

**COB-2023-2392**

**APPLYING A MULTI-ARMED BANDIT (MAB) PROBLEM-BASED TO  
OPTIMISE PRINTING ROUTES IN METAL GMA ADDITIVE  
MANUFACTURING**

**Rafael Pereira Ferreira**

Federal Institute of Education, Science and Technology of Maranhão (IFMA), Campus Barra do Corda, 65950-000, Barra do Corda, MA, Brazil

[rafael.ferreira@ifma.edu.br](mailto:rafael.ferreira@ifma.edu.br)

**Américo Scotti**

Center for Research and Development of Welding Processes (Laprosolda), Federal University of Uberlândia (UFU), 38400-901

[ascotti@ufu.br](mailto:ascotti@ufu.br)

**Abstract.** *GMA Additive Manufacturing (GMA-AM) may face setbacks when using conventional trajectory strategies, such as incomplete or excessive material deposition and deviations in areas with frequent arc strikes and stops. Space-Filling strategies like the "Pixel" strategy have been developed to address these issues, generating continuous trajectories that potentially minimise material accumulation and arc strikes and stops. However, computational efficiency remains a concern when dealing with complex geometries. This study proposed an upgraded version of the Pixel strategy, which incorporates reinforcement learning for trajectory planning. Grading up with the Multi-Armed Bandit problem solutions, the MAB-based-Pixel strategy aims at improving upon the existing Enhanced-Pixel strategy. Computational validation was performed. The results indicate that the MAB-based-Pixel algorithm achieves the shortest trajectory with fewer iterations than its predecessor, with the possibility to reduce the printing time. These findings establish the MAB-based-Pixel strategy as a promising solution for GMA-AM printing setbacks.*

**Keywords:** 3D printing, GMA-AM, Path planning, Reinforcement Learning, Multi-Armed Bandit problem.

## 1. INTRODUCTION

Figure 1 presents the most conventional routes for printing metallic parts using Directed Energy Deposition (DED) technology. The acronym DED, according to ISO/ASTM 52900, covers additive manufacturing processes in which thermal energy is used to pile up materials by melting them as they are being deposited. The electric arc is one of the forms of thermal energy. It is important to note that classifications such as the one outlined in Figure 1 are subject to inaccuracies. Even because there are many intersections between classes, separating a process between one category or another is difficult. In this way, classifications are more didactic. This figure also highlights the route that uses welding arc as the heat source, specifically the Gas Metal Arc Additive Manufacturing (GMA-AM) technique, the object of this work. One of the stages in the production chain of any DED technologies, including the GMA-AM, is trajectory planning.

Conventional trajectory planning strategies, such as Zigzag and Parallel Contour, are prone to flaws in the deposit, such as voids due to trajectory direction changes in sharper corners, bead starts and stops, and missed scan lines in the trajectory. An alternative solution to overcome the mentioned setbacks in trajectory planning for GMA-AM is utilising Space-Filling strategies, as Cox et al. (1994) proposed for NC tool-path generation. Recent studies by Vishwanath and Suryakumar (2022) have shown that fractal-based Space-Filling strategies in GMA-AM can improve temperature uniformity and reduce distortion in printed parts compared to conventional strategies. However, applying Space-Filling strategies to more complex parts remains an active area of research.

In line with this, the authors of this paper proposed a trajectory planning strategy named Pixel strategy, which the first version, known as Basic-Pixel, was introduced by Ferreira and Scotti in 2021. Using GMA-AM technology, they identified Space-Filling as a promising approach and demonstrated its potential applicability in printing complex parts, including those with sharper corner shapes. The Basic-Pixel strategy discretizes the surface into ordered nodes along the x-axis direction. Four trajectory planning heuristics create initial trajectories connecting these nodes, which are then optimized using the 2-opt algorithm. After that, a subsequent improvement of the Pixel strategy was introduced, called Enhanced-Pixel (to be published). This strategy was successfully applied to print parts with mixed characteristics (slender and bulky). Enhanced-Pixel strategy offers novelties to the current technology, by utilising a double axis ordering directions (x and y) and employing five heuristics of trajectory planning. Enhanced-Pixel strategy is basically an optimization approach, improving outcomes through extensive optimization. Notwithstanding, Enhanced-Pixel suffers from potential drawbacks, including increased computational time. This drawback is due to evaluating multiple

combinations of axis ordering and trajectory planning heuristics, requiring significant computational resources (requiring multiple iterations to achieve satisfactory results).

