3 (-1.0,0)$</td>
<td>0 0 0.5 × 0 2.5 × ×</td>
</tr>
</tbody>
</table>
Table 3. Experimental results on the minimum parting directions ($\tau = -0.001$).

<table>
<thead>
<tr>
<th>Test #</th>
<th>Screen capture</th>
<th>Total Tri #</th>
<th>Concave region #</th>
<th>Conve × face #</th>
<th>Resulted parting direction #</th>
<th>Running time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1" alt="Test 1 Screen Capture" /></td>
<td>12</td>
<td>0</td>
<td>12</td>
<td>$d_1, d_2, (0, 0, 1), (0, 0, -1)$</td>
<td>0.02</td>
</tr>
<tr>
<td>2</td>
<td><img src="image2" alt="Test 2 Screen Capture" /></td>
<td>220</td>
<td>1</td>
<td>66</td>
<td>$d_1, d_2, (0, 0, 1), (0, 0, -1)$</td>
<td>0.02</td>
</tr>
<tr>
<td>3</td>
<td><img src="image3" alt="Test 3 Screen Capture" /></td>
<td>64</td>
<td>3</td>
<td>40</td>
<td>$d_1, d_2, (0, 0, 1), (0, -1, 0)$</td>
<td>0.03</td>
</tr>
<tr>
<td>4</td>
<td><img src="image4" alt="Test 4 Screen Capture" /></td>
<td>108</td>
<td>12</td>
<td>68</td>
<td>$d_1, d_2, d_3, (0, 0, -1), (1, 0, 0), (-1, 0, 0)$</td>
<td>0.13</td>
</tr>
<tr>
<td>5</td>
<td><img src="image5" alt="Test 5 Screen Capture" /></td>
<td>1988</td>
<td>199</td>
<td>770</td>
<td>$d_1, d_2, d_3, (0, 0, 1), (0, 0, -1), (0, -1, 0)$</td>
<td>4.33</td>
</tr>
<tr>
<td>6</td>
<td><img src="image6" alt="Test 6 Screen Capture" /></td>
<td>528</td>
<td>86</td>
<td>250</td>
<td>$d_1, d_2, d_3, d_4, (0, 0, 0), (0, 0, -1), (-1, 0, 0), (1, 0, 0)$</td>
<td>0.73</td>
</tr>
<tr>
<td>7</td>
<td><img src="image7" alt="Test 7 Screen Capture" /></td>
<td>1020</td>
<td>105</td>
<td>456</td>
<td>$d_1, d_2, d_3, d_4, (0, 0, -1), (0, 0, 1), (0, -1, 0), (0, 1, 0)$</td>
<td>0.57</td>
</tr>
<tr>
<td>8</td>
<td><img src="image8" alt="Test 8 Screen Capture" /></td>
<td>2164</td>
<td>358</td>
<td>741</td>
<td>$d_1, d_2, d_3, (0, 0, -1), (0, 0, 1), (0, -1, 0), (0, 1, 0)$</td>
<td>4.02</td>
</tr>
<tr>
<td>9</td>
<td><img src="image9" alt="Test 9 Screen Capture" /></td>
<td>1300</td>
<td>173</td>
<td>726</td>
<td>$d_1, d_2, d_3, (0, 0, 1), (0, 1, 0), (0, -1, 0)$</td>
<td>3.93</td>
</tr>
<tr>
<td>10</td>
<td><img src="image10" alt="Test 10 Screen Capture" /></td>
<td>2760</td>
<td>516</td>
<td>801</td>
<td>$d_1, d_2, d_3, d_4, (0, 0, 1), (0, 0, -1), (-0.83, -0.1536, -0.0165), (0.984, -0.033, 0.049)$</td>
<td>18.78</td>
</tr>
</tbody>
</table>
refining mould piece regions. For example, it is desired to have a smooth parting line. Hence for a mould design result generated by our system, we can further refine it by changing the assignment of some faces in order to achieve a smoother parting line. Figure 9(a) shows such an example based on Test 4. The original parting lines are shown as blue lines; they can be changed to red lines to improve its smoothness. Accordingly, the constructed mould pieces can have better strength. Figure 9(b) shows the MPMD for regions \( R_1 - R_3 \) after the parting lines have been modified. The mould pieces \( M_1 - M_5 \) can be demoulded from the parting directions \( d_1 - d_3 \), respectively.

6. Experimental results

6.1. Orientations of input Computer-Aided Design models

We first tested our approach by using different orientations of an input CAD model. The test was based on the part as shown in Figure 5. For the original model that is aligned with the coordinate...
axes, two parting directions, $d_1$ and $d_2$, are identified in the Z and Y axes; accordingly two mould pieces, $M_1$ and $M_2$, can be constructed. In this test, we arbitrarily rotated the axis-aligned polygonal model into other orientations such that the normals of all the faces are not aligned with the coordinate axes (refer to Figure 10). We then formulate the MIP and applied the algorithm discussed in Section 3 to compute the minimum parting directions. The computed results are shown in Figure 10, which are consistent with the results for the original model. The test illustrates that the optimisation-based method is not sensitive to the orientations of the input CAD models.

