Architectural Deconstruction of Automated Bend Sequencing Logic for Salvagnini P4 Panel Benders

1. Introduction: The Deterministic Complexity of Panel Bending

1.1 The Theoretical Landscape of Computer-Aided Process Planning (CAPP)

The automation of process planning for sheet metal fabrication represents one of the most enduring challenges in computational geometry and manufacturing systems engineering. While subtractive manufacturing methods, such as CNC milling or turning, deal primarily with the removal of material from a rigid block—a process that generally preserves the overall stability and fixation of the workpiece—sheet metal bending is fundamentally transformative. It is a process of global geometric alteration where every operation changes the center of gravity, the moments of inertia, and the collision profile of the part relative to the machine tool. The computational complexity of determining a valid bending sequence for a part with N bends is well-documented in academic literature as being NP-Hard, with the search space growing factorially ($N!$).[1]

For a generalized press brake, the problem is complicated by the human factor; the operator can manipulate the part in six degrees of freedom (6-DOF) with high dexterity but low repeatability. However, the specific domain of this analysis—the **Salvagnini P4 Panel Bender**—introduces a different set of constraints. The P4 is a semi-autonomous manufacturing cell characterized by a horizontal worktable, an automatic manipulator (the "manipulator"), and a universal bending tool set that includes an Automatic Blankholder Adjustment (ABA) system. This topology shifts the algorithmic challenge from one of "path planning with high freedom" (press brake) to "constrained optimization within a deterministic volume" (panel bender).

The objective of this report is to reverse-engineer the "Bend Sequencing Algorithm" utilized by high-end Computer-Aided Manufacturing (CAM) software such as **STREAMBEND**. By dissecting the inputs (B-Rep geometry), the constraints (machine kinematics), and the search strategies (A* heuristics), we derive a comprehensive theoretical framework for the reasoning engine that drives these machines. This analysis does not merely describe what the machine does, but rather reconstructs the computational logic—the "brain"—that enables the hardware to function autonomously. We will explore the nuances of reverse unfolding, collision detection via swept volumes, and the combinatorial optimization of tooling segments, synthesizing data from patents, technical manuals, and academic research to provide a definitive architectural blueprint.

1.2 The Machine Topology: Defining the Physical Constraints

To derive the algorithm, one must first rigorously define the physical plant it controls. The Salvagnini P4 differs fundamentally from the traditional press brake topology, and these differences dictate the structure of the sequencing logic.[2]

The Manipulator Constraint: Unlike a press brake where a backgauge positions the part and a human holds it, the P4 uses a Cartesian manipulator with a pincer/gripper mechanism. This manipulator is responsible for X/Y positioning and C-axis rotation. The existence of the manipulator introduces the "Grasping Problem" into the sequencing algorithm. The software must identify valid "grasp regions" on the sheet that allow the part to be fed into the bending line without the gripper colliding with the bending tools or the machine frame. This creates a coupled dependency: the bend sequence determines the changing shape of the part, which determines the available grasp regions, which in turn constrains the valid bend sequence. This circular dependency is the primary knot the algorithm must untie.[4]

The Universal Bending Tool: The P4 employs a pair of interpolating blades (upper and lower) that articulate to perform bends. This allows for both positive (up) and negative (down) bends without flipping the sheet over—a critical advantage over press brakes. However, it imposes a "Swept Volume" constraint. The algorithm must guarantee that the trajectory of the blade during the bend does not intersect with any previously formed flanges. This is distinct from static collision detection; it requires dynamic simulation of the forming process.[3]

The Automatic Blankholder (ABA): The ABA system is perhaps the most algorithmically demanding component. It consists of a segmented upper tool that adjusts its length to match the bend line. This introduces a discrete combinatorial optimization problem—specifically, a variation of the **Subset Sum Problem** or **Knapsack Problem**. The sequencing engine must not only find a geometric path but also solve for the optimal composition of tool segments (e.g., 5mm, 10mm, 20mm blocks) to achieve the required length L while avoiding interference with side flanges.[7]

1.3 The Scope of Algorithmic Reconstruction

This report structures the reverse-engineered solution into three primary computational solvers that operate in concert within the CAM software:

1. **The Geometric Solver:** Responsible for parsing the boundary representation (B-Rep), recognizing features, and managing the topological "unfolding" of the part.
2. **The Search Engine:** The core decision-maker that navigates the state space of possible bend sequences using heuristic search algorithms (primarily A* variants) to find the

minimum-cost path.

