A Framework for Manufacturing System Reconfiguration based on Artificial Intelligence and Digital Twin*

Fan Mo[1], Jack C. Chaplin[1], David Sanderson[1], Hamood Ur Rehman[1], Fabio Marco Monetti[2], Antonio Maffei[2], and Svetan Ratchev[1]

[1] Institute for Advanced Manufacturing, University of Nottingham, Nottingham NG8 1BB, United Kindom,
[2] KTH Royal Institute of Technology, Department of Production Engineering, Brinellvägenn 68, 114 28 Stockholm, Sweden

Abstract. The application of digital twins and artificial intelligence to manufacturing has shown potential in improving system resilience, responsiveness, and productivity. Traditional digital twin approaches are generally applied to single, static systems to enhance a specific process. This paper proposes a framework that applies digital twins and artificial intelligence to manufacturing system reconfiguration, i.e., the layout, process parameters, and operation time of multiple assets, to enable system decision making based on varying demands from the customer or market. A digital twin environment has been developed to simulate the manufacturing process with multiple industrial robots performing various tasks. A data pipeline is built in the digital twin with an API (application programming interface) to enable the integration of artificial intelligence. Artificial intelligence methods are used to optimise the digital twin environment and improve system decision-making. Finally, a multi-agent program approach shows the communication and negotiation status between different agents to determine the optimal configuration for a manufacturing system to solve varying problems. Compared with previous research, this framework combines distributed intelligence, artificial intelligence for decision making, and production line optimisation that can be widely applied in modern reactive manufacturing applications.

Keywords: Artificial intelligence, multi-agent programming, digital twin, process simulation.

1 Introduction

Digital transformation is the integration of digital technologies in a company to improve performance and productivity. Digital transformations can be applied to any area of the business, and impacts the manufacturing shop floor, the business processes and models used, and the overall customer experience [1]. Digital twins and simulations are increasingly used to plan and optimise processes [2], and given the large volumes of data generated by these, there is a push toward using smart data analytics tools to analyze the stream of data and let managers take responsible, rapid decisions to regulate and improve productivity [3]. Machine Learning (ML) algorithms provide solutions for processing large volumes of data[4], for example, processing and making predictions with sensor signals [5].

By applying simulation and digital twins, it is more realistic to implement reconfigurable systems and customized, demand-based production. The application of artificial intelligence to the digital twin-based streams of data allows companies to analyze and react to sudden changes in demand or disruptions [2]. The reconfiguration of production processes in response to external changes is a difficult challenge. The scope of system reconfiguration includes system layout, process parameters, operation time of multiple assets, sequence of the operations, and material handling systems.

This paper proposes a new framework to enable the system to do different kinds of reconfiguration through the application of existing artificial intelligence algorithms, such as the genetic algorithm and particle swarm optimization (PSO), or newly developed algorithms. With this framework, the manufacturing system reconfiguration can be simulated at first in the virtual environment, and then transferred to the actual production plant. We developed a digital twin environment that simulates the manufacturing process with multiple industrial robots performing multiple tasks. A data pipeline with an API that integrates artificial intelligence is built in the digital twin. With the help of artificial intelligence and the digital twin, the performance of the decision-making process and the reconfiguration process is improved. Finally, a multi-agent program to communicate between different agents and share the negotiation status that determines an optimal configuration for the manufacturing system is made. Section 2 presents a brief overview of works related to the same topic. Section 3 introduces a new framework for manufacturing system reconfiguration and a framework of the reconfiguration engine. Section 4 describes an application with multiple robots, which applies the previously mentioned reconfiguration framework. Section 5 presents the conclusion and suggestions for future research.

* Supported by DiManD Innovative Training Network (ITN) project funded by the European Union through the Marie Sklodowska-CurieInnovative Training Networks (H2020-MSCA-ITN-2018) under grant agreement no. 814078.

2 Related work

Increasing market demand for mass customization is forcing manufacturers to adopt rapid reconfiguration capabilities, allowing for the sudden change of system configuration – whilst maintaining full system effectiveness – in the event of unpredictable customer demands, line failures or need for maintenance [6]. There is a large body of research on automation agent architectures – distributed artificially intelligent programs that collaborate to solve problems – that allow for 'on the fly' reconfiguration of manufacturing systems, exploiting asset flexibility and control to change the behaviour of the manufacturing system [7]. Multi-agent systems and the concept of cyber-physical systems (CPS) were key contributors to research in the reconfiguration of manufacturing environments at the beginning of Industry 4.0 [8,9].

