

Algorithmic Framework for Phase 2 CAPP: Constraint Identification and Precedence Graph Construction for Salvagnini P4 Panel Bending

Executive Summary

This engineering report constitutes the definitive technical specification for Phase 2 of the Computer-Aided Process Planning (CAPP) system development, specifically tailored for the Salvagnini P4 Panel Bender equipped with Automatic Blankholder Adjustment (ABA). Following the successful completion of Phase 1—which focused on geometric parsing and feature recognition—Phase 2 addresses the critical challenge of **Constraint Identification and Precedence Graph Construction**.

The primary objective of this phase is to transition from a static geometric model (B-rep) to a dynamic, topologically valid manufacturing plan. This requires a rigorous algorithmic approach to model the machine's unique kinematic capabilities, particularly the interpolated blade movements and the variable-width ABA tooling. Unlike conventional press brakes, the P4 operates with a high degree of automation that removes the human operator from the immediate process loop, thereby necessitating a CAPP system with near-perfect reliability in collision detection and feasibility analysis.¹

This report provides a deep dive into the mathematical modeling of the **Face-Adjacency Graph (FAG)** as the primary data structure for topological reasoning. It details the logic required to solve the **Box Closing Problem** via a modified Subset Sum algorithm that accounts for ABA segment granularity and "masked time" setup capabilities.³ Furthermore, it introduces a **Swept Volume** collision detection framework to handle complex **Internal Flange** and **C-Channel** geometries, ensuring that the return flanges do not interfere with the static blankholder or the moving blades during the bending cycle.⁵

The output of the algorithms described herein is a Directed Acyclic Graph (DAG) of dependencies—a Precedence Graph—that encodes all hard geometric and kinematic constraints. This graph serves as the input for Phase 3 (Operation Sequencing and Optimization), ensuring that the final generated NC code is not only efficient but physically executable on the specific P4 configuration (e.g., P4-2116, P4-3125).⁷

1. System Architecture and Kinematic Constraints

To engineer robust algorithms for the Salvagnini P4, one must first effectively model the physical machine as a system of kinematic constraints. The P4 is fundamentally different from a press brake; it is a semi-autonomous bending cell where the sheet is manipulated by a computer-controlled pincer system and formed by oscillating blades.⁸ This architecture introduces specific constraints that the CAPP system must enforce mathematically.

1.1 The P4 Kinematic Chain vs. Press Brakes

In a traditional press brake, the operator acts as the sophisticated sensor and manipulator, adjusting the part's position and avoiding collisions intuitively. In the P4 architecture, this responsibility shifts entirely to the algorithm. The machine consists of four primary interacting sub-systems that define the constraint space:

1. **The Manipulator (Pincer/Rotator):** This unit grips the sheet and moves it in the $X - Y$ plane. It also rotates the sheet by θ (typically $0^\circ, 90^\circ, 180^\circ, 270^\circ$).
 - *Constraint: The Grasping Problem.* For every operation O_i , there must exist a flat, accessible surface area A_{grasp} sufficient for the pincer to hold the part securely against the bending force F_{bend} .³
2. **The Blankholder (ABA):** This tool clamps the sheet against the counter-blade. Its width W_{tool} is variable.
 - *Constraint: The Enclosure Problem.* The tool must geometrically fit inside the profile of the part (e.g., inside a box) and must be extractable after the bend is complete.¹¹
3. **The Bending Unit (Blades):** Upper and lower blades move in interpolated paths to fold the metal.
 - *Constraint: The Collision Problem.* The trajectory of the blade $T_{blade}(t)$ must not intersect with any previously formed flanges of the part.²
4. **The Counter-Blade:** A static support that creates the reaction force.

Understanding this chain is vital because a violation in any one subsystem renders the entire process plan invalid. For instance, a sequence might be geometrically valid for the blades (no collision) but invalid for the manipulator (no place to hold the part).

1.2 Automatic Blankholder Adjustment (ABA) Mechanics

The ABA system is the technological cornerstone that enables "Batch One" production. Unlike manual tool changes that can take 15-30 minutes, the ABA adjusts the tool width in "masked time"—meaning the adjustment happens while the manipulator is repositioning the sheet, effectively reducing setup time to zero.³

However, the ABA is not infinitely variable. It is composed of discrete segments. The

algorithmic implication is that the tool width W_{tool} is a discrete variable, not a continuous one.