3. **The Constraint Engine:** A physics-based validator that enforces "Forbidden Rules" regarding collisions, tool accessibility, and manipulator masking.

By analyzing these components in depth, we reveal how software like STREAMBEND achieves "Batch One" efficiency—generating safe, optimal code for unique parts without human intervention.[9]

---

2. Geometric Representation and Topological Parsing

2.1 The B-Rep Graph Structure

The foundation of any bending algorithm is the rigorous mathematical representation of the sheet metal part. While the input file is typically a 3D CAD model (STEP, IGES), the internal representation used by the sequencing algorithm is a topological graph, specifically a **Face-Adjacency Graph** or **B-Rep Graph**.[10]

In this graph $G = (N, E)$:

- **Nodes (N):** Represent the planar faces of the sheet metal part. In the initial state (finished part), these are the distinct surfaces separated by bends. In the goal state (flat pattern), these nodes are coplanar.
- **Edges (E):** Represent the bend lines connecting these faces.

However, for a process planning system, this graph must be enriched with semantic manufacturing data. Each edge e_{ij} connecting Face i and Face j carries a payload of attributes:

- **Bend Angle (α):** The target deformation angle (e.g., $90^\circ, 135^\circ$).
- **Bend Radius (R):** The internal radius, which determines the bend deduction and material elongation.
- **Bend Direction (D):** A binary or signed vector indicating if the bend is "Positive" (Up) or "Negative" (Down) relative to the current reference face.
- **Bend Length (L):** The effective length of the flange, which directly inputs into the ABA tooling logic.

This graph representation abstracts the complex 3D geometry into a discrete set of relationships, allowing the algorithm to treat the "unfolding" process as a graph traversal problem. When a bend is "unfolded" in the simulation, the corresponding edge in the graph changes its state from "Active" to "Flat," and the transformation matrices of the connected nodes are updated to become coplanar.[10]

2.2 Feature Recognition and "Forbidden Zones"

Before the search for a sequence can begin, the Geometric Solver must scan the B-Rep for features that impose hard constraints on the process. This is the **Feature Recognition** phase.

Internal Contours and Cutouts: The algorithm identifies loops of edges that lie entirely within a face (i.e., holes). These are critical for the **Manipulator Masking** logic. A suction cup cannot grip a hole. Therefore, the geometric areas defined by these internal contours are marked as **Negative Space** in the grasp analysis. If a face is heavily perforated (e.g., a ventilation grille), the algorithm calculates the "effective surface area" density. If the density drops below a threshold (e.g., 60%), the face is flagged as "Ungrippable," forcing the search engine to find a sequence where the manipulator holds a different face.[13]

Formed Features (Louvers, Embossments, Extrusions):

Standard bending logic assumes faces are flat. However, parts often contain formed features like louvers or bridge lances. The Feature Recognition module identifies these by their topological signature (small, non-planar deformations within a face). These features are mapped as **Height Obstacles**.

- **Constraint:** The blankholder (ABA) clamps down on the sheet with significant tonnage. If the ABA clamps directly onto a louver, the feature will be flattened (destroyed).
- **Logic:** The bounding box of any formed feature is added to a "Forbidden Clamping Zone" layer. The Constraint Solver checks the intersection of the active ABA tool footprint with these zones. If an intersection is detected, the state is invalid.

Corner Reliefs and Topology Types:

The junction where two or three bends meet is a critical topological structure. The algorithm classifies corners into types:

- **Type A (Open Corner):** No overlap; bends can be performed in any order.
- **Type B (Closed/Welded Corner):** Flanges overlap or butt against each other. This implies a strict precedence constraint. For example, if Flange X has a tab that tucks inside Flange Y, Flange X *must* be bent before Flange Y (in forward planning) or unfolded *after* Flange Y (in reverse planning).
- **Type C (Relief):** A circular or rectangular cutout exists to relieve stress. This generally relaxes sequencing constraints.