Emerging artificial intelligence techniques are now being applied to the problem of manufacturing reconfiguration [10]. Human-machine collaboration is an increasingly important concept in flexible and reconfigurable manufacturing systems, and artificial intelligence is also useful here, as well as considerations of safety in such environments [11]. Another important topic that emerged from smart manufacturing's adaptability to demands and the consequent issues of decision-making based on data is the self-repair ability of smart systems; a recent study handled it again using an artificial intelligence approach to select the best strategy, based on product and module swapping, operation rescheduling and reconfiguration, to significantly reduce the capacity loss [12]. The volume of data coming from this knowledge base is a challenge to handle, but also represents an opportunity exploitable for the development of the Digital Twin (DT) concept, a virtual representation of a physical asset where both counterparts are connected to each other and are dynamically evolving through the whole life cycle [13]. DT emerged as a concept for monitoring processes and is evolving towards being an instrument used for reconfiguration. They can be used in parallel with an artificial intelligence approach to reconfiguring human-robot collaborative assembly lines [14,15]; they model the real world and exchange data back and forth with the various assets of a manufacturing environment to be able to affect the system's decisions and behaviour [16]; and they can represent the base on which to build smart, fast and responsive manufacturing systems based on robotics assets [17,18].

An implication of these proposed approaches is that integrating complex adaptive systems with artificial intelligence techniques, combined with flexible multi-functional manufacturing assets such as robotics, will result in a continuously evolving and changing knowledge base. This volume of data in this knowledge base is both a challenge, but also an opportunity for enabling real-time learning of new reconfigurations and behaviours in manufacturing areas such as production, logistics, and assembly. All the different methods of artificial intelligence aim to achieve the best possible performance in terms of scheduling, efficiency, and productivity, helping to apply autonomous technologies and behaviour to robots, manipulators, and eventually the whole system. Given the increasing availability of data, and the ability to be iteratively applied, this approach is also self-improving [19].

3 Framework

3.1 Overview of manufacturing system reconfiguration system

This section proposes a framework for the intelligent optimisation of manufacturing system reconfiguration management. It also provides a concept of the reconfiguration engine with multi-agent system integration. This reconfiguration framework covers two levels; the manufacturing execution level, and the controller level. The manufacturing execution level interacts in real-time with the controller level to deal with the customer's orders and make decisions. The controller level will receive the information from the manufacturing execution level and send the related control command to the physical device. The reconfiguration engine is a digital environment with multi-agent system integration and enables the system to determine the optimum reconfiguration based on the selection criteria it receives from the decision engine and the reconfiguration data bank (Figure 1). Layout, cost, time, sequence of the operation, and others can be used as the selection criteria. This framework consists of the following steps:

(1). **Product requirement**

The first step in the framework is to generate the product requirements, as these will specify the operations the system must perform. The bill of resources and the bill of processes will be generated, breaking the requirements into the equipment needed, and the order of processes required to manufacture the product. This step can be done with several different approaches, including the customer specifying exactly the bill of resources and bill of the process themselves, manually creating the bill of resources and process, or utilising a semantic model and matching algorithm to generate these automatically.

