# Enhancing Manufacturing with AI-powered Process Design

**Gianmarco Genalti**[1], **Gabriele Corbo**[1], **Tommaso Bianchi**[1], **Marco Missaglia**[2], **Luca Negri**[2], **Andrea Sala**[2], **Luca Magri**[1], **Giacomo Boracchi**[1], **Giovanni Miragliotta**[1], **Nicola Gatti**[1]

[1]Politecnico di Milano

[2]Agrati S.p.A.

{gianmarco.genalti, gabriele.corbo, tommaso.bianchi, luca.magri, giacomo.boracchi, giovanni.miragliotta, nicola.gatti}@polimi.it, {marco.missaglia, luca.negri, andrea.sala}@agrati.com

## Abstract

Manufacturing companies are experiencing a transformative journey, moving from labor-intensive processes to integrating cutting-edge technologies such as digitalization and AI. In this demo paper, we present a novel AI application to enhance manufacturing processes. Remarkably, our work has been developed in collaboration with Agrati S.p.A., a worldwide leading company in the bolts manufacturing sector. In particular, we propose an AI-powered application to address the problem of automatically generating the *production cycle of a bolt*. Currently, this decision-making task is performed by process engineers who spend several days to study, draw, and test multiple alternatives before finding the desired production cycle. We cast this task as a *model-based planning* problem, mapping *bolt technical drawings* and *metal deformations* to, potentially continuous, *states* and *actions*, respectively. Furthermore, we resort to *computer vision* tools and *visual transformers* to design *efficient heuristics* that make the search affordable in concrete applications. Agrati S.p.A.'s process engineers extensively validated our tool, and they are currently using it to support their work. To the best of our knowledge, this is the first example of an AI application dealing with production cycle design in bolt manufacturing.[1]

## 1 Industrial Context

Agrati S.p.A. S.p.A. is one of the world's leading companies in bolts manufacturing, with 12 production sites spread worldwide. Most of their customers ask for customized products. In particular, a customer produces an RFQ by supplying Agrati S.p.A. with a technical drawing of what the customized piece should look like, asking for thousands of identical units. Then, Agrati S.p.A.'s process engineers are asked for defining a novel production pipeline to allow the production of the entire amount of bolts by the time requested by the customer. Every batch of bolts is produced starting from a steel thread cut into cylinders. Every cylinder is shaped into a bolt through sequential steel-forming operations, such as extrusions. Engineers are called to propose both the starting diameter of the thread and which operations (and in what order) have to be performed. There are multiple constraints when deciding on the production pipeline, for example, limits on the ratio between radius and length or a maximum length that a machine can manage for a piece. There are more than 10 different forming operations, each deforming a metal piece differently. A set of parameters characterizes an operation, for example, the height at which a cut should be done or how much a radius has to be reduced. Operations can be applied along the sagittal axis on both senses, thus making the effective decision space at least two times bigger. In Figure 1, we report an example of RFQ provided to Agrati S.p.A., representing a bolt and the desired measures. This project's goal is to obtain both a sequence of operations, their parameters, and the radius and height of the starting thread. If the operations are applied in the supplied order to such a thread, the final component is obtained as the customer desires without violating any constraint. Engineers usually study the optimal sequence of operations by exploiting sequences developed in the past for similar components. However, this task may require several hours, and when the desired component is particularly involved, the engineers' team may fail to find the correct sequence. This may lead to important delays in production or in the loss of commercial opportunities. In Figure 2, we report an example of a full production cycle designed by the company's engineers.

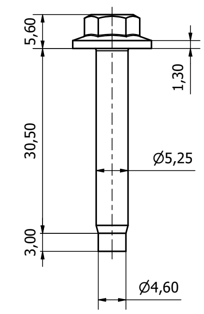


Figure 1: Example of RFQ provided by customer.

## 2 Solution Outline

In Figure 3, we report a screenshot of the demo interface. In this example, we uploaded the RFQ of a component, specified its length, and indicated the presence of a hexagonal head[2], and got the full production cycle (image and textual

[1]See the video presentation here (YouTube video).

[2]It is required to specify if the component is drilled or has a hexagonal head since this is in general, not easy to infer from an

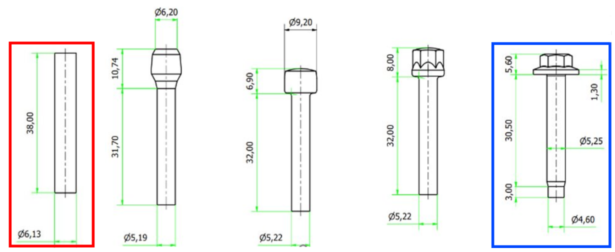

Figure 2: Example of full production cycle, where we highlight the RFQ supplied by the customer and the starting cylinder. The engineers aim to reconstruct the path between the cylinder and the final bolt, only knowing the latter.