- **Granularity:** The P4 ABA typically adjusts in steps (e.g., 5mm steps) or by combining specific modular segments (e.g., 25mm, 40mm, 50mm, 100mm).⁴
- **Composition:** The CAPP system must solve a combinatorial problem to determine if a set of available segments can sum up to a target width W_{target} that fits the constraint $L_{min} \leq W_{tool} \leq L_{max}$.

The ABA constraints are particularly unforgiving in **Box Closing** scenarios. If the planner assumes a continuous tool width and specifies a width of 453mm, but the ABA can only achieve 450mm or 455mm due to segment availability, the 455mm tool might collide with the side flanges, while the 450mm tool might leave an unacceptable unsupported gap near the corners.¹² Thus, the exact segment configuration logic must be replicated in the CAPP engine.

1.3 Machine Specifications and Hard Limits

The algorithms must be parameterized with the specific machine limits to be universal across the P4 product line. Key parameters from the research include:

Parameter	Symbol	Typical Value (P4-3125)	Algorithmic Implication
Max Bending Length	L_{max}	3125 mm (122")	Prunes edges in FAG where $L_{bend} > \quad$.
Max Bending Height	H_{max}	203 mm (8") - 254 mm (10")	Hard limit on the Z-height of the folded bounding box.
Throat Depth	D_{throat}	~300 mm (Configuration Dependent)	Limits the diagonal of the rotating part $Diag \leq \quad$ if rotation passes through throat.

Min Sheet Thickness	T_{min}	0.4 - 0.5 mm	Affects stiffness assumptions; thin sheets require more support (ABA width).
Max Bend Angle	θ_{lim}	$\pm 135^\circ$ (depends on material)	Determines the "Swept Volume" sector angle for collision checks.
ABA Segment Steps	Δ_{AB}	5 mm or Discrete List	Defines the domain of the valid tool width set S_{valid} .

These parameters serve as the "Boundary Conditions" for the feasibility solver. Any feature violating these (e.g., a 12-inch deep box on a P4-3125 with a 10-inch max height) allows the system to fail fast before attempting computationally expensive sequencing.¹¹

2. Mathematical Modeling: The Face-Adjacency Graph (FAG)

To perform topological reasoning, the raw B-rep geometry (vertices, edges, faces) must be abstracted into a structure that represents connectivity and manufacturing semantics. We utilize the **Face-Adjacency Graph (FAG)**, extended with specific attributes for sheet metal processing.

2.1 Formal Definition of the FAG

We define the Sheet Metal Part P as a graph $G = (V, E)$, where:

- $V = \{f_1, f_2, \dots, f_n\}$ is the set of nodes representing the planar faces of the sheet.
 - **Node Attributes:**
 - $Area(f_i)$: Surface area (crucial for manipulator grasping checks).
 - $Normal(f_i)$: Unit normal vector n_i in the flat pattern.
 - $Poly(f_i)$: The 2D polygon definition of the face boundary.
- $E = \{e_{ij}\}$ is the set of edges connecting adjacent faces f_i and f_j .

- **Edge Attributes:**

- $Line(e_{ij})$: The 3D line segment defining the bend axis.
- θ_{ij} : The bend angle (signed). $\theta > 0$ implies a "Positive" bend (up), $\theta < 0$ implies a "Negative" bend (down) relative to the P4's fixed horizontal plane.
- R_{ij} : The inside bend radius.
- L_{bend} : The length of the bend (used for tonnage and tool width calculations).
- $Type$: Enum {Standard, Hem, Curl, Offset}.

This graph structure is invariant to rigid body transformations, making it ideal for recognizing topological features regardless of the part's orientation in the CAD file.¹³

2.2 Feature Recognition via Subgraph Isomorphism

The power of the FAG lies in its ability to identify manufacturing features by detecting specific subgraph patterns. This is the first step in constraint identification.

2.2.1 The C-Channel (Return Flange) Pattern

A C-channel, which poses significant collision risks, is identified as a linear path of four faces in the FAG:

$$Path_{channel} = (f_1, f_2, f_3, f_4)$$

Where:

- Edges e_{12}, e_{23}, e_{34} are parallel.
- The bend angles $\theta_{12}, \theta_{23}, \theta_{34}$ have signs that create an enclosed shape (e.g., all positive for a generic C, or alternating for Z).
- Specifically for a **Return Flange collision**, we look for the pattern $(f_{flange}, f_{web}, f_{flange'}, f_{return})$.
- If we identify this subgraph, we tag edge $e_{web-flange}$ with a PotentialCollision: InternalFlange attribute.