6.2. Computing parting directions

The optimal parting directions based on the formulated MIP have been computed for a set of injection moulding parts with different complexities. A summary of the tested parts and related computed results is given in Table 3. A screen capture of the tested models and the related shape statistics are given in the table as well as the resulted parting directions. The presented running time is based on a 3GHz Intel Xeon CPU using CPLEX 9.0.

6.3. Constructing mould pieces

Based on the computed parting directions, we went through the mould piece construction process for some

Table 4. Numbers of infeasible region and related parting directions for different $\tau$ values.

<table>
<thead>
<tr>
<th>$\tau$ values</th>
<th>$-0.02$</th>
<th>$-0.1$</th>
<th>$-0.2$</th>
<th>$-0.5$</th>
<th>$-0.65$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infeasible concave region # for parting directions ($d_1, d_2, d_3$)</td>
<td>42</td>
<td>27</td>
<td>18</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Minimum parting direction #</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
test cases. Figure 11 shows the screen captures of the computing results for Test 6. The polygonal model in both top and bottom views are shown in Figure 1. Its 528 faces are classified into 86 concave regions and 250 convex faces, which are shown in Figure 11(a). We solved the linear MIP to identify parting directions \((d_1, d_2, d_3, d_4)\). Accordingly, four-mould piece regions \(R_1 - R_4\) are computed as shown in Figure 11(b). Due to the face connectivity problem as discussed in Section 4, an additional parting direction \(d_5\) was added. Accordingly, a mould piece region \(R_5\) and related mould piece \(M_5\) are constructed as shown in Figure 11(c). The resulting MPMD is shown in Figure 11(d). Physical experiments based on the mould design have also been performed by using a mask projection-based SLA machine and the urethane moulding (refer to a result as shown in Figure 1).

6.4. Mesh quality of input computer-aided design models and draft angle values

It was found in our tests that the mesh quality of the input CAD models and related \(\tau\) value settings (discussed in Section 4) play a large role in the computed parting directions. For example, Test 10 in Table 3, at first glance, seems only requiring three parting directions \((d_1, d_2, d_3, d_4)\) as shown in Table 3. Further investigation illustrates that the input polygonal model has poor mesh quality. In the three parting directions \(d_1 - d_3\), a set of small regions will not be able to be demoulded. Consequently, a fourth parting direction \(d_4\) is required. For example, when we set \(\tau = 0.2\), there are 18 concave regions that cannot be demoulded as shown in Figure 12. The magnified views of some of such regions are also shown in the figure. Obviously some of the surfaces have normals \((N_{x_1}, N_{y_1}, N_{z_1})\) such that \(N_{x_1}d_1 + N_{y_1}d_2 + N_{z_1}d_2 < \tau\).

We tested different \(\tau\) values for Test 10. As shown in Table 4, the number of infeasible regions for the given parting directions \(d_1 - d_5\) is different for various \(\tau\) values. Since our approach can identify such problematic regions, a design-for-manufacturing strategy can be developed to modify the identified regions such that the input CAD model is more suitable for the multi-piece injection moulding process.

7. Conclusion and future work

We present a novel approach to the MPMD for a production part. The methodology involves a mixed-integer programming problem that is primarily based on concave regions and convex faces of the production part as a set of basic elements. By using a state-of-the-art optimisation solver in conjunction with a decomposition algorithm, such a problem can be solved in reasonable time. Hence, the lower and upper bound on the number of mould pieces can be identified in a MPMD. Additional measures and insights for improving the generated multiple-piece mould design have also been discussed. A set of test examples were given and the experimental results demonstrate the effectiveness and efficiency of the approach.

Future work directions we are currently investigating include: (1) testing our approach by using CAD models with higher complexity; (2) investigating a design-for-manufacturing strategy to handle input CAD models with small features that are not suitable for the injection moulding process and (3) performing more physical experiments based on additive manufacturing to verify the effectiveness of the multi-piece injection moulding process for limited-run productions. In addition, some challenging problems that remain to be addressed within the scope of MPMD include formulating the face/region connectivity into the mix-integer program, and how to automatically refine a set of freeform surfaces based on the multi-piece injection moulding process.

Acknowledgments

The authors acknowledge Professor Satyandra K. Gupta at the University of Maryland for providing them the part models of Tests 4-8 and 10. The authors are also grateful to the comments and suggestions from the anonymous reviewers.

References


