Algorithmic Transformation of Semantic Face-Adjacency Graphs into Kinematically Valid Precedence DAGs for Salvagnini P4 Process Planning

1. Introduction: The Geometric Imperative in Automated Process Planning

The domain of Computer-Aided Process Planning (CAPP) for sheet metal fabrication is currently undergoing a paradigm shift, moving from heuristic-based expert systems toward geometrically rigorous, algorithmic solvers. This shift is necessitated by the increasing complexity of component geometries and the sophisticated kinematics of modern forming hardware, specifically the panel bender. Unlike the traditional press brake, which relies on a vertical ram and often manual or robotic manipulation that allows for significant degrees of freedom in part re-orientation, the panel bender—exemplified by the Salvagnini P4 architecture—operates within a tightly constrained kinematic envelope. The machine utilizes a horizontal reference plane, an automated manipulator for part positioning, and a universal bending unit that interpolates to form flanges in both positive and negative directions.

In this context, Phase 2 of the CAPP development lifecycle—**Constraint Identification and Precedence Graph Construction**—is the critical bridge between the static design intent and the dynamic execution of manufacturing. The input to this phase is a Semantic Face-Adjacency Graph (FAG), a topological data structure derived from Phase 1 feature recognition. The FAG represents the "what" of the part: which faces are connected, the angles of these connections, and the geometric attributes of the constituent surfaces. The objective of Phase 2 is to transform this static representation into a Directed Acyclic Graph (DAG) of dependencies, representing the "how" and "when." This DAG must encode the hard constraints that dictate the immutable order of operations, ensuring that the generated process plan is not merely theoretically sorted but physically executable on the specific topology of a Salvagnini P4 machine.

The central hypothesis driving the algorithmic approach detailed in this report is that the most robust method for generating valid bending sequences is the **Reverse Unfolding Simulation**. By mathematically simulating the disassembly of the folded part into a flat pattern, we can identify geometric interferences that would prevent the inverse operation (bending) from occurring. If the removal of a flange via unfolding results in a collision with an adjacent geometric feature, a precedence constraint is established: the obstructing feature must be removed first in the unfolding sequence, which implies it must be formed last in the forward manufacturing sequence. This report provides an exhaustive analysis of the geometric reasoning required to detect these interferences, specifically utilizing **Ray Casting algorithms** and **Separating Axis Theorem (SAT)** to resolve complex corner topologies and hierarchical tab-on-flange relationships.

2. Machine Topology and Kinematic Constraints: The Salvagnini P4 Ecosystem

To accurately define the constraint field within which the CAPP system must operate, it is essential to rigorously parameterize the target hardware. The Salvagnini P4 Panel Bender is not a generic forming tool; it is a system of interacting kinematic modules, each imposing distinct limitations on the process plan. The constraints identified here are "hard" constraints; any process plan violating them results in a collision or a failure to form, and thus they must be encoded as blocking edges in the Precedence Graph.

2.1 The Manipulator and Horizontal Kinematics

The primary distinction of the P4 architecture is its horizontal material handling. The sheet metal blank rests on a horizontal table and is gripped by a manipulator that positions the part along the X and Y axes and provides rotation in the horizontal plane.[1] Unlike a press brake, where the part can be flipped over (Z-axis rotation) by an operator to access reverse bends, the P4 manipulator maintains the sheet in a constant orientation relative to gravity, although the bending unit itself can form upwards and downwards.[2]

The manipulator imposes a **Gripping Constraint**. For every bend operation, there must exist a secure, flat surface area on the remaining unbent portion of the sheet (or previously bent flanges that are coplanar with the table) for the manipulator to engage. In the context of the Reverse Unfolding logic, this means that as we unfold the part, we must ensure that at every stage, the "flat" surface area increases or remains sufficient for gripping. If a sequence of unfolds results in a geometry where the center of mass shifts significantly or the gripping surface is occluded by a "downward" flange, that sequence is kinematically invalid.

2.2 The Bending Unit and Throat Constraints

The bending unit of the P4 consists of a C-frame structure housing the upper and lower bending blades. These blades interpolate to perform the bending action. The physical dimensions of this unit impose absolute limits on the geometry of the flanges.

- **Maximum Bending Length (L_{max}):** This is the width of the machine's throat. If a bend line exceeds this length, it cannot be processed in a single stroke.
- **Maximum Bending Height (H_{max}):** This is the vertical clearance available for a flange to rise (in a negative bend) or descend (in a positive bend) without colliding with the

machine frame.