2.3 The "Reverse Unfolding" Paradigm

The overwhelming consensus in the research material [10] is that effective sequencing algorithms utilize a **Reverse Unfolding** (or "De-bending") strategy rather than forward planning.

Forward Planning (Flat $\rightarrow$ Folded):

Starting with a flat sheet, the algorithm must choose the first bend. The "start state" is simple (a rectangle), but the "goal state" is complex (the folded part). The branching factor is massive because any bend could theoretically be the first. Furthermore, simulating the intermediate states is difficult because the part becomes increasingly non-convex and collision-prone as it folds.

Reverse Planning (Folded $\rightarrow$ Flat):

The algorithm starts with the finished 3D part—the "Goal State" of manufacturing, but the "Start State" of the algorithm. The objective is to unfold it flat.

- **Advantage 1 (Reduced Branching):** In a finished 3D part, many bends are physically "buried" or blocked by other flanges. We cannot unfold a flange that is tucked inside another. Therefore, the set of *candidate bends* for the first unfolding step is small—only the "outermost" flanges are available. This naturally prunes the search tree.
- **Advantage 2 (Collision Simplification):** Checking if a flange can be unfolded *out* of the machine is often computationally simpler than checking if it can be folded *in*.
- **Advantage 3 (Known Target):** The target state is always a convex, flat polygon (the blank). Convergence is easier to measure.

Algorithm 1: High-Level Reverse Unfolding Strategy

1. **Initialize:** Set S_0 = Finished 3D Part Model.
2. **Generate Candidates:** Identify all bends b_i in S_0 that are topologically "exposed" (i.e., not covered by other flanges).
3. **Feasibility Check:** For each candidate b_i:
    - **Grip Check:** Can the manipulator hold the *remaining* part if b_i is unfolded?
    - **Tool Check:** Can the ABA tool fit inside b_i?
    - **Collision Check:** Does the trajectory of unfolding b_i intersect the machine frame?
4. **Branch:** If b_i is feasible, create a new state S_{next} where b_i is flat.
5. **Cost Assignment:** Calculate the cost of the transition (Rotation, Tool Change).
6. **Recurse:** Repeat until S_{final} is the flat blank.

3. The Search Engine: A* and Heuristic Optimization

The geometric solver generates a graph of possible states (a state-space). The **Search Engine** is responsible for navigating this graph to find the optimal sequence. The primary algorithm employed is *A (A-Star)**, favored for its ability to find the lowest-cost path while efficiently pruning the search space using heuristics.[10]

3.1 The A* Cost Function Formulation

In the context of the Salvagnini P4, "Cost" is a proxy for "Cycle Time" and "Process Risk." The standard A* function is $f(n) = g(n) + h(n)$.

3.1.1 The Accumulated Cost ($g(n)$)

The term $g(n)$ represents the actual cost to reach the current state of partial unfolding. It is the sum of time penalties for all operations performed so far.

$$g(n) = \sum_{i=1}^{k} (T_{bend} + T_{trans} + T_{rot} + T_{tool} + T_{repo})$$

- T_{bend} **(Bending Time):** The time for the blades to perform the stroke. This is relatively constant per bend and has a low weight in optimization.
- T_{trans} **(Translation Time):** The time for the manipulator to move the sheet in X/Y from the previous bend coordinate to the current one. This is distance-dependent but generally fast (high-speed axes).
- T_{rot} **(Rotation Time): High Cost Factor.** The manipulator must rotate the sheet (C-axis) to align a new side. This is physically slow (large moment of inertia) and introduces positioning errors. Minimizing the total number of rotations is a primary objective.[10]
- T_{tool} **(Tool Change Time): Variable Cost Factor.** If the ABA system must resize, this takes time. However, P4 machines allow **Masked Time** operations—changing the tool *during* a rotation or translation.
  - *Logic:* If $Time(ABA\_Resize) < Time(Rotation)$, then $T_{tool} = 0$ (Masked).
  - *Logic:* If $Time(ABA\_Resize) > Time(Rotation)$, then $T_{tool} = Time(ABA\_Resize) - Time(Rotation)$ (Unmasked penalty).