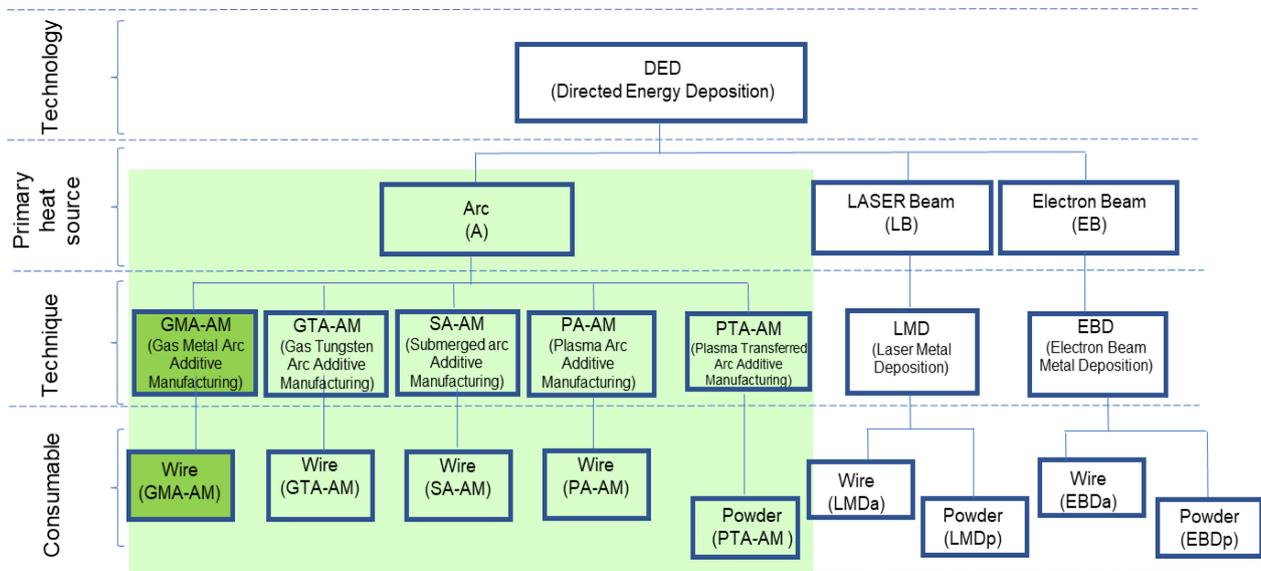


Figure 1. Classification of Additive Manufacturing of metals by Directed Energy Deposition (DED), with highlights for the Arc based modality (lighter shading) and the technique GMA-AM with the use of wires as consumable (rectangles with more intense shading)

Therefore, the authors propose using Reinforcement Learning to enhance the Pixel strategy by allowing the algorithm to learn and adapt during execution. One of the most suitable reinforcement learning techniques is based on a Multi-Armed Bandit (MAB) problem (see this technique background in section 2.2). The integration of MAB in an algorithm form is able to face the challenge of selecting optimal axis ordering and trajectory planning heuristics. Therefore, this work aimed at studying the possibility of using the MAB problem to accelerate the optimisation algorithm for trajectory planning of GMA-AM.

## 2. BACKGROUND

### 2.1 Reinforcement Learning

Learning through interaction with the environment is fundamental to our understanding of learning. Children, animals, and new employees, for instance, learn through playing and exploring without explicit teachers. Their daily connection to the environment provides valuable information about cause and effect, the consequences of actions, and the strategies for achieving goals. Throughout their lives, interactions with the environment serve as a significant source of knowledge about themselves and the world around them. And Reinforcement Learning is typically a learning method through interaction with the environment.