The research data indicates specific limits for various models in the P4 lineup, which must be loaded into the CAPP constraint engine as boundary parameters:

Model	Max Bending Length (mm)	Max Bending Height (mm)	Min Thickness (mm)	Max Thickness (Steel 410 N/mm²)
P4-2120	2180	203	0.5	3.2
P4-2226	2200	260	0.5	3.2
P4-2535	2500	350	0.5	2.5
P4-2720	2750	203	0.5	3.2
P4-3126	3100	260	0.5	3.2

Table 1: Kinematic Limits of Salvagnini P4 Models. Note the variation in bending height, which is a critical constraint for deep-box applications.[3]

These limits generate **Geometric Cumulative Constraints**. A single flange might be 100mm high, well within the limit. However, if a U-shaped sequence is formed, the *cumulative* height of the part as it rotates might exceed the opening height (H_{max}). The Precedence Graph must therefore prune any branch where the intermediate part geometry violates these envelopes.

2.3 Automatic Blankholder (ABA) and Tooling Composition

The Automatic Blankholder Adjustment (ABA) is a defining feature of the P4, allowing for batch-one production by automatically resizing the hold-down tool to match the bend line length.[1] However, this automation introduces a **Tool Composition Constraint**. The blankholder is composed of discrete segments (e.g., 5mm, 10mm, 20mm widths).

- **Constraint:** The bend line length must be composable from the available segments.
- **Clearance:** There must be sufficient clearance on the "Base" face (the face being held) to accommodate the footprint of the blankholder without interference from adjacent flanges. If an adjacent flange (Flange B) is already bent and leans inward (acute angle), it might occupy the volume required by the blankholder to grip the Base for bending Flange

A. This dictates that Flange A must be bent *before* Flange B. In Reverse logic: Unfold B *before* Unfold A (to clear the space).

2.4 Auxiliary Tooling (CLA)

For complex geometries, specifically tabs or interrupted bend lines, the P4 utilizes **CLA (Auxiliary)** blades. These are modular blades that can be engaged or disengaged to bend sections shorter than the full side length.[4]

- **Constraint:** CLA tools have a maximum length (e.g., 1000mm) and specific engagement kinematics. The use of CLA often implies a specific sequence relative to standard bends. For instance, standard long bends are typically performed first to establish the global shape, with CLA operations following for local features, although geometric interference (tabs) often inverts this for feasibility, as will be discussed in Section 6.

3. Mathematical Formalism: From Face-Adjacency to Precedence

The transformation from Phase 1 to Phase 2 is a transformation from topology to chronology. We move from a graph describing *connectivity* to a graph describing *causality*.

3.1 The Face-Adjacency Graph (FAG)

The input FAG is defined as a graph $G = (V, E)$ where:

- $V = \{f_1, f_2, \dots, f_n\}$ is the set of planar faces. Each face f_i is a node carrying attributes such as Area, Normal Vector ($\hat{n}$), and a Boundary Representation (B-Rep) loop of vertices.[6]
- $E = \{e_{ij}\}$ is the set of edges representing connections between faces. An edge e_{ij} exists if face f_i shares a geometric boundary with f_j.
  - **Attributes of E:**
    - **Type:** Bend (Plastic hinge) or Virtual (Coplanar adjacency).
    - **Angle (θ):** The bend angle (e.g., 90°, -45°).
    - **Radius (R):** Inside bend radius.
    - **Convexity:** Mountain (+/Up) or Valley (-/Down).

This structure is static; it represents the final state of the part.[7]

3.2 The Precedence Directed Acyclic Graph (DAG)

The target output is the DAG $D = (N, A)$ where:

- N: The set of Nodes, where each node n_k corresponds to a **Bend Operation**. Note the dimensional shift: An *Edge* in the FAG becomes a *Node* in the DAG.
- A: The set of Arcs. A directed arc (n_p, n_q) exists if and only if Operation n_p must strictly precede Operation n_q.
  - **Interpretation:** In the Forward manufacturing sequence, n_p is bent before n_q. In the Reverse Unfolding simulation, this implies n_q is unfolded before n_p.

The construction of A is the primary output of this report. We seek to find all pairs (n_p, n_q) where a **Hard Constraint** exists.