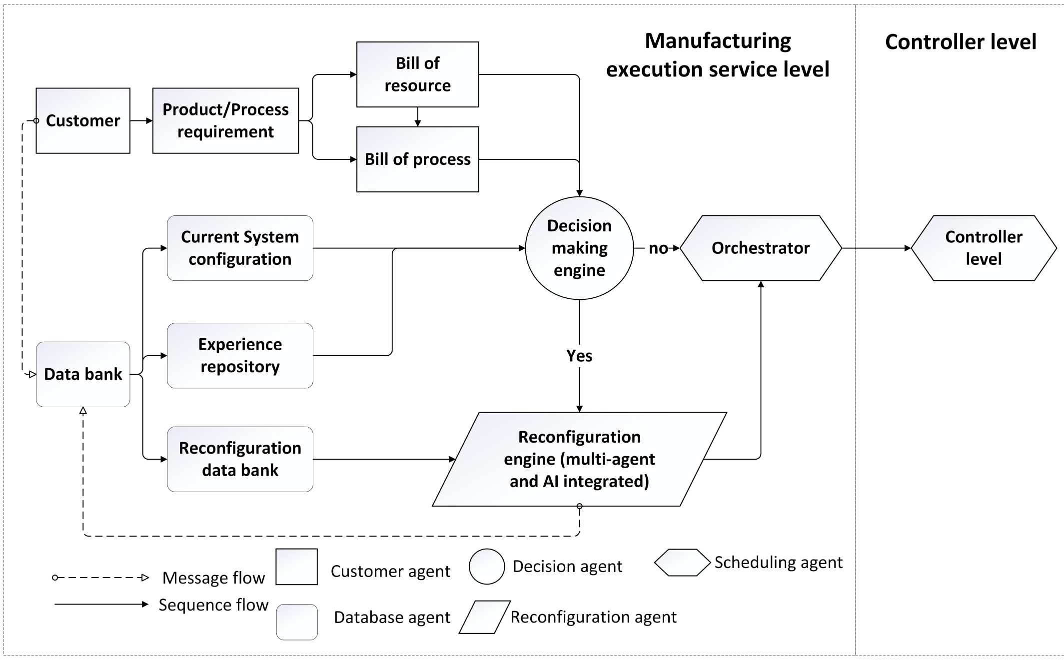

Fig. 1. Overview for system reconfiguration

(2). **Experience data bank**

The data bank is a database for storing data from (i) the current system configuration, (ii) prior experience and configurations, and (iii) stored algorithms and simulations required to support reconfiguration decision making. The current system configuration data (i) consists of information about the current system, such as process information, equipment information, layout information, machine runtime condition, and value-chain information. Standard data models are used to represent the current system configuration information. For instance, asset administration shells can be used to represent system configuration information [20]. The experience repository (ii) is a data bank of historical data for helping the system's decision making. For instance, configuration information from previous system reconfigurations and processes and their performance. The reconfiguration data bank (iii) contains the offline data components of digital twins for the current use case, the simulation part of the digital twin optimisation algorithms, simulation results from the previous system configurations, and libraries of PLC code for updating the control programs. To store the offline data of the digital twin, multiple approaches are used. For example, the simulation model, the interfaces to access data in the live physical system, and other information such as device information, operation information, and signal information are all stored in the database for future use.

(3). **Decision-making engine**

The information from the bill of resource/process, system configuration, and the experience repository are analyzed by the decision-making engine. The decision-making engine determines if a reconfiguration is needed based on different aspects, such as cost, time, utilization of the equipment, and productivity. If a reconfiguration is needed, the decision-making engine will give recommendations via an agent to the reconfiguration engine about the reconfiguration approach to be used based on previous successful experiences. The reconfiguration approach can be an algorithm or a combination of algorithms that can be best utilized.

(4). **Reconfiguration engine**

The reconfiguration engine is used by the decision-making engine to begin the reconfiguration process. Depending on the type of system being reconfigured, different approaches are selected for use, such as inbound reconfigurable transportation systems [21], system layout problems [22], reconfigurable manufacturing system configuration selection [23], or planning & scheduling in reconfigurable manufacturing systems [24,25]. The optimized approach will be sent to the orchestrator after the reconfiguration has been made. Additionally, the optimized approach will also be stored in the data bank for future reference.

(5). **Orchestrator**

The orchestrator has a similar function to an ERP (Enterprise resource planning system) and can work with existing ERP systems. The orchestrator is responsible for arranging, controlling, and optimizing workloads in production. The orchestrator will allocate manufacturing resources, plan human resources, and plan and schedule production processes, based on the information received from the decision-making and reconfiguration engines.

(6). **Multi-agent integration**

To make the system resilient and robust, the multi-agent approach is used. There are five types of agents in this framework: customer agent, database agent, decision agent, reconfiguration agent and scheduling agent (see Table 1).

Table 1. Agent description

Agent Type	**Functionalities**
Customer agent	The customer agent handles new product requests and may read the information from the current system and get the necessary information from the data bank to generate a bill of process.
Database agent	The database agent is responsible for generating data models and storing information. It responds to requests from the decision making engine to supply the necessary data for the decisions.
Decision agent	The decision agent will submit requests for relevant information from the database agent and customer agent and will decide if a reconfiguration is needed based on the decision criteria.
Reconfiguration agent	The reconfiguration agent responds to requests from the decision agent for changes to the system configuration. It invokes the artificial intelligence algorithms as required based on different optimization scenarios.
Scheduling agent	The scheduling agent receives the output from the decision agent and the reconfiguration agent and determines how the system should respond to enact the new configuration at the control level.