- T_{repo} **(Repositioning Time): Extreme Cost Factor.** If the manipulator must release the part, move to a new grip position, and re-clamp (a "Repo"), this is the most expensive operation in the cycle (seconds of dead time). The algorithm treats this as a last resort.[17]

3.1.2 The Heuristic Function ($h(n)$)

The heuristic $h(n)$ estimates the *remaining* cost to finish unfolding the part. For A* to be admissible (guarantee optimality), $h(n)$ must not overestimate the true cost.
A robust heuristic for panel bending includes:

- **Remaining Side Count (H_{rot}):** If the remaining folded flanges are located on 3 different sides of the sheet, the heuristic estimates at least 2 rotations are required.
    - $H_{rot} = (Count_{sides} - 1) \times Cost_{Rotation}$.
- **Tooling Variance (H_{tool}):** The heuristic looks at the lengths of the remaining bends. If they are all identical, $H_{tool} = 0$. If they vary significantly, it estimates the number of ABA adjustments needed.
    - $H_{tool} = Count_{distinct\_lengths} \times Cost_{Adjustment}$.
- **Embedding Complexity (H_{embed}):** If the topological graph shows deep nesting (flanges inside flanges), the heuristic adds a penalty to account for the restricted manipulator access, hinting at potential future "Repo" operations.

3.2 Heuristic Rules and Tie-Breakers

While A* provides the mathematical framework, practical implementation relies on "Heuristic Rules" to guide the search and break ties between paths with similar costs. Research [15] identifies specific rules used in this domain:

1. The "Outside-In" Rule:

In reverse planning, priority is given to unfolding the outermost flanges first. These are the flanges that enclose the bounding box of the part. Processing them first in reverse (last in forward) ensures that the final geometry is formed when the sheet is smallest and stiffest, but the algorithm works backwards from the finished state where "outer" implies "most accessible."

2. The Collinearity Rule:

If multiple bends lie on the same axis (e.g., interrupted bend lines on the same side), they should be processed sequentially.

- *Reasoning:* No rotation is required between them. The tool setup (ABA) is likely identical or requires minimal adjustment. The manipulator does not need to change the orientation, only the Y-position.
- *Implementation:* The algorithm groups collinear bends into "Macro-Nodes" in the search graph, treating them effectively as a single operation step.

3. The "Longest Side" Rule (with ABA modification):

Standard press brake logic suggests bending the longest side first for stability. However, P4 logic is more nuanced due to the ABA.

- *Insight:* In P4 bending, shorter sides are often bent *before* longer sides (in forward sequence). Why? Because the ABA segments must fit *between* the previously bent flanges to bend the final long side. This is the **"Box Closing Constraint"**.[19]
- *Reverse Logic:* Therefore, in reverse unfolding, the algorithm prefers to unfold the *longest* side first (because the tool fits easily), leaving the shorter sides (which might be "locking" the tool) for later.

4. The Symmetry Rule:

For parts with axial symmetry, the algorithm favors sequences that mirror operations (e.g., Bend Left, Bend Right). While less critical for cold bending than thermal processes, this maintains geometric balance and often simplifies manipulator path planning by keeping the Center of Gravity (CoG) predictable.

---

4. Deep Dive: Collision and Accessibility Constraints

The **Constraint Engine** is the most computationally intensive module of the STREAMBEND software. It acts as the "Gatekeeper," validating every node in the A* search tree. If a state violates a constraint, it is assigned an infinite cost, effectively pruning that branch.

4.1 Manipulator Masking: The "Dead Zone" Logic

The P4 manipulator is a rigid body that occupies space on the sheet. It cannot hold the part at a location that will enter the bending line. This creates a **"Dead Zone"** or **"Forbidden Grasp Region"**.[4]

4.1.1 Polygon Intersection Algorithms

To validate a bend line, the algorithm performs a 2D geometric check:

1. **Grip Projection:** The footprint of the active grippers (suction cups or clamps) is projected onto the flat pattern of the sheet. Let this polygon be P_{grip}.

