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] and **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 tool 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 tool 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, ours is the first AI tool dealing with production cycle design in bolt manufacturing.

1 Industrial Context

Agrati 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 provides an RFQ (Request For Quotation) with a technical drawing of the customized component and the number of needed units (usually of the order of thousands). Then, Agrati S.p.A.'s process engineers are asked to promptly define a novel production pipeline to allow the completion of all the bolts by the time requested by the customer. Every batch of bolts is produced starting from a steel thread cut into cylinders. Furthermore, every cylinder is shaped into a bolt through sequential steel-forming operations, such as extrusions. Process engineers are called to propose both the starting diameter of the thread and the operations (and in what order) 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 component. Remarkably, there are more than 10 different forming operations, each deforming a metal component differently. Each forming operation is characterized by a set of parameters, for example, the height at which a cut needs to be done or how much a radius needs to be reduced. Forming operations can be applied along the sagittal axis on both senses, thus making the effective decision space at least two times bigger. In Fig. 1, we report an example of customer RFQ, specifying the shape of the bolt and its 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 requested by the customer without violating any constraint. Process engineers usually study the optimal sequence of operations by adapting sequences developed in the past for similar components. However, this task may require several hours or even days, and, when the desired component is particularly involved or no similar components have been developed in the past, the process engineers' team may fail to find the correct sequence. This may lead to important delays in production or loss of commercial opportunities. In Fig. 2, we report an example of production cycle designed by the company's engineers.

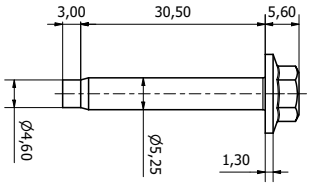

Figure 1: Example of customer RFQ.

2 Solution Outline

In Fig. 3, we report a screenshot of the interface of our tool. On the left, in the blue rectangle, the uploaded RFQ of a component is shown. It specifies its length and the presence of a

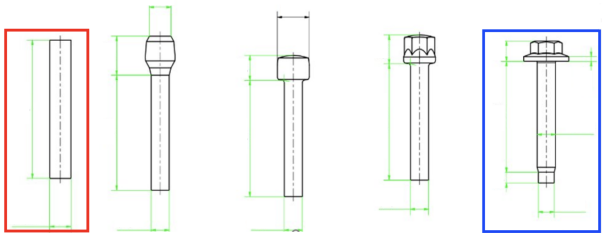

Figure 2: Example of full production cycle, where we highlight the RFQ supplied by the customer and the starting cylinder. The engineers aim to reconstruct the path between the cylinder and the final bolt, only knowing the latter.

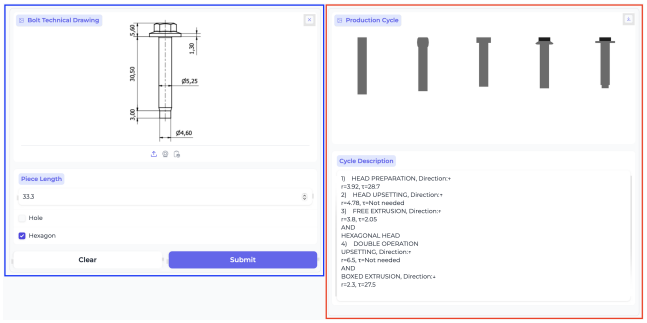

Figure 3: Interface of the tool developed for automatic production cycle computation. On the left, a user can supply the component's RFQ and its length in millimeters. Instead, on the right, the algorithm's output shows a rendering of the proposed cycle and the ordered list of operations together with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

hexagonal head[1]. On the right, in the red rectangle, the full production cycle (including the image and textual description comprising all operations together with their parameters) is shown. This production cycle corresponds to that one represented in Fig 2, showing that the cycle generated by our tool (Fig. 3) perfectly matches that produced by the process engineers (Fig. 2). We extensively evaluated the computing time of our tool, and we observed that the tool always returns an output 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 tools for process planning [Kumar, 2017], there are no examples of AI solutions dealing with the problem of production cycle design for bolts. To the best of our knowledge, ours is the first attempt to cast the bolts production cycle as an *AI model-based planning* problem and design algorithms to face it.

In particular, a finished component is the product of a sequence of operations applied to 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. If multiple sequences of operations lead to different feasible threads, we apply some pre-defined rule to break the tie (*e.g.*, choosing the thinnest thread). Flipping our perspective, the finished component is our starting state, and the actions we can make are the inverse of the operations available. Thus, there's a sequence of inverted operations (that, from now on, we will call operations) 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 Fig. 4. However, to search for the production cycle, we need a complete representation of the model and, thus, a formal model for both states and actions.

[1]It is required to specify if the component is drilled or has a hexagonal head since this information cannot be directly inferred from an image.