3.2 Reconfiguration engine algorithm based on digital twin

The reconfiguration engine can select multiple algorithms (either provided by the user or previously stored in the data bank) to determine the optimum system configuration for a given product. In complex robotic manufacturing use cases, many types of optimisation decisions are needed to specify the reconfiguration. These include robot locations, robot paths, and the operation sequence. A digital twin is needed to simulate the process to test multiple scenarios and optimise it. Here we present a framework for utilising digital twins and multi-objective optimisation algorithms. The framework consists of three parts: the *digital twin*, the *training gym*, and the *communication block*.

The *digital twin* environment will be taken from the experience repository. Digital twins themselves are a combination of three primary components: a virtual twin, a physical twin (i.e. the physical manufacturing system), and the data flow component. The data flow component is used to feed data from a physical twin to its virtual twin and returns information from the virtual twin.

The digital twin and the training gym share data via the *communication block*. The communication can be sockets, OPC-UA, MQTT, TCP/IP or other communication protocols depending on the application domain, and can be dynamically chosen by the reconfiguration engine depending on the system context. For instance: if a cloud-based environment is needed, then the MQTT approach could be used to enable communication.

In the *training gym*, different artificial intelligence algorithms and approaches are used to optimize the simulation in the digital twin to meet the different optimization targets. For example, if the optimization target is multi-objective, then multi-objective optimization algorithms are selected. If the manufacturing system looks to optimise a single parameter with the new product – such as faster operation, or lower cost – then reinforcement learning can be used in the training gym. For highly complex or difficult problems, deep reinforcement learning will be suggested by the decision engine. After the optimized result is found and evaluated successfully in the simulation environment, the optimized result (such as the optimized robot program, robot path, and the structure of the production line) will be applied to the physical device via the orchestrator. The orchestrator may make direct changes to PLC code or robot controllers to instantiate the change or make recommendations to human operators to make the changes manually. The orchestrator will receive change requests via the communication protocol suggested by the reconfiguration engine.

4 Application

This section describes a demonstrator used to verify the effectiveness of the proposed approach in achieving system reconfiguration. The following example illustrates the reconfiguration of a pick and weld production station, using Siemens Tecnomatix Process Simulate as the digital twin enabler. As shown in Figure 2, this use case consists of two KUKA KR2700ultra robots, one conveyor for transporting the product, one welding station, and one work table for storing the parts after welding. For the two KUKA robots, one robot (robot 1) is armed with the pick and place end effector. The other robot (robot 2) is equipped with an arc welding gun.

Normally, robots are drilled into concrete or mounted to metal plates, so cannot quickly be moved. This application applies a modern reconfigurable approach that allows the robot to find the optimized position at first in the simulation environment and then applied to the real physical device. This is either done before the cell is commissioned to determine the optimal location, for situations where the robot can be moved to different locations with a reconfigurable flooring solution (which can be seen in Figure 2), or the robot will (depending on payload restrictions) be placed on an automated ground vehicle (AGV) and moved to the new position after it receives the optimized location. The process consists of five steps:

(1). The product (yellow part) is placed on the conveyor by the previous process in the production line.
(2). The part will be transported via the conveyor to robot 1 for picking.
(3). Robot 1 will pick the part up and move it to the welding station.
(4). After the part is successfully put on the welding station, robot 2 will perform arc welding on the product.
(5). After welding, robot 1 will pick the product up and put it on the worktable, and return to the home position.

In this use case, it is assumed that the customer wants a welded cubic product and the decision-making engine has already decided that a system layout reconfiguration is needed. Both robots need to be relocated to their optimal locations (based on the suggestions from the reconfiguration engine) in order to optimize the production time of the process. Optimizing process time is a common industrial requirement, for example, to minimize cycle time or meet a takt time. An additional constraint on the optimisation is ensuring that neither robot collides with other parts during operations.

The application for the framework of the system reconfiguration engine for this use case is shown in Figure 3. The digital twin environment will send operation time and collision situation to the training gym via a socket (C#) based on Tecnomatix .NET API. Tecnomatix .NET API is connected to the Tecnomatix Process Simulate simulation environment with Tecnomatix .NET viewers.