To better understand the idea behind algorithms that employ the Reinforcement Learning technique, the reader needs first to familiarise themselves with its main elements and concepts (terms), as it is proposed to do using Table 1. At its core, Reinforcement Learning revolves around the interaction between an agent and its environment. In other words, Reinforcement Learning problems involve an agent that learns by interacting with an environment to achieve a specific goal. Unlike other machine learning techniques, reinforcement learning provides rewards (positive or negative) for each action taken (positive rewards for good actions and negative rewards, regrets, for bad actions). In each agent interaction with the environment (called episode), the agent performs actions that return rewards or regrets. As the agent interacts with the environment, it learns to make better decisions according to a value function and policy. Furthermore, Ochi and Kamiura (2015) state that a good policy should reach a good trade-off between exploration and exploitation to maximise the reward.

To better illustrate the concepts discussed above, let one consider two practical examples, the first to demonstrate model-free reinforcement learning in training dogs and the second to showcase model-based reinforcement learning in a maze navigation scenario. In the context of training dogs, a reward-based system (model-free) is used to teach them specific commands. Through iterative trial and error, the dogs (the agent) learn to associate the commands with specific actions (for example, sitting). In a household environment (the environment), the training process involves recognising the dogs (positive rewards), typically with food, when they correctly respond to the desired action, and applying penalties

or punishments for incorrect behaviours (negative rewards). This repetitive exposure to commands (episode) and reinforcement of rewards and regrets gradually shape the dogs' behaviour, leading to the desired response. It is important to note that not all concepts and elements discussed earlier are necessarily utilised in this example, as their applicability depends on the complexity of the momentary problem.

Table 1. Reinforcement Learning-related terms and their definitions

Reinforcement Learning terms	Definition
Agent	Who or what takes actions that affect the environment.
Action	The set of all possible operations/moves the agent can make.
Environment	The place where the agent gathers information, interacts with its surroundings, and acquires knowledge through learning processes.
Reward	A provided feedback signal by the environment to the learning agent (it can be positive rewards, also called simply rewards, or negative rewards, also named simply regrets).
State	A particular situation in which the environment is at a given moment.
Episode	A set of interactions between the agent and the environment that starts from an initial state and ends in a terminal state.
Policy	Set of rules that an agent at a given state must follow to select an action to maximise the reward and avoid regrets.
Value function	The metric used to estimate the expected return or cumulative reward an agent can obtain in a given state.
Exploration	To gather information to understand the environment better.
Exploitation	To use the information from exploration to reach the target results.
Model-based	Type of Reinforcement Learning that constructs a representation of the environment to interact with and utilises it to simulate different actions and plan its decisions.
Model-free	Type of Reinforcement Learning that does not rely on a pre-defined model, but learns directly from interactions with the environment, only observing rewards and states

The second scenario comprises a robot (the agent) learning to navigate through a maze (the environment). The maze consists of a grid with walls, open paths, and a goal location. In this case, a model-based approach is employed in which a model of the environment is built to simulate and plan ahead. The model captures the dynamics of the maze, enabling the agent to predict the next state and reward based on its current state and chosen action. With the model in place, the agent can employ planning algorithms to simulate and evaluate different sequences of actions. The model helps the agent determine the optimal actions to take at different states. In model-based reinforcement learning, the policy is often updated based on the planning results and the estimated values of different actions in different states (value function).

In both cases (model-free and model-based), the agent must balance exploration and exploitation to achieve success. It explores the environment by taking different actions to gather information and learn more about (the dog's personality and the maze). At the same time, it exploits its learned policy to take actions that maximise its expected cumulative reward. By balancing exploration and exploitation, the agent can effectively train the dog, navigate the maze, and reach the desired goal.

Considering that modelling each of the possible environments in a Space-Filling path planning strategy (as in Enhanced-Pixel) would not be doable (depending on the shape of the different printable parts), it becomes evident that using a model-free method is more suitable. Among the various Reinforcement Learning methods, the Multi-Armed Bandit (MAB) algorithm is a special model-free technique. The use of the MAB algorithm has gained significant attention in various real-world applications, including recommender systems, information retrieval, healthcare, and finance. The reason for this widespread adoption lies in the algorithm's suitability for tackling sequential decision-making problems that involve recommending actions from a set, where the reward distribution of each action is unknown.