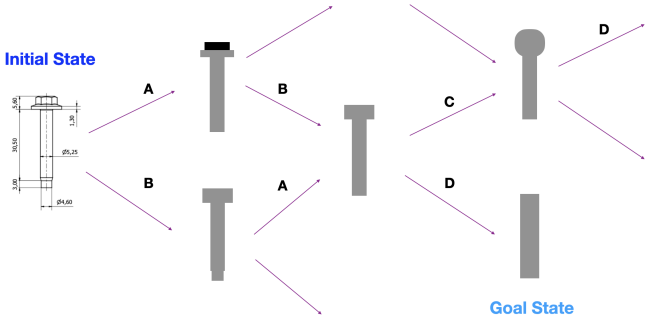

Figure 4: Example of an (inverse) operations tree leading to a metal thread starting from the finished component.

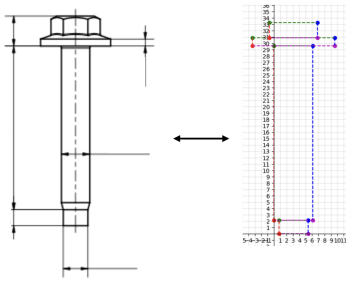

Figure 5: Mapping of a metal component's silhouette to Cartesian coordinates.

First, the formal model of a metal component describes both the final bolt and 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 safely encode the image of a metal component using its corners' coordinates, as in Fig. 5.

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

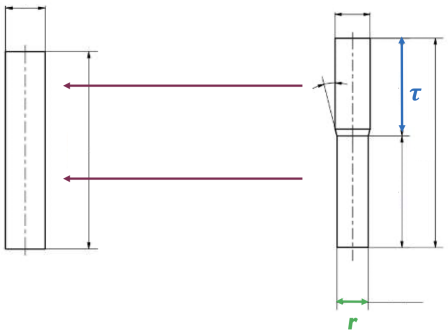

Figure 6: Example of inverse free extrusion.

trapezium-to-trapezium transformations. In Fig. 6, which represents a *free extrusion* (one of the most common forming operations), we can observe the following: a thread, whose silhouette is a rectangle, is transformed in two cylinders having the same total volume, and one having a smaller radius. Given height τ and 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 operation principle for our model. Finally, mechanical constraints can be easily ensured by quantities like τ and r since they can be expressed as relationships between such quantities.

4 Data-driven Heuristics for Computational Feasibility

Our problem presents more than 10 different transformations, most parametrized by continuous values. Moreover, some components may require a sequence of more than 6 operations to be formed. Thus, the search tree can be huge, precluding an exhaustive search for the correct production cycle. Our solution is to provide a data-driven heuristic to efficiently guide search and explore a dramatically smaller number of nodes. In particular, while performing a *depth-first search*, we compute heuristics that prioritize the actions most likely leading to feasible starting threads.

The available dataset is composed of both successful and failed (inverse) production cycles (*i.e.,* not leading to a thread in few operations or where the thread is not feasible due to constraints)[2]. 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 component 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 check the correctness of any automated tool doing this. To avoid this issue, we 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 a Vision

[2]A non-sufficiently populated dataset can be augmented by generating synthetic cycles, using information on the model dynamics.

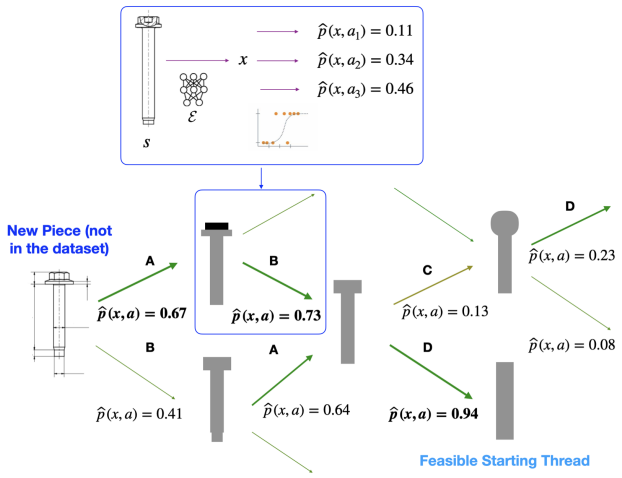

Figure 7: This scheme represents the working of our data-driven heuristic. When exploring the tree, for every encountered state, we embed it and predict the likelihood of success for every action. Then, actions are chosen from the most likely to the least likely.

Transformer (ViT) encoder $\mathcal{E}$, pre-trained on a large collection of images in a self-supervised fashion (DINO, [Caron *et al.*, 2021]).

Now, for every embedded state $\mathbf{x} = \mathcal{E}(s)$, we can evaluate for any available action a whether there is at least one path to a goal state, *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, namely $\widehat{p}(\mathbf{x}, a)$. In particular, we used 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.

Fig. 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, ours is the first tool of AI in this specific field. An automatic production cycle design allows Agrati S.p.A. to improve in-site operations planning, saving efforts, costs, 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.

Acknowledgements

This paper is supported by FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence).

References

[Caron *et al.*, 2021] 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.

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

[Kumar, 2017] 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.