In the training gym, after optimization via the selected artificial intelligence approach, the updated robot location will be sent to the digital twin, to get a new operation time and check for collisions. After all the iterations, an optimised location of the two robots will be found. The results (locations of the robots) will at first be validated in the digital twin. Then the final optimized robot locations will be sent to the orchestrator. It is the role of the orchestrator to communicate the robot locations to the relevant resource that will make the change - e.g. the human worker to arrange the robot move, or the AGV the robot is mounted on. Finally, the physical KUKA robots will receive the new optimized robot location via PLC. In this example, the CEE (Cyclic Event Evaluator) simulation mode in Tecnomatix Process Simulate is used. The CEE, which functions as a PLC, is used inside the Tecnomatix Process Simulate to control how a typical robotics simulation progresses using logic. Once the start signal is true, the simulation will start. Originally, there are no robot move relocation functions defined. With Tecnomatix .NET API, the move operation can be generated in each simulation. In the first iteration, the training gym will send random coordinates of both robots to the simulation environment, and then two object flow operations in Tecnomatix Process Simulate will be generated and be linked with other operations.

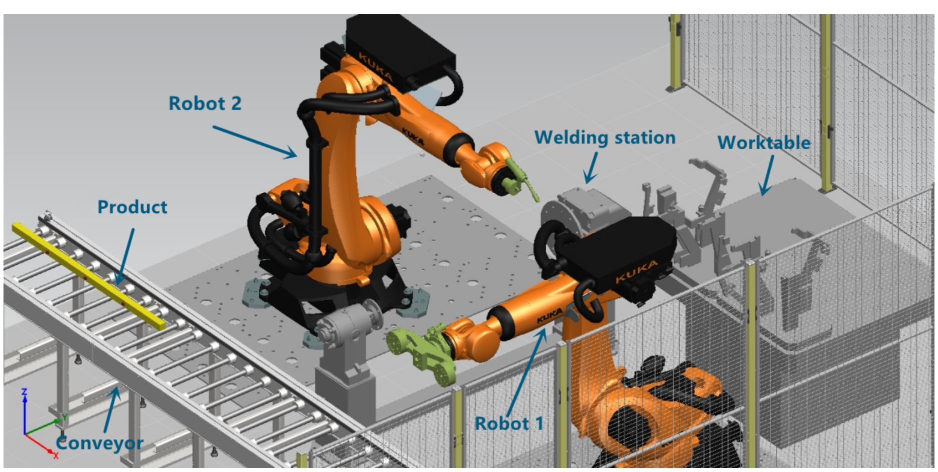

Fig. 2. Layout of the simulation environment

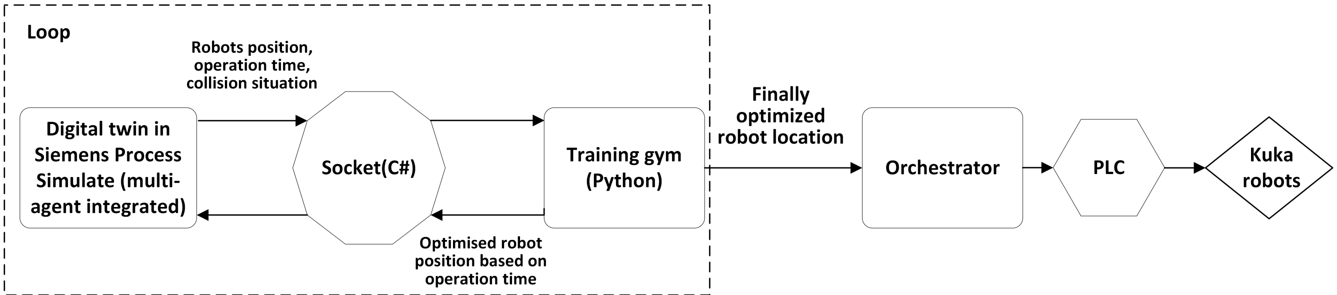

Fig. 3. Framework for reconfiguration engine

To let the result converge faster, penalties are given to situations where a robot collides with other parts or cannot complete the operation due to reachability. As a multi-objective optimization problem, multiple different artificial intelligence approaches can be applied to this use case. The reconfiguration engine can choose which approach is the best solution based on past experience. In this use case, two different types of optimization algorithms will be used to do robot location optimization. One is the global Particle Swarm Optimization (PSO) approach. The other is a Genetic Algorithm (GA). The sequence flow of these two algorithms is listed below (Figure 4). PSO is a bio-inspired algorithm that searches for an optimal solution through iterative improvement. In this use case, the robot's initial random locations will be set in the initialization swarm, and progressively optimized based on the simulation time received from Tecnomatix Process Simulate. GA is a search heuristic inspired by the theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring for the next generation. The hyper-parameter for the PSO and GA are listed below (Table 2).