2.2.2 The Box Corner Pattern

A box corner represents a "cycle" in the spatial adjacency, even though the FAG is a tree (for a manifold sheet without holes). In the flat state, faces f_1 (side 1) and f_2 (side 2) are not connected. However, they share a common neighbor f_{base} .

- **Pattern:** f_{base} connected to f_1 and f_2 via orthogonal edges $e_{b1} \perp e_{b2}$.
- **Implication:** If both e_{b1} and e_{b2} are bent upwards, f_1 and f_2 will become spatially adjacent. This identifies a **Box Corner**.
- **Constraint:** This flags the need for "Box Closing" logic. The order of bending e_{b1} and e_{b2} matters less than the order of bending the *other* parallel sides if the box is 4-sided.

2.3 The Folded State Vector

Process planning is a search through the state space of the part. We define the state of the part at step k as a vector $S_k \in \{0, 1\}^{|E|}$, where:

- $S_k[i] = 0$: Edge e_i is unbent (flat).
- $S_k[i] = 1$: Edge e_i is bent to its final angle θ_{final} .
- (Advanced implementations may use scalar values for partial bends, but binary is sufficient for binary precedence logic).

The CAPP solver's goal is to find a permutation of edges $\pi = (e_1, e_2, \dots, e_m)$ such that the transition from state S_k to S_{k+1} is valid for all k .

3. Algorithmic Core: Automatic Blankholder Adjustment (ABA) Logic

The ABA is the most critical constraint variable in the P4 system. Unlike a fixed-length tool on a press brake, the ABA adapts, but this adaptation is subject to strict combinatorial and geometric rules.

3.1 The "IsToolingValid" Predicate

The core function of the constraint engine is `IsToolingValid(Bend, State)`. This function returns True if and only if a valid configuration of ABA segments exists that:

1. **Fits:** $W_{tool} \leq W_{available}$ (Geometric constraint).
2. **Supports:** $W_{tool} \geq L_{bend} \times \alpha$ (Process constraint, where α is typically 0.7-1.0 depending on material stiffness).
3. **Exists:** The width W_{tool} can be formed by the available segments (Combinatorial

constraint).

3.2 Solving the Subset Sum Problem for ABA

The ABA tooling consists of a set of discrete segments. Let the available segment widths be defined by the multiset $S_{seg} = \{s_1, s_2, \dots, s_n\}$.

Common segment lists for a P4 might look like:

$$S_{seg} = \{5, 10, 20, 25, 30, 40, 50, 100, 200, \dots\} \text{ (mm)}$$

Some models utilize a "central" unit that expands, plus modular add-ons.⁸

For a required bend length L_{bend} , and taking into account the clearance gaps δ_{gap} on either side (required to avoid collision with side flanges), the target width range is:

$$W_{min} = \alpha \cdot L_{bend}$$

$$W_{max} = L_{bend} - 2 \cdot \delta_{gap}$$

The problem is to find a subset $C \subseteq S_{seg}$ such that:

$$W_{min} \leq \sum_{s \in C} s \leq W_{max}$$

This is a variation of the **Subset Sum Problem** (a special case of the Knapsack problem).¹²

While NP-complete in the general case, the instance size here is small ($n < 50$), and the values are integers (mm).

- **Optimization:** We use a dynamic programming approach with memoization or a greedy heuristic (since standard segments are often multiples of 5 or 10, greedy algorithms work well).
- **Salvagnini Specifics:** The ABA often adjusts in 5mm steps.⁴ This simplifies the problem to:

$$W_{achievable} = \{w \in \mathbb{Z} \mid w = 5k, k \in \mathbb{N}\}$$

However, exact segment composition is required when using specific setups like the "P-tool" or specialized corner segments.

Algorithm 1: ABA Validity Check

Function IsABAVValid(Bend b, CurrentState S):

1. Calculate Available Width (W_{\max}):

- Project side flanges of the current folded state S onto the bend line of b.
- Measure the gap between the projected flanges.
- $W_{\max} = \text{Gap} - 2 * \text{SafetyClearance}$ (typically 1.0mm).

2. Calculate Required Support (W_{\min}):

- $W_{\min} = \text{Length}(b) * \text{SupportRatio}$ (e.g., 0.8).