2. **Safe Zone Definition:** For a specific bend b_i, the machine geometry defines a "Safe Zone" Z_{safe}. This is the area of the sheet that remains outside the tool throat and blade operational area during the bend.
3. **Constraint Test:** The condition for validity is $P_{grip} \subset Z_{safe}$.
   - This is typically solved using **Sutherland-Hodgman** or **Weiler-Atherton** polygon clipping algorithms.
   - If P_{grip} intersects the boundary of Z_{safe}, a collision is imminent (the machine would crush its own gripper).

4.1.2 The "Repo" (Regripping) Logic

If the current grip position is invalid for the next required bend (i.e., the optimal bend is inside the Dead Zone), the algorithm triggers a **Repositioning Routine**.[17]

- **The Problem:** Find a new position P'_{grip} on the sheet such that the manipulator can execute the next sequence of bends.
- **The Algorithm:**
  1. Identify the "Free Space" on the sheet (Total Area - Holes - Features - Current Dead Zones).
  2. Overlay the "Future Dead Zones" of the *next* k bends in the queue.
  3. Find the intersection of Free Space and the Safe Zones of future bends.
  4. Select a candidate P'_{grip} within this intersection that maximizes stability (closest to CoG).
- **Cost:** This "Repo" step is injected into the sequence as a distinct node with a very high time cost (T_{repo}), forcing the A* search to avoid it unless absolutely necessary.

4.1.3 Rotation Constraints

When the manipulator rotates the part (C-axis), the entire sheet sweeps a circular area.

- **Throat Check:** The maximum radius of the part (R_{max}) from the rotation center is calculated. If $R_{max} > Throat_{depth}$, the rotation is physically impossible at the current X/Y coordinates.
- **Injection of Translation:** The logic injects a translation step: Move Part to (X_{safe}, Y_{safe}) $\rightarrow$ Rotate $\rightarrow$ Move back. This ensures the corners of the swinging sheet do not hit the machine columns.[10]

4.2 Internal Collision: Swept Volume Analysis

Collision detection in panel bending is 4-dimensional (3D space + Time). A static interference check is insufficient because the flanges move through space to form the bend.

4.2.1 BEND_UP vs. BEND_DOWN Dynamics

The P4 supports bidirectional bending, which doubles the collision complexity.[1]

- **BEND_UP (+):** The lower blade holds the sheet, and the upper blade moves. The flange swings *upward* towards the Blankholder (ABA).
    - **Conflict:** The inner face of the rising flange may collide with the face of the ABA tool if the flange is too tall or has an inward return (C-bend).
    - **Rule:** $Height_{flange} < Clearance_{ABA}$.
- **BEND_DOWN (-):** The upper blade holds the sheet, and the lower blade moves. The flange swings *downward* towards the table/counterblade.
    - **Conflict:** The outer face of the descending flange may collide with the counterblade structure or the machine table.
    - **Rule:** Negative bends are often more restricted in height due to the table structure below.

4.2.2 Swept Volume Generation

To validate a bend θ, the software generates a **Swept Volume**.[22]

$$V_{swept} = \bigcup_{t=0}^{1} Volume(Part(t))$$

Where $Part(t)$ represents the geometry of the flange deformed by angle $t \times \theta$.

- **Implementation:** Efficient implementation uses **AABB Trees** (Axis-Aligned Bounding Box hierarchies).[1] The flange geometry is encapsulated in a box hierarchy. As it rotates, the boxes are updated. Intersection tests are performed between these dynamic boxes and the static boxes of the Machine (ABA, Blades, Manipulator).
- **Precision:** If a coarse AABB check indicates a collision, the algorithm drills down to the exact mesh geometry (Triangle-Triangle intersection) to confirm. This two-phase approach optimizes processing speed.