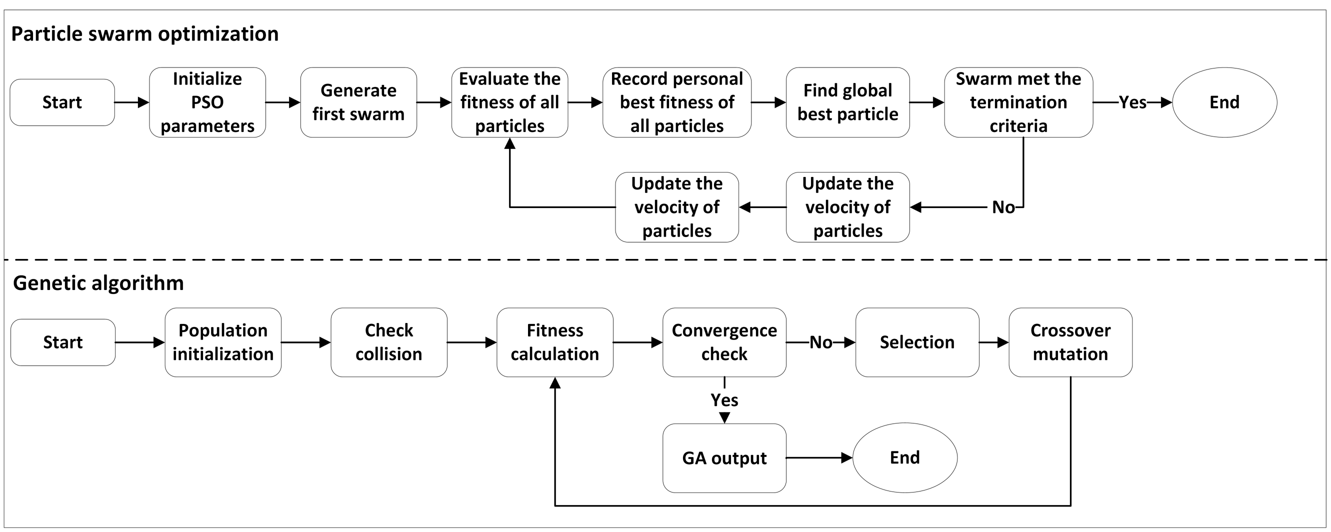

Fig. 4. Flow diagram for PSO and GA algorithms

After the training gym has finished the optimization, the optimized location of the robots will be found. The key performance indicators in this example are the cell's total operation time and the number of iterations the algorithm took to find an optimal operation time. The comparison of these two approaches is listed below (Figure 5 and Figure 6). The reconfiguration engine will choose the scenarios based on different performances. From Figure 5 and Figure 6, we find that the GA algorithm converges quicker than the PSO algorithm approach for this use case. Furthermore, the optimized time found by the GA algorithm is almost the same as the optimized time found by the PSO algorithm approach. The best process time found by the PSO algorithm is 23.72 seconds compared to 23.74 seconds for the GA approach. In this situation, the reconfiguration engine will recommend using the GA algorithm as the optimized approach and can save this information in the data bank for future reference.

Table 2. Hyperparameters for Genetic algorithm and PSO algorithm

Hyperparameters	PSO algorithm	Genetic algorithm
Population size		10
Number of genes		6
Number of parents mating		4
Swarm size	10	
Acceleration coefficient	c1 = 1.5, c2 = 1.5	
Inertia weight	w = 0.5	
Pick robot's coordinates	([-2000, 2000], [-500, 500], [0, 0])	
Welding robot's coordinates	([-2000, 2000], [-500, 500], [0, 0])	
Number of iterations	50	
Penalties	50	

There are some limitations to the approach used in the applications. For instance, if the initial position of the robot is too far from the work-piece where they can't execute the operations, the result will not easily converge – if the robots can't execute the operations, the training gym will always receive a penalty (bigger than the operation time) instead of the valid operation time. Convergence speed is also highly dependent on the chosen penalty value.