## 2.2 The Multi-Armed Bandit (MAB) problem

The denomination Multi-Armed Bandit is due to the analogy with a set of slot machines (slang for slot machine is "one-armed bandit"). Therefore, the name is not related to a robbery or a thief, yet to the probabilistic chances of winning more times using the same set of slot machines over and over. In the literature and various sources from search engines, the MAB problem is often symbolised by an octopus, with each tentacle playing in one machine simultaneously (see Figure 2). But this is symbolic (according to the understanding of the researchers involved in this project), because the concept of Reinforcement Learning and its derivative techniques, such as MAB problem, is based on probabilistic

(accepting that different machines would have a variable award rate would be unthinkable, unless biased by cheating). The symbolism is to show the use of several alternatives at the same time, and choose the ones to continue that were given more rewards, by change (a question of luck day of a player). If it were not based on probabilistic, there would be no need to continue with the same alternatives for another day the player would play (another episode).

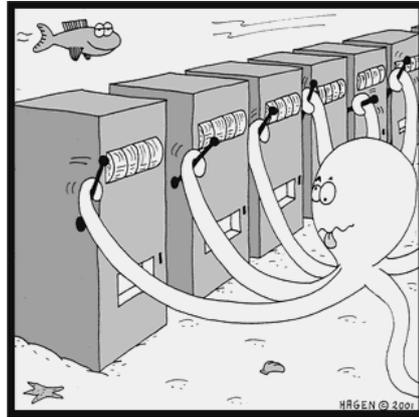


Figure 2. A symbolic representation of the Multi-Armed Bandit problem  
Fonte: <https://paperswithcode.com/task/multi-armed-bandits>. Accessed on: 03/07/2023

A good example to justify how the MAB problem works would be the case of online advertising. Imagine a company that wants to determine the most effective product advertisement (hereafter referred to only as advertising) to display on its website to maximise user engagement and click-through rates (famous market metrics nowadays). Just to explain, clicks are a marketing metric that counts the number of times users have clicked on a digital advertisement to reach an online property. In this scenario, the company treats each product advertisement as a "slot machine" and actions in the reinforcement learning nomenclature. At the start, the company's recommender system (the agent) has limited prior information about the effectiveness of each product advertisement. The goal is to allocate the available advertising resources to maximise the user clicks (the rewards). However, the company has a fixed and limited budget or resource to allocate among the different advertisements. To solve this problem using the MAB framework, the recommender system iteratively selects advertisements based on a strategy (policy) that balances exploration and exploitation. Initially, the recommender system may allocate resources to each advertisement uniformly or randomly (exploration). At the beginning, the same probability for each advertisement to be displayed, allowing it to gather feedback and collect data on the performance of each advertisement. As the recommender system receives rewards, such as ad clicks from users, along with the interactions with the environment (episode), it updates its estimate of the expected compensation or value associated with each advertisement (usually known as value function).

Based on the estimated values, the recommender system can recommend the advertisement with the highest value function (exploitation) to display to users. However, exploration mechanisms must be applied to ensure that other advertisements are given a chance to prove their effectiveness. According to Almasri et al. (2021), the most popular tools used in MAB for solving exploration-exploitation dilemmas are the  $\epsilon$ -greedy, Upper Confidence Bound (UCB), and Thompson Sampling (TS). They help strike the right balance between exploiting the best-performing advertisement and exploring other options to discover better-performing ones potentially. In dynamic scenarios where the effectiveness of advertisements can change over time due to shifts in user preferences or market trends, more exploration is required to adapt to these changes and continuously identify the best advertisement to display. By employing the Multi-Armed Bandit approach in this advertising context, the company can optimise its resource allocation and progressively learn which advertisement yields the highest rewards. This allows the company to improve the effectiveness of its advertising campaign, increase user engagement, and maximise click-through rates, ultimately leading to better business outcomes. After explaining the Multi-Armed Bandit (MAB) algorithm, the next section delves into applying the MAB problem in the Pixel strategy context, ultimately giving rise to the MAB-based-Pixel strategy.