4.3 Tool Segment Logic: The ABA Dependence

The Automatic Blankholder Adjustment (ABA) is a defining feature of the Salvagnini P4. It is not a single solid block but a dynamic assembly of segments that can expand or contract. This creates a unique algorithmic challenge: **The Subset Sum Problem** in tooling.[7]

4.3.1 The Subset Sum Formulation

Let the ABA be composed of a set of available segments $S = \{s_1, s_2, ..., s_m\}$ (e.g., 5mm, 10mm, 20mm, 50mm, 100mm, 200mm).

For a target bend length L, the algorithm must find a subset $S' \subseteq S$ such that:

$$\sum_{s \in S'} Length(s) \leq L_{effective}$$

And:

$$L_{effective} - \sum_{s \in S'} Length(s) < Tolerance$$

Why $L_{effective}$?

If the part has side flanges that are already bent, the tool cannot equal the full length of the current bend line. It must fit *inside* the side flanges.

$$L_{effective} = L_{bend} - 2 \times (Thickness + BendRadius + SafetyGap)$$

This is a classic variation of the **Knapsack Problem** or **Change-Making Problem**. The algorithm uses **Dynamic Programming** to find the optimal combination of segments that maximizes width (for clamping stability) without exceeding the rigid limit (collision with side flanges).

4.3.2 Tooling Strategy in Search

The search engine treats the ABA configuration as a state variable.

- **State Persistence:** The algorithm prefers sequences where the ABA configuration remains constant. $Cost_{state\_change}$ is non-zero.
- **Masked Time Logic:** This is the critical efficiency driver. The P4 can adjust the ABA *while* the manipulator is positioning the sheet.
  - The algorithm calculates the time for manipulator travel (t_{manip}) and the time for ABA adjustment (t_{aba}).
  - If $t_{aba} < t_{manip}$, the cost of the tool change is effectively **Zero**.

    - This leads to "Counter-Intuitive" optimal paths: The algorithm might choose a sequence that requires a tool change (which seems slow) over one that doesn't, *if* that tool change can be perfectly masked by a necessary rotation, whereas the "no tool change" path might require a slow, unmasked Repo.

---

5. Heuristic Rules and Forbidden Logic (The "Red Flags")

In addition to calculating costs, the algorithm employs a set of "Forbidden Rules"—hard logic gates that immediately invalidate a sequence. Identifying these is crucial for reverse-engineering the system.

5.1 Forbidden Rule 1: The "Box Closing" Failure

This is the most common failure mode in panel bending.

- **Scenario:** Bending the 4th side of a box.
- **Logic:** The tool must fit inside sides 1 and 3.
- **Constraint:** If $Width_{ABA\_min} > (Width_{Box} - 2 \times Flange_{Side})$, the part cannot be made.
- **Detection:** The algorithm checks the minimum collapsible width of the ABA. If the part geometry is tighter than this limit, the "Box Closing" branch is pruned. The software will return a "Tooling Interference" error.[19]

5.2 Forbidden Rule 2: Counterblade Collision on Down-Bends

- **Scenario:** Performing a negative bend when the part has a deep positive flange on the opposite side.
- **Logic:** As the part tilts down to make the negative bend, the previously bent positive flange (now pointing down relative to the table) swings towards the machine base.
- **Constraint:** If the swing radius of the existing flange intersects the machine table volume, the bend is forbidden.
- **Remediation:** The search engine backtracks and attempts to perform the negative bend *earlier* in the sequence (before the conflicting positive flange exists).

5.3 Forbidden Rule 3: Insufficient Grip Area

- **Scenario:** The part is heavily perforated or has very narrow borders.
- **Logic:** The manipulator requires a minimum surface area for vacuum suction or clamping friction.
- **Constraint:** $Area_{contact} < Threshold$ (e.g., 60% of pad area).
- **Error Code:** "Part Slippage Risk" or "Insufficient Vacuum".[13]

- **Algorithmic Consequence:** The algorithm marks nodes requiring this grip as invalid. If no valid grip exists for any orientation, the part is flagged as "Unmanufacturable on P4."