4. Geometric Reasoning Algorithm: Reverse Unfolding Simulation

The determination of constraints is not a heuristic process but a geometric simulation. We employ a **Reverse Unfolding** strategy. We start with the fully folded 3D part and attempt to "peel" faces flat onto the reference plane.

4.1 The Simulation Loop

The algorithm proceeds iteratively:

1. **State Initialization (S_0):** The part is in its final folded 3D configuration.
2. **Candidate Identification:** Identify all bends that are currently "active" (i.e., not yet flat).
3. **Tentative Unfold:** For each candidate bend b, simulate the kinematic rotation of the associated flange f by angle $-\theta$ around the bend axis.
4. **Interference Check:** Check if the sweeping volume of f intersects with any other non-flat face in the assembly.
5. **Constraint Generation:**
   - If Unfolding b causes a collision with face g (where face g is controlled by bend c), then b is **blocked**.
   - To make b unblockable, face g must be removed (unfolded) first.
   - **Reverse Precedence:** Unfold $c \rightarrow$ Unfold b.

- **Forward Precedence:** Bend $b \rightarrow$ Bend c.
- *Note:* This seemingly counter-intuitive inversion is the core of the logic. If B blocks A's path to unfolding, B is "in the way." B must be moved out of the way (unfolded) before A can move. In assembly terms, A is put in place, and then B is put in place, blocking A's removal.

4.2 Matrix Transformation for Unfolding

The unfolding of a flange is modeled using Homogeneous Transformation Matrices. Let face F_{child} be connected to F_{parent} via a bend axis defined by point P and unit vector $\hat{u}$. The unfolding rotation $R(-\theta)$ is applied to all vertices of F_{child} and its descendants (the kinematic chain).

$$M_{unfold} = T(P) \cdot R_{\hat{u}}(-\theta) \cdot T(-P)$$

where $T(P)$ is the translation to the origin. This matrix allows us to compute the instantaneous position of the flange at any increment of the unfolding process, which is critical for the Ray Casting check.

5. Algorithmic Collision Detection: Ray Casting and SAT

The user query specifically requests the implementation of a **Ray Casting** rule for overlap detection. This section defines that logic and contextualizes it within a broader collision detection framework suitable for industrial CAPP.

5.1 Overlap Detection Logic: The Ray Casting Theorem

The core geometric query is: *Does the moving flange penetrate the static volume of another flange?*

While bounding boxes (AABB) provide a fast filter, they yield false positives for non-orthogonal flanges. Ray Casting provides the necessary precision for "Closed Corners" and "Tabs."

5.1.1 The Rule Formulation

Constraint Logic Rule:

Let F_{moving} be the flange being tentatively unfolded.

Let F_{static} be any other face in the current folded state.

Construct a set of Rays Ψ originating from the boundary vertices and surface sampling points of F_{moving} .

The direction vectors d of these rays correspond to the instantaneous tangential velocity of the unfolding rotation (or simplified sweeping vectors along the arc).

IF $\exists ray \in \Psi$ such that $Intersect(ray, F_{static}) = TRUE$,

THEN a Hard Constraint exists.

IMPLICATION: The geometric feature F_{static} physically obstructs the path of F_{moving} . Therefore, F_{static} must be removed (unfolded) before F_{moving} is processed.

DAG Edge: Node(F_{static}) $\rightarrow$ Node(F_{moving}).

5.1.2 Implementation of Ray Casting

The Ray Casting algorithm is implemented as follows [9]:

1. **Discretization:** The boundary of F_{moving} is discretized into a set of vertices $\{v_1, v_2, \ldots, v_k\}$. Additional internal points are sampled if the face is large to detect mid-face collisions.
2. **Vector Generation:** For a bend axis $\hat{a}$ and rotation center C, the motion path of a vertex v_i is a circle. The collision check can be approximated by casting a ray along the tangent $t = \hat{a} \times (v_i - C)$.
3. **Intersection Test:** We compute the intersection of the ray $R(t) = v_i + tt$ with the plane of F_{static} .
   - Calculate intersection parameter t.
   - If $0 < t < ArcLength$, calculate the intersection point P_{hit} .

    - Perform a Point-in-Polygon test (Jordan Curve Theorem) to see if P_{hit} lies within the boundaries of F_{static}.
4. **Tolerance:** A collision is flagged only if the intersection depth exceeds a manufacturing tolerance ϵ (e.g., 0.1mm), to account for idealized CAD zero-thickness geometry vs. real-world gaps.