3. Combinatorial Check:

- Retrieve machine segment list L_{seg} .
- Find if there exists a subset sum W_{opt} from L_{seg} such that $W_{\min} \leq W_{\text{opt}} \leq W_{\max}$.
- IF solution exists:
Return TRUE, W_{opt}
- ELSE:
Return FALSE (Tooling Constraint Violation).

3.3 The "Masked Time" Cost Function

The P4 boasts "Zero Setup Time" due to ABA.³ However, this is only true if the setup change happens *during* the handling time (masked time). If the tool change is drastic (e.g., moving from 500mm to 2000mm width), it might exceed the masked window, adding to cycle time.

- **Implication for CAPP:** The Precedence Graph optimization (Phase 3) needs a cost function.
- $Cost(O_i, O_j) = f(|W_{\text{tool}}(O_i) - W_{\text{tool}}(O_j)|)$.
- Small width changes are free (masked). Large changes incur a penalty.
- This encourages the "**Group by Width**" heuristic in sequencing: perform all bends requiring width $W \approx 500\text{mm}$ together before resizing to $W \approx 800\text{mm}$.

4. Advanced Constraint Identification: The Box

Closing Problem

The **Box Closing Problem** is the classic feasibility test for panel bending. It arises when bending the final sides of a box, where the tool must fit *inside* the previously bent sides.

4.1 Geometric Derivation of the Box Constraint

Consider a rectangular box with sides S_1, S_2, S_3, S_4 .

Let S_1, S_3 be the "long" sides and S_2, S_4 be the "short" sides.

If we bend S_1 and S_3 first (forming a channel), then to bend S_2 , the tool must have width $W_{tool} < Distance(S_1, S_3)$.

- **Variables:**

- L_2 : Length of the bend for side 2.
- H_1, H_3 : Heights of sides 1 and 3 (already bent).
- T : Material thickness.
- R : Bend radius.

The distance between the *inner* surfaces of S_1 and S_3 is:

$$D_{inner} = L_2 - 2(R + T)$$

(Note: This depends on the specific unfolding algorithm and K-factor used, but generally the inner gap is the critical dimension).

The constraint is:

$$W_{ABA} + 2\delta_{safe} \leq L_2 - 2(R + T)$$

If L_2 is small, it may be impossible to form a tool small enough (min segment width limits).

If H_1, H_3 are very tall, they may collide with the tool holder body, not just the active segments.

4.2 Handling Return Flanges in Box Closing

The complexity increases exponentially if S_1 and S_3 have their own return flanges (forming a C-profile).

Now, the "Opening" at the top of the box is narrower than the base.

$$Opening = D_{inner} - (L_{return_1} + L_{return_3})$$

The ABA tool must be inserted into this cavity.

- **Insertion Logic:**

- Can the tool enter vertically? Only if $W_{tool} < Opening$.
- Can the tool enter horizontally (slide in)? Only if the box is open on the other axis (which it is, before S_2, S_4 are bent).

- **Extraction Logic:**

- After bending S_2 , the tool is trapped on three sides (S_1, S_2, S_3).
- To extract, it must retract (move away from S_2 in Y) and then lift (Z).
- If S_1, S_3 have return flanges, the vertical lift is blocked if $W_{tool} > Opening$.
- **The Trap:** If W_{tool} was wide enough to bend S_2 (close to D_{inner}), it is now wider than $Opening$.
- **Solution:** The ABA must *contract* (shrink width) while inside the box before extraction.
- **Constraint:** Does the specific P4 model support **In-Cycle ABA Contraction**?
 - Research suggests ABA setup happens *between* cycles or masked during loading.¹ It generally does *not* dynamically change width while clamped *under* a bend.
 - Therefore, this is a **Hard Constraint**: W_{tool} must be small enough to clear the return flanges *during the bend*, OR the return flanges must be bent *after* the box closing (which is usually topologically impossible).

This logic dictates the **Precedence Rule**:

- **Rule:** "Internal Flanges" (Return flanges on S_1, S_3) imply that S_1, S_3 should likely be bent *after* S_2, S_4 if feasible, OR special tooling (segmented with side reliefs) is required.

5. Collision Dynamics: Internal Flanges and Swept

Volumes

Collision detection in the P4 is non-trivial due to the complex motion of the workpiece. The "Internal Flange" collision is the most common failure mode for C-channels.