### 3. THE PIXEL STRATEGY USING MAB PROBLEM

Recently, the authors of this work worked on an improved version of the Basic-Pixel strategy, presented in Ferreira and Scotti (2021). Enhanced-Pixel is the proposed name of this strategy version (not published yet). This strategy incorporates a new axis ordering and trajectory planning heuristic, allowing for extensive exploration of the optimisation space, aiming to generate the shortest trajectories while ensuring that the entire layer surface is filled. In brief, the process begins with the input by the user of the desired number of iterations. The algorithm then performs a loop where, in each iteration, a random node is chosen as the starting point for a trajectory generation (by a greedy algorithm). The nodes,

which have been discretised beforehand, are sorted based on their x and y coordinate axis ordering. Next, for each ordering along the x and y axes, the trajectory is generated compulsorily using a number of heuristic methods for trajectory planning. Concomitantly, the 2-opt algorithm is applied to optimise each of the trajectories. In total, ten trajectories per iteration are generated and stored in a matrix and saved. The algorithm then extracts the shortest trajectory distance per iteration from the matrix and stores it into another matrix, represented by "Store in a best value matrix." This matrix serves as a repository for all the good values found in each iteration. At this point, the algorithm either restarts for a new iteration, repeating the process, or proceeds to the end. However, the computational time required for evaluating each combination is a potential drawback. Despite potentially finding the best option for each iteration, the algorithm requires multiple iterations to converge. Therefore, a more intelligent recommendation for the axis ordering (AO) and heuristic of trajectory planning (HTP) is crucial to enhance results and achieve smaller trajectory distances.

The MAB-assisted algorithm, as proposed in this work, follows the same steps of entering the number of iterations and selecting a random starting node. But the Multi-Armed Bandit (MAB) concept is incorporated in the previous Enhanced-Pixel strategy by generating a single trajectory using a specific combination of the axis ordering and heuristic of trajectory planning (the action) amongst all combinations available. The combination recommendation at each iteration is guided by a chosen policy, which defines the criteria for choosing the most promising option. By utilising the MAB framework, the novel strategy (the agent) eliminates the need for the conventional greedy algorithm to generate multiple trajectories (time-consuming), as the Enhanced-Pixel version does. This strategy focuses instead on selecting the best combination (AO and HTP), accordingly to a MAB decision. Once a trajectory is generated per iteration, it can be stored. At this point, the algorithm (the environment) either restarts for a new iteration, repeating the process, or reaches the end (completing the episode).

A policy is used to choose among the combinations of axis ordering (AO) and heuristic of trajectory planning (HTP) in each iteration, and it is based on the trajectory distances (rewards). While the policy selects the AO and HTP combination (exploration), but if this selection is the one with the lowest trajectory distance (value function) in each iteration, it operates from an exploitation perspective. However, a good policy should balance between exploration and exploitation. The Thompson Sampling (TS) policy tool was adapted for the proposed strategy. As this policy tool has the objective of minimising the trajectory distance, its value is considered the average value of the trajectory distances obtained by a given combination of AO and HTP ( $h$ ) until that iteration  $t$ . Its value function is represented by  $Q_t(h) = (\sum_{t=1}^n D_t(h)) / (N_t(h))$ , where  $N_t(h)$  is the number of times that  $h$  has been chosen until that iteration  $t$ , and,  $\sum_{t=1}^n D_t(h)$  is the sum of the trajectory distance given by  $h$  until that iteration  $t$  ( $1 \leq t \leq n$ ), where  $n$  is the number of iterations.

Thompson Sampling has gained considerable attention for its effectiveness. According to Jain et al. (2016), this policy tool was specifically developed for scenarios where the rewards from actions follow a Bernoulli distribution, i.e., they can be categorised as successes or failures. As a Bayesian method, the beta distribution is a suitable model for estimating the success probability of each slot machine at each iteration, as it takes into account the number of successes and failures observed up to that point. The beta distribution is a continuous probability distribution defined on the interval  $[0, 1]$ . It is characterised by two positive shape parameters,  $\alpha$  and  $\beta$ , which act as exponents of the random variable and shape the distribution. For example, as shown in Figure 3, when  $\alpha=1$  and  $\beta=1$  (dashed yellow lines), a uniform distribution is obtained. This means that each action (or  $h$ ), i.e., the combination of the axis ordering and heuristic of trajectory planning, will have the same probability of selection at each iteration. Dirkx and Dimitrakopoulos (2018) state that this approach is a good practice to initialise the algorithm for applying the Thompson policy tool (prior distribution), so that a good exploration phase of the heuristics occurs at the beginning.