5.2 Separating Axis Theorem (SAT) for Convex Hulls

While Ray Casting is precise for "point" interference, the **Separating Axis Theorem (SAT)** is superior for detecting volumetric overlap, particularly for Lapped Corners where flanges essentially occupy the same space.[12]

- **Theorem:** Two convex polyhedra do not overlap if and only if there exists a separating axis onto which their projections do not overlap.
- **Axes to Test:**
    - Face normals of Polyhedron A.
    - Face normals of Polyhedron B.
    - Cross products of edges of A and edges of B.
- **Application in CAPP:** SAT acts as the "Broad Phase" filter. Before running the expensive Ray Cast (Narrow Phase) on every vertex, we run SAT on the convex hulls of the flanges. If SAT returns "No Overlap," we skip the Ray Cast. This optimization is crucial for performance when processing complex assemblies with dozens of bends.[14]

5.3 Bounding Volume Hierarchies (BVH)

To further optimize the collision detection for parts with high face counts ($N > 50$), we employ a **Bounding Volume Hierarchy (BVH)**.[13] Faces are grouped into clusters, and their Axis-Aligned Bounding Boxes (AABBs) are checked first. Only if parent boxes intersect do we descend to check the child faces. This reduces the complexity from $O(N^2)$ to $O(N \log N)$ for global collision checks.

6. Topology-Specific Constraints: Corner Logic

The generic collision algorithms described above must be applied to specific topological patterns found in sheet metal. The most critical of these is the **Corner**—the convergence of two or more flanges.

6.1 Classification of Corner Topologies

We classify corners based on the geometric relationship between the converging flanges

(Flange A and Flange B).[15]

Corner Type	Description	Gap Distance	Constraint Implication
Open Corner	Flanges do not touch.	$>$ Thickness	**Soft Constraint.** Independent unfolding is usually possible.
Butt Corner	Flanges meet edge-to-edge.	$\approx$	**Kinematic Risk.** Simultaneous bending preferred, or strictly ordered if relief notches are absent.
Lap Corner	One flange overlaps the other.	Negative (Intersection)	**Hard Constraint.** The "Top" flange must be unfolded first.
Miter Corner	Flanges meet at 45^c angle.	$\approx$	**Hard Constraint.** Requires strict ordering if asymmetric.

Table 2: Sheet Metal Corner Topology Classification and Constraint Implications.

6.2 Closed Corner (Butt/Lap) Logic

In a **Closed Corner**, the sequence is dictated by the physical obstruction.

- **Overlap Detection:** We detect "Closed Corners" by identifying cycles in the FAG (Base $\rightarrow$ Flange A $\rightarrow$ Flange B $\rightarrow$ Base) where the connection between A and B is "Virtual" but the distance is minimal.
- **The "Top" Flange:** In a Lap joint, one flange is physically on top of the other relative to the bend direction.
  - **Detection:** By projecting the volume of Flange A onto Flange B along the bend normal.
  - **Rule:** The Flange that is "on top" (let's say A) covers B. In Reverse Unfolding, A must

be lifted (unfolded) before B can be unfolded.

  - **Precedence:** Unfold A $\rightarrow$ Unfold B.
  - **Forward Equivalence:** Bend B $\rightarrow$ Bend A (Bending B first puts it "underneath," then Bending A folds it "on top").

7. Hierarchical Constraints: The Tab-on-Flange Problem

A specific requirement of the user query is to address the **Tab-on-Flange** constraint. This represents a hierarchical "Parent-Child" relationship in the graph.

7.1 Topological Definition

- **Parent:** The main Flange (F) attached to the Base.
- **Child:** A small Tab (T) attached to an edge of F.
- **Graph:** Base $\leftrightarrow$ Flange $\leftrightarrow$ Tab.

7.2 The Constraint: Child Before Parent (Reverse)

The analysis of swept volumes dictates a strict precedence rule for this topology.

Hypothesis: Why must the tab (T) usually be unfolded before the main flange (F) in a reverse simulation?

Analysis:

1. **Scenario A: Unfold Parent First.**

   Consider the state where F is bent 90° relative to Base, and T is bent 90° relative to F (parallel to Base, creating a return or "C" shape).

   If we attempt to rotate F around the Base-Flange axis:

   - The Tab T moves with F as a rigid body.
   - Because T is protruding from F, it sweeps a large radius. The "tip" of the tab traces a path that is highly likely to intersect with the machine table, the blankholder, or the adjacent flanges.[18]
   - Furthermore, mechanically, the blankholder cannot grip the Base effectively if the "return" tab is crashing into the tooling during the swing.
2. **Scenario B: Unfold Child First.**

We first rotate T around the Flange-Tab axis until T is coplanar with F.