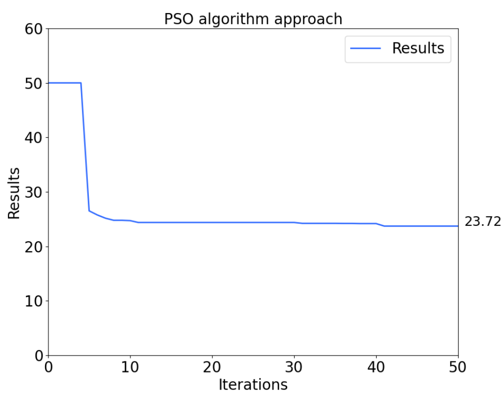

Fig. 5. PSO approach

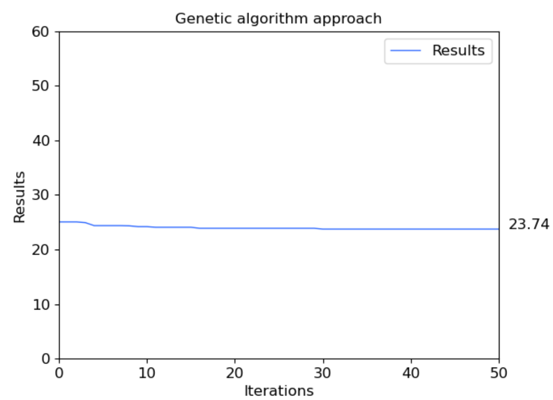

Fig. 6. Genetic algorithm approach

5 Conclusions and future work

The design and operation of manufacturing systems are increasingly changeable as the markets require quicker responses to new products, supply disruptions, and volume demands. Reconfiguration and optimisation of the production process in response to external changes is a difficult challenge for complex and flexible systems. This paper proposes a new framework to enable the system to find optimized operation parameters and configurations autonomously. With this framework, the manufacturing system reconfiguration can be enabled at first in the simulation environment and then deployed to the physical system. Though currently applied to a simulation environment, our next step is to apply the framework to our physical robotic manufacturing cells. More key performance indicators will be introduced as optimisation criteria for this framework. Lastly, more complicated use cases will be considered to use this framework. Specifically, applications where manufacturing is not limited to one workstation.

References

1. G. Westerman, C. Calméjane, D. Bonnet, P. Ferraris, and A. McAfee, "Digital Transformation: A roadmap for billion-dollar organizations," *MIT Center for digital business and capgemini consulting*, vol. 1, pp. 1–68, 2011.
2. C. da Cunha, O. Cardin, G. Gallot, and J. Viaud, "Designing the digital twins of reconfigurable manufacturing systems: application on a smart factory," *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 874–879, 2021.
3. J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli, "Smart manufacturing, manufacturing intelligence and demand-dynamic performance," *Computers & Chemical Engineering*, vol. 47, pp. 145–156, Dec. 2012.
4. A. Torayev and T. Schultz, "Interactive classification of multi-shell diffusion mri with features from a dual-branch cnn autoencoder," 2020.