As the iterations occur, the values of  $\alpha$  and  $\beta$  are updated according to the following protocol:

- a) The parameter  $\alpha$  is incremented every time the selected  $h$  generates a trajectory with a distance lower than  $Q_t(h)$  (the mean value of the trajectory distances obtained up to that iteration) in an iteration. This can be considered a good result (a success case) achieved by that selected  $h$ , so the value of  $\alpha$  is increased. The Beta distribution with higher values of  $\alpha$  behaves as shown in Figure 3 (dashed green lines), where there is a higher probability of that  $h$  being selected in the next iteration;
- b) On the other hand, the parameter  $\beta$  is incremented for every unsuccessful iteration, that is, when the selected  $h$  generates a trajectory with a distance greater than the mean value of the trajectory distances obtained up to that iteration. This causes the distribution behaviour to be higher for low probabilities, as illustrated in Figure 3 (dashed blue lines), which reduces the possibility of that  $h$  being selected in the next iteration.

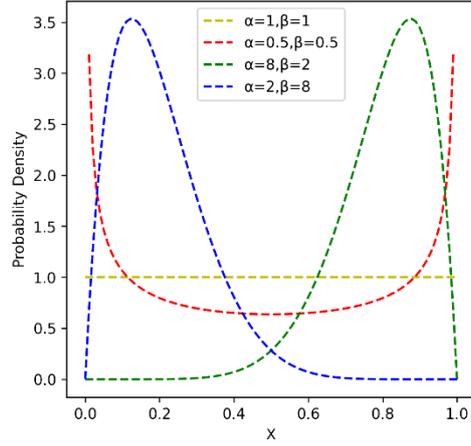


Figure 3. Beta Probability density function (PDF) with varying values of the parameters  $\alpha$  and  $\beta$

Therefore, at each iteration of the algorithm, the estimation of the success probability of the  $h$  is updated. The selection of the  $h$  at each iteration  $t$  is based on the estimated probability that is considered optimal at that moment, as shown in Eq. (1):

$$h_t = \operatorname{argmax}_{h \in H} \tilde{Q}_t(h) \quad (1)$$

where  $\tilde{Q}_t(h)$  is the estimate of the success probability of each  $h$ , which is sampled a priori from the beta distribution.

#### 4. COMPUTATIONAL VALIDATION

##### 4.1 Methodology

One rectangular-ring-shaped figure was studied (Figure 4) to evaluate a possible computational advantage of the novel strategy (MAB-problem based) over the Enhanced-Pixel strategy (previous version). The Thompson Sampling was used as the policy tool. Aiming at a higher performance of each tool, an extensive episode of 500 iterations was set for each path planning strategy. It is important to state that the number of iterations which go through the combinations axis ordering (AO)-heuristics of trajectory planning (HTP) are not the same in both strategies. While Enhanced-Pixel evaluates 10 sets of combined AO-HTP per iteration, the novel strategy evaluates only one combination per iteration. Thus, for comparison purposes, each of the 10 trajectory generations produced by a combination of AO-HTP was considered one iteration in the Enhanced-Pixel strategy. As a result, the iteration criteria turned equivalent between both strategies, by making  $10 = 1$  (10 for Enhanced-Pixel and 1 for the MAB-assisted strategy).