5.4 Forbidden Rule 4: K-Factor/Elongation Limits

- **Scenario:** Bending thick material with a tight radius.
- **Logic:** The software calculates the unfolded length based on the material's K-factor.
- **Constraint:** If the required bend radius is smaller than the minimum bend radius for the material type/thickness (causing cracking), the operation is forbidden. While primarily a design constraint, the sequencing engine checks this to ensure the *force* required doesn't exceed the P4's tonnage rating.[24]

---

6. Architecture and Implementation

How is this logic structured in software? Based on the analysis of **STREAMBEND** workflow [13], the architecture follows a hierarchical solver pattern.

6.1 The Pre-Computation Layer

Before any search begins, the software performs static analysis:

1. **Parsing:** Convert STEP to B-Rep Graph.
2. **Unfolding:** Calculate the 2D flat pattern and all intermediate "De-bent" states.
3. **Grip Mapping:** Pre-calculate valid grip polygons for the flat pattern and assign them to the graph nodes.
4. **Tooling Lookup:** For every bend edge e, compute the required ABA segment configuration S'.

6.2 The Solver Core (The "Black Box")

This is the A* engine.

- **OPEN List:** Priority queue of state nodes, ordered by $f(n)$.
- **CLOSED List:** Hash map of visited states to prevent cycles.
- **Node Expansion:** From state S_i, generate children S_{i+1} by unfolding valid edges.
  - *Apply Forbidden Rules:* Prune children that violate constraints (Collision, Grip).
  - *Apply Heuristics:* Calculate $h(n)$ for remaining children.
  - *Calculate Cost:* Compute $g(n)$ based on machine kinematics (Time).

6.3 The Post-Processor

Once the "Reverse" path is found (Flat $\leftarrow$ Folded), the software:

1. **Inverts** the sequence to Forward (Flat $\rightarrow$ Folded).
2. **Simulates** the forward sequence with high-fidelity physics (Collision check verification).
3. **Optimizes** the manipulator trajectory. It smoothes out the "Repo" moves and rotations to create fluid motion profiles.
4. **Generates NC Code:** The final output is the machine language file.

---

7. Conclusion

The "Bend Sequencing Algorithm" for a Salvagnini P4 is not a single algorithm but a composite of three distinct solvers: a **Geometric Solver** for topology, a **Constraint Solver** for physics, and a **Heuristic Search Engine** for optimization.

The analysis confirms that the industry standard approach is **Reverse Unfolding A***. The complexity of the problem is managed not by brute force, but by the intelligent application of **Forbidden Rules** (derived from the machine's specific constraints like the ABA width and Manipulator Dead Zones) and **Masked Time Heuristics** (which prioritize operations that can be hidden behind others).

For any entity attempting to replicate this logic, the critical path lies in:

1. Implementing a robust **Polygon Intersection Engine** for the Manipulator Masking.
2. Developing a **Dynamic Programming Solver** for the ABA Subset Sum problem.
3. Tuning the **A* Cost Weights** to accurately reflect the machine's physical axis speeds, ensuring that the software chooses not just the shortest path, but the fastest one.

This architecture enables the "Panel Bending 4.0" capability: the ability to take a batch of N unique parts and produce them sequentially with zero setup time, a feat of algorithmic elegance that matches the mechanical ingenuity of the machine itself.

Works cited

1. Collision detection during planning for sheet metal bending by bounding volume hierarchy approaches | Semantic Scholar, accessed February 1, 2026, https://www.semanticscholar.org/paper/3910d3639613f1d89ce9ebe6cee213c2aae4969b
2. Automatic bending machine, P4 panel bender - Salvagnini, accessed February 1, 2026, https://www.salvagninigroup.com/en-INT/products/panel-benders/P4
3. p4-benders.pdf - Empire Machinery, accessed February 1, 2026, https://www.empire-machinery.com/wp-content/uploads/2021/06/p4-benders.pdf
4. Panel Bender vs CNC Press Brake - ADH Machine Tool, accessed February 1,