- Now, F and T form a single flat plane.
- We then rotate this combined flat plane ($F + T$) around the Base-Flange axis.
- **Result:** The swept volume is minimized to the thickness of the sheet. There are no protruding features to catch on tooling. The kinematic path is clean.

Conclusion:

In Reverse Unfolding, the **Child (Tab)** must be unfolded before the **Parent (Flange)** to resolve the geometry into a planar state that facilitates the subsequent parent operation.

> **Constraint:** Edge(Tab $\rightarrow$ Flange) in Reverse Logic.
>
> **Implication:** In Forward Manufacturing, the Flange is typically bent first, and the Tab is bent second (often using CLA tools).

7.3 Integration with P4 Tooling (CLA)

This logic aligns perfectly with the Salvagnini P4's tooling capabilities.

- The P4 uses standard blades for the main flanges (Parent).
- It uses **CLA (Auxiliary)** blades for tabs and interrupted bends.[4]
- The kinematics of engaging the CLA blades usually require the main flange to be already formed and held in position, or for the tab to be formed as a secondary operation after the main box shape is established.
- Exceptions exist for "internal" tabs (tabs inside a cutout), but for "external" tabs on flange edges, the **Parent** $\rightarrow$ **Child (Forward)** sequence is the standard hard constraint derived from both geometric collision avoidance and tooling availability.

8. Precedence Graph Construction and Cycle Resolution

Having identified the constraints via geometric simulation, we proceed to construct the DAG.

8.1 Construction Algorithm

The algorithm for Phase 2 is summarized as follows:

1. **Initialize DAG:** Create a node for every bend edge in the FAG.
2. **Parent-Child Pass:**
   - Traverse FAG to identify Tab-Flange hierarchies.
   - Add Hard Constraints: Unfold Child $\rightarrow$ Unfold Parent.

3. **Reverse Unfolding Simulation Pass:**
   - Set $State = Folded$.
   - Loop until $State = Flat$:
     - Get set of candidate bends C (bends with no currently bent dependents).
     - For each $b \in C$:
       - **Ray Cast Test:** Check collision of b against all other non-flat faces.
       - If b collides with face f (controlled by bend d):
         - Add Edge $d \to b$ to DAG. (Obstacle d precedes b).
       - **Kinematic Test:** Check if b's height/length fits P4 limits.
         - If limit exceeded, mark b as "Late Stage Only" (must be unfolded early in reverse).
4. **Cycle Detection:**
   - Run a Topological Sort (e.g., Kahn's Algorithm) on the DAG.
   - If the graph contains a cycle (e.g., A blocks B, and B blocks A), the part is **Infeasible** for standard bending.
   - *Resolution:* Flag the cycle for "Simultaneous Bending" (Phase 3 optimization) or return error to CAD design.

8.2 Handling Infeasible Geometries

In cases where a "Closed Corner" results in a mutual interlock (e.g., a dovetail joint or zero-clearance miter without relief), the DAG will contain a cycle. For the P4, simultaneous bending of two sides is generally not possible (unlike a custom die on a press brake). Therefore, cycle detection in Phase 2 acts as a **Design for Manufacturability (DFM)** filter. The system must report the specific faces causing the cycle to the user, suggesting the addition of a corner relief or a change in gap tolerance.

9. Conclusion

The transformation of the Face-Adjacency Graph into a Precedence DAG is the foundational step in automating process planning for the Salvagnini P4. By strictly adhering to a **Reverse Unfolding Logic**, we convert the complex problem of "how to assemble" into the more tractable problem of "how to disassemble."

This report has defined the rigorous geometric rules required for this transformation:

1. **Ray Casting** acts as the primary arbiter of geometric feasibility, identifying "Closed Corner" and "Lap" constraints where one flange obstructs another.

2. **Swept Volume Minimization** dictates the hierarchical precedence of Tabs over Flanges, enforcing a "Child-First" unfolding sequence.
3. **Kinematic Envelopes** of the P4 machine (throat depth, bending height) impose cumulative geometric constraints that prune the search space of valid sequences.