5.1 Kinematics of the Interpolated Bend

The P4 blades do not pivot; they trace a curve.² Let the blade profile be a polygon P_{blade} . Let the trajectory be a function $T(t) : \rightarrow \mathbb{R}^2$ (in the X-Z plane). The volume swept by the blade is $V_{blade} = \bigcup_t (P_{blade} + T(t))$.

Simultaneously, the part itself deforms.

Let the flange being bent be F_{flange} . It rotates about the bend center C by angle $\alpha(t)$.

The return flange F_{return} attached to F_{flange} also rotates.

$$Pos_{return}(t) = Rot(C, \alpha(t)) \cdot Pos_{return}(0)$$

5.2 The Swept Volume Collision Algorithm

We utilize the **Swept Volume** approach for collision detection.⁵ Instead of checking for collision at discrete time steps (which might miss intersections), we construct the swept volume of the hazard (the return flange) and intersect it with the static volumes (Blankholder, Upper Blade Holder).

5.2.1 Constructing the Swept Volume

For a return flange F_{return} (modeled as a rectangle), rotating 90° or 135° around the bend axis:

1. The vertices of F_{return} trace circular arcs.
2. The volume is a toroidal sector (or a prism with curved sides).
3. We approximate this by creating a **Convex Hull** of the start and end positions, plus intermediate samples to capture the curvature.

5.2.2 Intersection Checking

We define three Hazard Zones:

1. **Zone A (Tool Face):** The front face of the ABA blankholder.
2. **Zone B (Blade Holder):** The structural support of the upper blade (the "C" frame of the

blade unit).

3. **Zone C (Machine Throat):** The rear limit of the workspace.

Constraint Function CheckInternalCollision(Bend):

1. Define Safe Zone Polygon P_{safe} (Geometry of the ABA cross-section).
 - Note: ABA segments often have "cutouts" (horns) allowing inward bends up to ~45mm.⁹
 - If Segment Type == Standard: P_{safe} is rectangular.
 - If Segment Type == Relieved: P_{safe} has a concave cutout.
2. Generate Trajectory Volume V_{flange} of the Return Flange tip.
 - Radius r = Height of Flange.
 - Arc = Angle of bend (e.g., 90 degrees).
3. Intersection Test:
 - If V_{flange} intersects P_{safe} :
COLLISION DETECTED.
 - Else:
SAFE.

5.3 C-Channel Constraints and CLA

If CheckInternalCollision returns TRUE, the standard tool cannot be used. The logic then checks for **CLA (Auxiliary Blade)** availability.³

- **CLA Logic:** The CLA is a modular blade that engages/disengages.
- If the collision is only at the ends of the bend (where the return flanges are), we might use a shorter segmented blade (CLA) to bend the central portion without hitting the side flanges?
 - No, CLA is for the *bending blade*, not the blankholder.
 - The collision usually happens with the *Blankholder*.
 - **Solution:** Use ABA segments with **Side Grooves** (negative reliefs).
 - The CAPP system must check the MaxSideGrooveDepth parameter (typically 30-45mm).¹⁶
 - **Hard Constraint:** If $Length(ReturnFlange) > MaxSideGrooveDepth$, the part is **Unmanufacturable** in that sequence.

6. Precedence Graph Construction

Phase 2 culminates in the generation of the Precedence Graph, a Directed Acyclic Graph (DAG) $G_{prec} = (N, A)$.

6.1 Node Definition: The Operation

Nodes in G_{prec} are not single bends, but **Operations** (Op).

- An Operation is a set of collinear bends performed simultaneously.
- $Op_i = \{e_j, e_k, \dots\}$ where $Line(e_j)$ is collinear with $Line(e_k)$.
- Grouping logic: If edges are collinear and the ABA tool can span them (bridging the gap), they are grouped into one node.

6.2 Edge Generation: Defining Dependencies

We systematically identify dependencies $Op_i \rightarrow Op_j$ (Op_i must precede Op_j).

6.2.1 Type 1: Geometric Covering (Box Closing)

This is the most common constraint.

- If Op_j creates a flange that blocks the tool for Op_i .
- *Test*: Run `IsToolingValid(Op_i)` in the state *after* Op_j . If invalid, add edge $Op_i \rightarrow Op_j$.
- *Result*: This usually enforces "Short sides first" or "Sides with Return Flanges last."