2026, https://www.adhmt.com/what-is-a-panel-bender/
5. The Fabricator - October 2011 - ABB, accessed February 1, 2026, https://search.abb.com/library/Download.aspx?DocumentID=9AKK105408A4132&LanguageCode=en&DocumentPartId=&Action=Launch
6. Universal Sheet Metal Panel Bending Machine - Salvagnini, accessed February 1, 2026, https://www.salvagninigroup.com/en-US/products/panel-benders
7. Øker produktiviteten med Salvagnini - Maskinregisteret, accessed February 1, 2026, https://www.maskinregisteret.no/tungt?p=/maskinregisteret/oker-produktiviteten-med-salvagnini-6691590
8. The automatic hybrid adaptive Panel Bender. - Sanson Machinery Group, accessed February 1, 2026, http://www.sansonmachinery.com/wp-content/uploads/2013/12/P4Xe-UK-3962020312.pdf
9. Sustainable, compact bending solution - SHM Consulting, accessed February 1, 2026, https://shmconsulting.hu/wp-content/uploads/2022/11/P1.pdf
10. Automatic determination of bending sequences for sheet metal parts with parallel bends, accessed February 1, 2026, https://www.researchgate.net/publication/233222153_Automatic_determination_of_bending_sequences_for_sheet_metal_parts_with_parallel_bends
11. A CAD-Driven and Cloud-Based Autonomous Process Planning Framework for Reconfigurable Bending Press Machines - Taiwan Association of Engineering and Technology Innovation, accessed February 1, 2026, https://ojs.imeti.org/index.php/PETI/article/view/14044/1670
12. Bending Sequence Automation for Sheet Metal | PDF | Computers - Scribd, accessed February 1, 2026, https://www.scribd.com/document/668802543/Bend-order-in-sheetmetal
13. Streambend en | PDF | Rotation | Art - Scribd, accessed February 1, 2026, https://www.scribd.com/document/761106869/Streambend-En
14. On sheet metal unfolding. Part 6: bend sequences | by Analysis Situs - Medium, accessed February 1, 2026, https://medium.com/@analysis-situs/on-sheet-metal-unfolding-part-6-bend-sequences-11d06840c5e8
15. 255 Title: Bending Simulation Framework for Rapid Feasibility Checks of Sheet Metal Parts Authors - CAD Conference, accessed February 1, 2026, https://www.cad-conference.net/files/CAD25/CAD25_255-260.pdf
16. A Methodology for the Sequencing of Sheet Metal Bending Operations - ResearchGate, accessed February 1, 2026, https://www.researchgate.net/publication/314446965_A_Methodology_for_the_Sequencing_of_Sheet_Metal_Bending_Operations
17. Intelligent system for generating and executing a sheet metal bending plan - Google Patents, accessed February 1, 2026, https://patents.google.com/patent/US20040019402A1/en
18. a novel feature-based bending sequence planning for sheet metal components for rail parts - International Journal of Applied Engineering & Technology,

accessed February 1, 2026,
https://romanpub.com/resources/ijaet20v6-1-2024-168.pdf

19. Salvagnini P2 User Manual | PDF - Scribd, accessed February 1, 2026, https://www.scribd.com/document/761106673/Salvagnini-P2-User-Manual
20. Salvagnini S4 punching & shearing machines - Empire Machinery, accessed February 1, 2026, https://www.empire-machinery.com/wp-content/uploads/2021/06/s4-sl4-punching-centers.pdf
21. ProtoTRAK ® - Engineering Subcontractor, accessed February 1, 2026, https://engineeringsubcontractor.com/images/magazine_pdf_downloads/2022/ES%20NOVDEC%202023.pdf
22. Neural Implicit Swept Volume Models for Fast Collision Detection - arXiv, accessed February 1, 2026, https://arxiv.org/html/2402.15281v3
23. Collision Detection and Part Interaction Modeling to Facilitate Immersive Virtual Assembly Methods, accessed February 1, 2026, https://dr.lib.iastate.edu/bitstreams/70378854-8eed-49d6-83dd-48e9744de172/download
24. TruBend 2100 - TRUMPF, accessed February 1, 2026, https://www.trumpf.com/filestorage/TRUMPF_MX/TruBend2100/pdfs/TUS_TruBend_2100_P61.1_V1_1.pdf
25. #3 Matt Humerick Salvagnini Software - YouTube, accessed February 1, 2026, https://www.youtube.com/watch?v=-gHDB6IRQ8A