5. T. Li, S. Sun, M. Bolić, and J. M. Corchado, "Algorithm design for parallel implementation of the smc-phd filter," *Signal Processing*, vol. 119, pp. 115–127, 2016.
6. Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, "Reconfigurable manufacturing systems," *CIRP annals*, vol. 48, no. 2, pp. 527–540, 1999.
7. W. Lepuschitz, A. Zoitl, M. Vallée, and M. Merdan, "Toward self-reconfiguration of manufacturing systems using automation agents," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 41, no. 1, pp. 52–69, 2011.
8. H. U. Rehman, T. Pulikottil, L. A. Estrada-Jimenez, F. Mo, J. C. Chaplin, J. Barata, and S. Ratchev, "Cloud based decision making for multi-agent production systems," *Progress in Artificial Intelligence. EPIA 2021. Lecture Notes in Computer Science*, vol. 12981, no. 9, p. 673–86, 2021.
9. L. A. Estrada-Jimenez, T. Pulikottil, N. N. Hien, A. Torayev, H. U. Rehman, F. Mo, S. N. Hojjati, and J. Barata, "Integration of Cutting-Edge Interoperability Approaches in Cyber-Physical Production Systems and Industry 4.0," in *Design, Applications, and Maintenance of Cyber-Physical Systems*, P. Rea, E. Ottaviano, J. Machado, and K. Antosz, Eds. IGI Global, 2021, ch. 7, pp. 144–172. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-7998-6721-0.ch007
10. D. Schwung, J. N. Reimann, A. Schwung, and S. X. Ding, "Self Learning in Flexible Manufacturing Units: A Reinforcement Learning Approach," *9th International Conference on Intelligent Systems 2018: Theory, Research and Innovation in Applications, IS 2018 - Proceedings*, pp. 31–38, 2018.
11. M. El-Shamouty, X. Wu, S. Yang, M. Albus, and M. F. Huber, "Towards Safe Human-Robot Collaboration Using Deep Reinforcement Learning," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4899–4905, 2020.
12. B. I. Epureanu, X. Li, A. Nassehi, and Y. Koren, "Self-repair of smart manufacturing systems by deep reinforcement learning," *CIRP Annals*, vol. 69, no. 1, pp. 421–424, 2020.
13. B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019.
14. P. Tsarouchi, G. Michalos, S. Makris, T. Athanasatos, K. Dimoulas, and G. Chryssolouris, "On a human–robot workplace design and task allocation system," *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 12, pp. 1272–1279, dec 2017. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/0951192X.2017.1307524
15. N. Kousi, C. Gkournelos, S. Aivaliotis, K. Lotsaris, A. C. Bavelos, P. Baris, G. Michalos, and S. Makris, "Digital Twin for Designing and Reconfiguring Human–Robot Collaborative Assembly Lines," *Applied Sciences*, vol. 11, no. 10, p. 4620, may 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/10/4620
16. N. Kousi, C. Gkournelos, S. Aivaliotis, C. Giannoulis, G. Michalos, and S. Makris, "Digital twin for adaptation of robots' behavior in flexible robotic assembly lines," *Procedia Manufacturing*, vol. 28, pp. 121–126, 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2351978918313623
17. M. C. Magnanini and T. A. Tolio, "A model-based Digital Twin to support responsive manufacturing systems," *CIRP Annals*, vol. 70, no. 1, pp. 353–356, 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0007850621000676
18. C. Zhang, W. Xu, J. Liu, Z. Liu, Z. Zhou, and D. T. Pham, "Digital twin-enabled reconfigurable modeling for smart manufacturing systems," *International Journal of Computer Integrated Manufacturing*, vol. 34, no. 7-8, pp. 709–733, aug 2021. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/0951192X.2019.1699256
19. Q. Chen, B. Heydari, and M. Moghaddam, "Leveraging task modularity in reinforcement learning for adaptable industry 4.0 automation," *Journal of Mechanical Design, Transactions of the ASME*, vol. 143, no. 7, 2021.
20. S. Cavalieri and M. G. Salafia, "Asset administration shell for plc representation based on iec 61131–3," *IEEE Access*, vol. 8, pp. 142 606–142 621, 2020.
21. E. Carpanzano, A. Cesta, A. Orlandini, R. Rasconi, M. Suriano, A. Umbrico, and A. Valente, "Design and implementation of a distributed part-routing algorithm for reconfigurable transportation systems," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 12, pp. 1317–1334, 2016.
22. Y. Yamada, K. Ookoudo, and Y. Komura, "Layout optimization of manufacturing cells and allocation optimization of transport robots in reconfigurable manufacturing systems using particle swarm optimization," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 2. IEEE, 2003, pp. 2049–2054.
23. Y. Koren and M. Shpitalni, "Design of reconfigurable manufacturing systems," *Journal of manufacturing systems*, vol. 29, no. 4, pp. 130–141, 2010.
24. A. Li and N. Xie, "A robust scheduling for reconfigurable manufacturing system using petri nets and genetic algorithm," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 2. IEEE, 2006, pp. 7302–7306.
25. J.-M. Yu, H.-H. Doh, J.-S. Kim, Y.-J. Kwon, D.-H. Lee, and S.-H. Nam, "Input sequencing and scheduling for a reconfigurable manufacturing system with a limited number of fixtures," *The International Journal of Advanced Manufacturing Technology*, vol. 67, no. 1-4, pp. 157–169, 2013.