description comprising all operations together with their parameters). The component is the same whose cycle has been reported in Figure 2. We can observe how the cycle proposed by our tool (Fig. 3) is the same that human engineers computed to produce a large amount of these components (Fig. 2). For every tested component, the tool returned an output

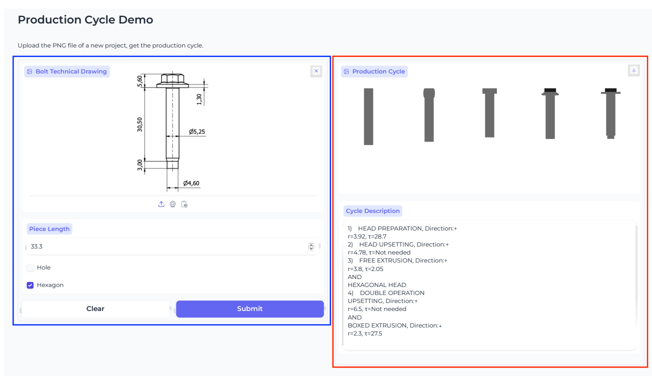


Figure 3: Interface of the tool developed for automatic production cycle computation. On the left, a user can supply the component's RFQ and its length in millimeters. Instead, on the right, the algorithm's output shows a renderization of the proposed cycle and the ordered list of operations together with the required parameters.

in less than 10 seconds, running on a single core of an Intel Xeon Platinum 8358 processor with 512 Gigabytes of RAM.

Additional demonstrations of the tool's capabilities are shown here (YouTube Video).

## 3 A Tool for Automatic Production Cycle Design

While scientific literature is rich in AI applications for process planning, there are no examples of AI solutions dealing with this specific problem (Kumar [2017]). To the best of our knowledge, this is the first attempt at modeling the production cycle of a bolt as a *model-based planning* problem.

In particular, a finished component is the product of a sequence of operations applied on a metal thread. Different operations lead to different outcomes. We then search for the sequence of operations that lead exactly to our component when applied to a certain (and feasible) metal thread. Flipping our perspective, the finished component is our starting state, and the inverse of the metal deformations available are the actions we can make. Thus, there's a sequence of inverted transformations (that, from now on, we will call transformations) that lead to a feasible metal thread, which is our goal state. Operations are deterministic, and the model is, in principle, fully known. Thus, our goal is to simulate the production process and search for a successful sequence of actions. Ultimately, we can represent this planning problem as a search in a (recombinant) tree, and an example is reported in Figure 4. However, to search for the production cycle, we

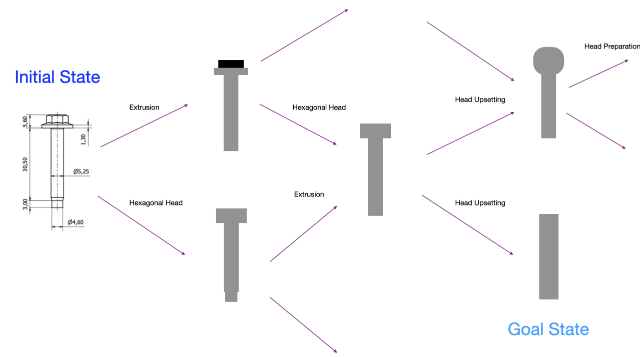


Figure 4: Example of an (inverse) operations tree leading to a metal thread starting from the finished component.

need a complete representation of the model and, thus, a formal model for both states and actions.

First, we introduce a formal model of a metal component: in particular, we require a model of both the final bolt and the intermediate steps between forming operations (including the starting thread). All the products in our scope possess rotational symmetry (possibly discrete), which allows us to model a bolt only using its silhouette. Thus, we encode the image of a metal component using its corners' coordinates, as in Figure 5. Our available metal deformations are all assumed to be

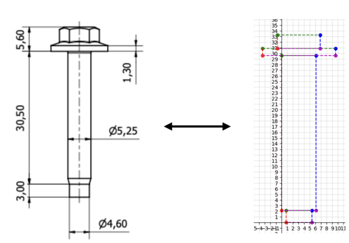

Figure 5: Mapping of a metal component's silhouette to cartesian coordinates.

*isovolumetric*. Moreover, components are assumed to be rotationally symmetric. Under these two assumptions (which most components respect), all transformations can be formally written as compositions of 2D linear transformations,

image, but it's trivial to observe for a human.

particularly trapezium-to-trapezium transformations. In Figure 6, which represents a *free extrusion* (one of the most common forming operations), we can observe this: a thread, whose silhouette is a rectangle, is transformed in two cylinders having the same total volume, and one having a smaller radius. Given a height $\tau$ and a new radius $r$, and thanks to the isovolumetry, the transformation's output is uniquely defined. Moreover, this transformation is easily invertible, allowing us to use the inverse transformation principle for our model. Finally, mechanical constraints can be easily ensured by quantities like $\tau$ and $r$ since they can be expressed as relationships between such quantities.