The resulting DAG is not merely a suggested order; it is a mathematical encoding of the physical reality of the manufacturing process. It serves as the valid input for Phase 3, where soft constraints (tool changes, rotation minimization) will be applied to optimize the efficiency of the valid sequences identified here.

(This concludes the Phase 2 Research Report. All analysis is derived from the provided snippets and standard geometric algorithms in computational manufacturing.)

Works cited

1. P4 Abt: Range of Automatic Panel Benders. More Flexible, More Versatile, More Productive | PDF | Sheet Metal - Scribd, accessed February 2, 2026, https://www.scribd.com/document/342321104/P4-ENG-3962020307-1006
2. Panel Bending Tools - JEELIX, accessed February 2, 2026, https://www.jeelix.com/panel-bending-tools/
3. Automatic Sheet Metal Bending Machine - Salvagnini P4, accessed February 2, 2026, https://www.salvagninigroup.com/en-US/products/panel-benders/P4
4. p4-benders.pdf - Empire Machinery, accessed February 2, 2026, https://www.empire-machinery.com/wp-content/uploads/2021/06/p4-benders.pdf
5. Panel Bender -P4 - MachineryHost, accessed February 2, 2026, https://f.machineryhost.com/71b2fab0316bfa7fa24ce0a0e5460bad/Brochure%20-%20Panel%20Bender%20-P4.pdf
6. Dynamic Cost Modelling Through Factory Simulation - ePrints Soton, accessed February 2, 2026, https://eprints.soton.ac.uk/413583/1/Cooper_Thesis_2017_April_20.pdf
7. a 3-D model for sheet metal enclosure. b FAG. c WFAG - ResearchGate, accessed February 2, 2026, https://www.researchgate.net/figure/a-3-D-model-for-sheet-metal-enclosure-b-FAG-c-WFAG_fig4_257337897
8. A hybrid learning framework for manufacturing feature recognition using graph neural networks | Request PDF - ResearchGate, accessed February 2, 2026, https://www.researchgate.net/publication/366019541_A_hybrid_learning_framework_for_manufacturing_feature_recognition_using_graph_neural_networks
9. Bending Simulation Framework for Rapid Feasibility Checks of Sheet Metal Parts - CAD Journal, accessed February 2, 2026, https://www.cad-journal.net/files/vol_23/CAD_23(1)_2026_85-100.pdf
10. Solid Modeling - Spatial Automation Laboratory - University of Wisconsin–Madison, accessed February 2, 2026,

https://spatial.engr.wisc.edu/wp-content/uploads/sites/715/2014/04/2001-2.pdf
11. Ray-casting algorithm - Rosetta Code, accessed February 2, 2026, https://rosettacode.org/wiki/Ray-casting_algorithm
12. The sweep plane algorithm for global collision detection with workpiece geometry update for five-axis NC machining | Request PDF - ResearchGate, accessed February 2, 2026, https://www.researchgate.net/publication/223631480_The_sweep_plane_algorithm_for_global_collision_detection_with_workpiece_geometry_update_for_five-axis_NC_machining
13. Collision detection during planning for sheet metal bending by bounding volume hierarchy approaches | Request PDF - ResearchGate, accessed February 2, 2026, https://www.researchgate.net/publication/324742687_Collision_detection_during_planning_for_sheet_metal_bending_by_bounding_volume_hierarchy_approaches
14. Physics Manager - Unity - Manual, accessed February 2, 2026, https://docs.unity3d.com/2017.4/Documentation/Manual/class-PhysicsManager.html
15. How to Create an Exact Closed Corner on a Sheet Metal - LearnVern, accessed February 2, 2026, https://learnvern.com/solidworks-tutorial/closed-corners-rip-and-sketched-bend-in-sheet-metal-solidworks
16. Closed Corner- CSWP Sheet Metal - 3D Engineer, accessed February 2, 2026, https://www.3dengr.com/closed-corner-cswp-sheet-metal.html
17. Closed Corners and the Closed Corner PropertyManager - 2024 - Solidworks Help, accessed February 2, 2026, https://help.solidworks.com/2024/english/SolidWorks/sldworks/c_closed_corners.htm
18. CE Tooling - H&O Die Supply, Inc., accessed February 2, 2026, http://www.hodie.com/Catalog/Catalogs/CE_Tooling/CE-Sec1TechnicalReference.pdf
19. Comprehensive Guide to Bending Tools and Techniques - JEELIX, accessed February 2, 2026, https://www.jeelix.com/guide-to-bending-tools-and-techniques/