6.2.2 Type 2: Kinematic Collision (Internal Flange)

- If Op_j creates a feature (e.g., a return flange) that would collide with the tool during the execution of Op_i .
- *Test*: Run `CheckInternalCollision(Op_i)` in the state *after* Op_j . If collision, add edge $Op_i \rightarrow Op_j$.

6.2.3 Type 3: Manipulator Stability

- The manipulator needs a flat surface.
- If Op_j consumes the last graspable face (e.g., the final close of a profile), it must be done last.
- *Test*: Calculate $Area_{flat}$ after Op_j . If $Area_{flat} < MinGraspArea$, then Op_j is a "Terminator" operation. All other operations point to it.

6.3 Reverse Unfolding Strategy

To construct the graph efficiently, we employ a **Reverse Unfolding** algorithm.¹⁷ Instead of simulating forward from the flat sheet (branching factor is high), we simulate backward from the finished part (disassembly).

Algorithm 2: Reverse Precedence Construction

1. Initialize CompletedGraph G.
2. Queue Q = {FinishedPartState}.
3. While Q is not empty:
State S = Q.pop()
Identify "Unfoldable" Bends in S:
 - Bends that are accessible.
 - Bends where tooling is valid (IsToolingValid).
 - Bends where extraction is possible (SweptVolume check).For each Unfoldable Bend b:
 - Create Precedence: For all other bends b' in S, b' → b (Forward logic: b is done LAST).
 - Generate PreviousState S_prev = Unfold(S, b).
 - Q.push(S_prev).This recursive decomposition naturally builds the valid DAG.

6.4 Cycle Breaking

If the graph contains cycles ($A \rightarrow B$ and $B \rightarrow A$), the part is theoretically impossible.

- **Resolution:**
 - Check for **Special Tooling** (P-Tool, CLA).
 - Check for **Simultaneous Bending**: Can A and B be done in the same operation?
 - If unresolved, flag as "Unmanufacturable - Design Change Required."

7. Operational Optimization and Sequencing (Preview of Phase 3)

While Phase 2 focuses on *feasibility* (the DAG), the constraints implicitly define the optimization landscape.

7.1 Minimizing Rotations

The P4 manipulator rotation is a slow operation relative to bending. The Precedence Graph often allows "chains" of bends on the same side.

- **Optimization:** In the DAG, edges between operations on the same side have weight 0. Edges between different sides have weight $Cost_{rotation}$.

7.2 Minimizing Tool Changes

Although ABA is "masked," large changes in width might exceed the masked window.

- **Optimization:** Prefer sequences that maintain a constant or monotonically decreasing

tool width.

7.3 Data Presentation: Constraint Summary

Constraint Type	Physical Source	Algorithmic Check	Severity
Tool Width	ABA Granularity	Subset Sum Problem	Hard
Box Closing	Flange Interference	$\text{Gap} > W_{\text{tool}} + 2 \cdot \text{Clearance}$	Hard
Internal Collision	Return Flange vs. Holder	Swept Volume Intersection	Hard
Grasping	Manipulator Physics	$\text{FlatArea} > \text{MinThreshold}$	Hard
Throat Depth	Machine Frame	$\text{PartDiagonal} < \text{ThroatMax}$	Hard
Tool Extraction	Cavity Trap	$\text{Opening} > W_{\text{tool}}$	Hard

8. Conclusion

This report has detailed the algorithmic architecture for **Phase 2: Constraint Identification & Precedence Graph Construction** for the Salvagnini P4. We have established that the P4's versatility—enabled by ABA and interpolated blades—comes at the cost of significantly increased process planning complexity compared to traditional press brakes.

The **Face-Adjacency Graph (FAG)** provides the robust topological foundation required to recognize features like boxes and C-channels. By overlaying this with the **"IsToolingValid"** predicate (solving the Subset Sum problem for ABA) and the **"Swept Volume"** collision detector (handling internal flanges), we can deterministically validate manufacturing feasibility.

The resulting **Precedence Graph (DAG)** is not merely a sequence of steps; it is a mathematical encoding of the physics of the machine. It ensures that no sequence generated in Phase 3 will ever result in a collision or a "trapped tool" scenario. This moves the system

from "Computer-Aided" (where the human checks validity) to "Autonomous" (where the system guarantees validity), fulfilling the requirements of modern Industry 4.0 "Batch One" production environments.