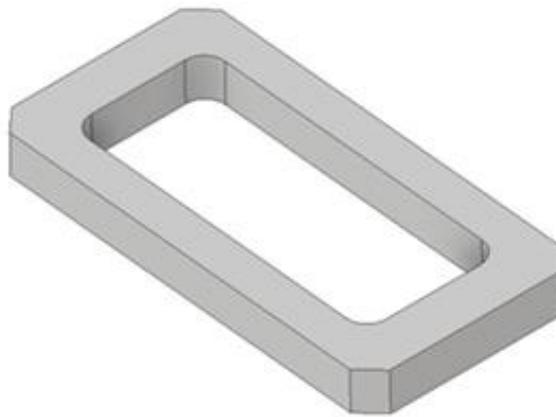


Figure 4. A rectangular-shape figure (630 nodes)

This study considered two criteria for the analyses: Convergence Analysis about the minimisation process of the trajectory distance and Analysis of Cumulative Regret. Convergence Analysis is standard in optimisation studies, while Cumulative Regret is commonly used for comparison between MAB algorithms. In practical terms, Convergence

Analysis examines the speed at which the minimal trajectory distance is achieved for each strategy over the course of iterations. It helps identify the best strategy based on minimal trajectory distances. Cumulative Regret, the second criterion, represents the accumulated loss resulting from the chosen policy tool not selecting the optimal AO-HTP combination. Eq. 2 is the way to calculate the progress of this metric, along with the number of iterations. The ideal combination, yielding the best reward up to a given iteration (represented by  $D^*$  in Eq. 2, is compared to the actual combination chosen at that iteration (represented by  $D_{t,h}$ , Eq. 2). More regrets for the same number of iterations mean fewer positive rewards. Cumulative Regret tells how much one loses by betting more on the wrong combination AO-HTP; exploring more than exploiting. The analysis of Cumulative Regret curves follows a similar approach to Convergence Analysis for the minimisation process, with the distinction that regret increases as the number of iterations grows, and higher values indicate larger regrets (poor results).

$$Regret = \sum_{t=1}^n [-(D^* - D_{t,h})] \quad (2)$$

## 4.2 Results and Discussions

The first analysed criterion was carried out through the convergence curves, presented in Figure 5(a). The comparison between the proposed MAB-based strategy (orange line) and the Enhanced-Pixel (blue line) strategies showed the predominance of the MAB-based-Pixel strategy. Cumulative Regret was the second criterion evaluated and its curves is illustrated in Figure 5(b), and it is evident from the part studied that the Enhanced-Pixel strategy (blue line) has a higher cumulative regret than the MAB-based-Pixel (orange line). This is because the Enhanced-Pixel strategy follows a non-intelligent structure that does not prioritise the best combinations of AO and HTP. Consequently, it can always yield poor results and does not learn from them. This leads to an increase in regret and makes this strategy unreliable. On the other hand, the MAB-based-Pixel strategy acquires knowledge about the AO and HTP that generate the best results as the number of iterations progresses. After evaluating the two criteria, it was found that MAB-based-Pixel performed better than the Enhanced-Pixel strategy.