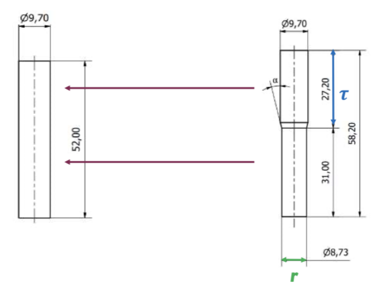


Figure 6: Example of inverse free extrusion.

## 4 Data-driven Heuristics for Computational Feasibility

There are more than 10 different transformations, most of which are parametrized by continuous values. Moreover, some components may require more than 6 operations to be formed. Due to these reasons, the tree size quickly becomes huge, precluding an exhaustive search for the correct production cycle.

Our solution is to provide a data-driven heuristic to guide search and explore fewer nodes. In particular, while performing a *depth-first search*, we want to prioritize actions most likely leading to feasible starting threads.

The available dataset is composed of both successful and failed production cycles (*i.e.,* not leading to a thread in few operations or where the thread is not feasible due to constraints)[3]. In a cycle, we isolate all state-action-state triples, allowing us to map each state-action couple to the subsequent state. However, we encountered two main issues in representing states in a tabular fashion as the coordinates of their corners. First, the dimension may vary drastically, *e.g.,* a metal thread is only characterized by four coordinates while a finished piece may have many more corners; and second, since the available dataset is mainly composed of past production cycle images, it would require a large human effort to individuate all corners' coordinates properly or to sanity check the correctness of any automated tool doing this. To avoid this issue, we decided to split cycle images, extract the images of all the states involved, and embed them in equally-sized lower dimensional arrays. To make this conversion, we use as a pre-trained embedder the Vision Transformer (ViT) model trained using the DINO method, a transformer encoder model pre-trained on a large collection of images in a self-supervised fashion (Caron *et al.* [2021]).

[3]Even if the dataset is not sufficiently large, fully knowing the model dynamics allows to generate a large number of synthetic cycles randomly.

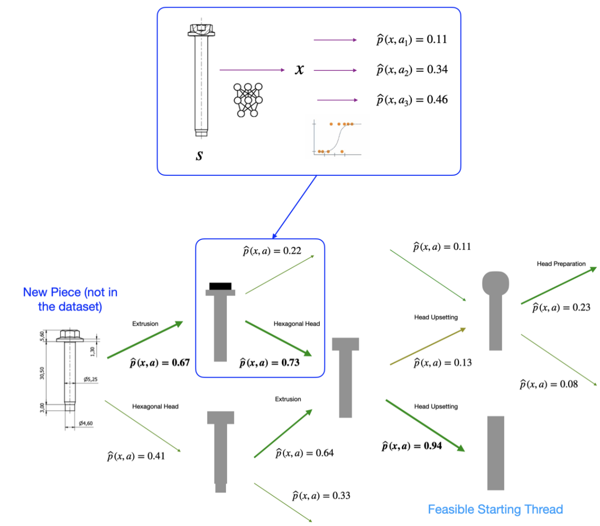


Figure 7: This scheme represents the working of our data-driven heuristic. When exploring the tree, for every encountered state, we embed it and predict the likelihood of success for every action. Then, actions are chosen from the most likely to the least likely.

Now, for every embedded state, we can associate the action that has been performed there and the associated output, *i.e.,* a binary label indicating whether or not the cycle resulted was successful. Any supervised learning algorithm can use such a labeled dataset to predict the probability of a state-action couple resulting in a successful cycle. In particular, we trained a logistic regression (Hosmer Jr *et al.* [2013]). For every new state-action couple, even if not present in the historical dataset, we can now assign a weight indicating the likelihood of it conducting a successful cycle.

Figure 7 reports the functioning scheme of our data-driven heuristic. Actions most likely to reach a feasible thread are chosen before the others, according to the ordering provided by the supervised model prediction. Even if this real-time inference of embedder plus supervised model brings some additional computational burden to the single-node decision-making, in practice, the reduction in the number of visited nodes is so high that this results in dramatic advantages.

## 5 Conclusions and Future Developments

We provided an AI tool to assist engineers in designing the production of custom metal components. To the best of our knowledge, this is the first application of AI in this specific field. An automatic production cycle design allows Agrati S.p.A. to improve in-site operations planning, saving a large amount of money and time. In the future, we plan to extend this approach to different (and possibly harder) manufacturing domains.

## Ethical Statement

There are no ethical issues.

## References

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

SP Leo Kumar. State of the art-intense review on artificial intelligence systems application in process planning and manufacturing. *Engineering Applications of Artificial Intelligence*, 65:294–329, 2017.