Future Work (Phase 3)

The next phase will ingest this DAG to perform:

1. **Path Planning:** Generating the exact G-code for the manipulator.
2. **Fine-Tuning:** Applying MAC 2.0 corrections for material variability.²
3. **Throughput Optimization:** Selecting the specific topological sort of the DAG that

minimizes cycle time ($Cost = \sum w_{rot} + \sum w_{setup}$).

This framework positions the CAPP system to fully leverage the Salvagnini P4's capabilities, ensuring high utilization and zero-error production runs.

Works cited

1. Salvagnini panel bending: P4-2520 bends 4 different parts - YouTube, accessed February 2, 2026, <https://www.youtube.com/watch?v=gOMHcSt2yQQ>
2. Automatic Sheet Metal Bending Machine - Salvagnini P4, accessed February 2, 2026, <https://www.salvagninigroup.com/en-US/products/panel-benders/P4>
3. p4-benders.pdf - Empire Machinery, accessed February 2, 2026, <https://www.empire-machinery.com/wp-content/uploads/2021/06/p4-benders.pdf>
4. Salvagnini P2 User Manual | PDF - Scribd, accessed February 2, 2026, <https://www.scribd.com/document/761106673/Salvagnini-P2-User-Manual>
5. Efficient Algorithms for Two-Phase Collision Detection - Mitsubishi Electric Research Laboratories, accessed February 2, 2026, <https://www.merl.com/publications/docs/TR97-23.pdf>
6. Swept Volume Computation with Enhanced Geometric Detail Preservation - arXiv, accessed February 2, 2026, <https://arxiv.org/html/2509.09325v1>
7. Panel Bender -P4 - MachineryHost, accessed February 2, 2026, <https://f.machineryhost.com/71b2fab0316bfa7fa24ce0a0e5460bad/Brochure%20-%20Panel%20Bender%20-P4.pdf>
8. The automatic hybrid adaptive Panel Bender. - Sanson Machinery Group, accessed February 2, 2026, <http://www.sansonmachinery.com/wp-content/uploads/2013/12/P4Xe-UK-3962020312.pdf>
9. Universal Sheet Metal Panel Bending Machine - Salvagnini, accessed February 2, 2026, <https://www.salvagninigroup.com/en-US/products/panel-benders>
10. Automated Process Planning for Sheet Metal Bending Operations - Carnegie Mellon University's Robotics Institute, accessed February 2, 2026, https://www.ri.cmu.edu/pub_files/pub1/kim_kyoung_k_1998_1/kim_kyoung_k_1998_1.pdf

11. Panel Bending Tools - JEELIX, accessed February 2, 2026,
<https://www.jeelix.com/panel-bending-tools/>
12. Variants Of Bin Packing Problem Information Technology Essay | UKEssays.com, accessed February 2, 2026,
<https://www.ukessays.com/essays/information-technology/variants-of-bin-packing-problem-information-technology-essay.php>
13. Automated Process Planning of Sheet Metal Bending with Handling Robot | Request PDF, accessed February 2, 2026,
https://www.researchgate.net/publication/317896144_Automated_Process_Planning_of_Sheet_Metal_Bending_with_Handling_Robot
14. NP-Complete Reduction for Subset Sum - algorithm - Stack Overflow, accessed February 2, 2026,
<https://stackoverflow.com/questions/27352478/np-complete-reduction-for-subset-sum>
15. Collision Resolutions in Cloth Simulation - University of Pennsylvania, accessed February 2, 2026,
<https://repository.upenn.edu/bitstreams/39fb3423-25ed-41a4-9c1c-1e7715afa2cc/download>
16. No. 136/04, accessed February 2, 2026,
[https://www.centromacchine.net/images/pdf/29547%20-%20NSE%20P4%20Bendingcenter%20\(1\).pdf](https://www.centromacchine.net/images/pdf/29547%20-%20NSE%20P4%20Bendingcenter%20(1).pdf)
17. Digital Strip Layout Design of Multi-Station Progressive Die - ResearchGate, accessed February 2, 2026,
https://www.researchgate.net/publication/286790350_Digital_Strip_Layout_Design_of_Multi-Station_Progressive_Die
18. Our paper in Computer-Aided Design and Applications | by Analysis Situs - Medium, accessed February 2, 2026,
<https://analysis-situs.medium.com/our-paper-in-computer-aided-design-and-applications-19b7b6292e6a>
19. Salvagnini panel bending: P4-2116 kit production with heavy gage parts - YouTube, accessed February 2, 2026,
<https://www.youtube.com/watch?v=oOxaZ5l5Zsg>