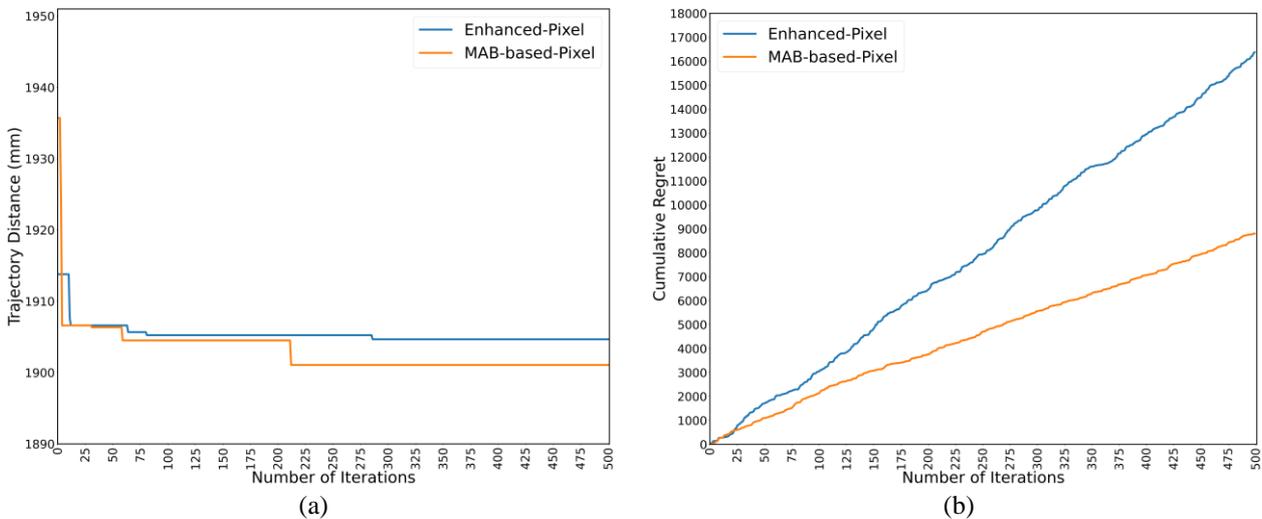


Figure 5. Convergence and cumulative regret charts of the trajectory plan for the studied part

## 5. CONCLUSIONS

The concept of the MAB problem in the algorithm was applied to the space-filling-like Enhanced-Pixel strategy to optimise the trajectory, as stated in the objective of this work. The proposed MAB-Based algorithm checks only one of the 10 axis ordering (AO) and heuristics for trajectory planning (HTP) combinations per iteration. In contrast, Enhanced-Pixel checks each of the ten combinations of AO and HTP per iteration. In summary, it was noted that:

- The algorithm performance gain shows that MAB-based-Pixel converges to the shortest trajectory, with possibility to significant savings in printing time, the latter due to fewer iterations than its predecessor (Enhanced-Pixel). However, it is worth noting that the solution applied in this novel strategy is based on probabilistic concepts, and one cannot expect the novel version to beat the predecessor in 100% of cases;
- Therefore, the implementation of Reinforcement Learning through the MAB problem succeeded well towards “grading up” the Pixel family of Space-Filling trajectory planners.

## 6. REFERENCES

- Almasri, M.; Mansour, A.; Moy, C.; Assoum, A.; Le Jeune, D.; Osswald, C., 2021. "Distributed competitive decision making using multi-armed bandit algorithms". *Wireless Personal Communications*, Vol. 118, p. 1165-1188. <https://doi.org/10.1007/s11277-020-08064-w>
- Cox, J.J.; Takezaki, Y.; Ferguson, H.R.P.; Kohkonen, K.E.; Mulkay, E.L., 1994. "Space-filling curves in tool-path applications". *Computer-Aided Design*, Vol. 26, p. 215-224. [https://doi.org/10.1016/0010-4485\(94\)90044-2](https://doi.org/10.1016/0010-4485(94)90044-2)
- Dirkx, R.; Dimitrakopoulos, R., 2018. "Optimizing Infill Drilling Decisions Using Multi-Armed Bandits: Application in a Long-Term, Multi-Element Stockpile". *Mathematical Geosciences*, Vol. 50, p.35–52. <https://doi.org/10.1007/s11004-017-9695-9>
- Ferreira R.P., Scotti, A., 2021. "The concept of a novel path planning strategy for wire + arc additive manufacturing of bulk parts: pixel". *Metals*, Vol. 11(3). <https://doi.org/10.3390/met11030498>
- Jain, S.; Bhat, S.; Ghalme, G.; Padmanabhan, D.; Narahari, Y., 2016. "Mechanisms with learning for stochastic multi-armed bandit problems". *Indian Journal of Pure and Applied Mathematics*, Vol. 47, p. 229–272. <https://doi.org/10.1007/s13226-016-0186-3>
- Ochi, K.; Kamiura, M. Overtaking method based on sand-sifter mechanism: Why do optimistic value functions find optimal solutions in multi-armed bandit problems? *Biosystems*, v. 135, p.55-65, 2015. <https://doi.org/10.1016/j.biosystems.2015.06.009>
- Vishwanath, N.; Suryakumar, S., 2022. "Use of fractal curves for reducing spatial thermal gradients and distortion control". *Journal of Manufacturing Processes*, Vol. 81, p. 594-604, 2022. <https://doi.org/10.1016/j.jmapro.2022.07.028>

## 7. RESPONSIBILITY NOTICE

The authors are the only ones responsible for the printed material included in this paper.