Journal of Manufacturing Systems 60 (2021) 22–34

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Towards online reinforced learning of assembly sequence planning with interactive guidance systems for industry 4.0 adaptive manufacturing

Andrea de Giorgio *,[1], Antonio Maffei, Mauro Onori, Lihui Wang

KTH Royal Institute of Technology, Department of Production Engineering, SE-11428, Stockholm, Sweden

ARTICLE INFO

Keywords:
Reinforcement learning
Adaptive assembly
Assembly sequence planning
Assembly guidance system
Manufacturing
Industry 4.0
Optimization
Knowledge retrieval

ABSTRACT

Literature shows that reinforcement learning (RL) and the well-known optimization algorithms derived from it have been applied to assembly sequence planning (ASP); however, the way this is done, as an offline process, ends up generating optimization methods that are not exploiting the full potential of RL. Today's assembly lines need to be adaptive to changes, resilient to errors and attentive to the operators' skills and needs. If all of these aspects need to evolve towards a new paradigm, called Industry 4.0, the way RL is applied to ASP needs to change as well: the RL phase has to be part of the assembly execution phase and be optimized with time and several repetitions of the process. This article presents an agile exploratory experiment in ASP to prove the effectiveness of RL techniques to execute ASP as an adaptive, online and experience-driven optimization process, directly at assembly time. The human-assembly interaction is modelled through the input-outputs of an assembly guidance system built as an assembly digital twin. Experimental assemblies are executed without pre-established assembly sequence plans and adapted to the operators' needs. The experiments show that precedence and transition matrices for an assembly can be generated from the statistical knowledge of several different assembly executions. When the frequency of a given subassembly reinforces its importance, statistical results obtained from the experiments prove that online RL applications are not only possible but also effective for learning, teaching, executing and improving assembly tasks at the same time. This article paves the way towards the application of online RL algorithms to ASP.

1. Introduction

Sometimes good things have to be broken in order to be rebuilt even better, a process referred as disruption. Assembly sequence planning (ASP) seems to require it because of the recent introduction of the paradigms of Industry 4.0 and smart manufacturing [1] that ask for manufacturing systems and, specifically, assembly lines to be adaptive to changes [2–5], flexible [6], evolvable [7,8], resilient to errors [9] and attentive to the more knowledgeable operators' skills and needs [10,11]. All these characteristics cannot be expressed by a static execution of a predetermined ASP that is produced offline, before that the assembly takes place in the real-life industrial environment. Relying on fixed instructions can certainly enforce the strength of an almost-optimal solution that neglects small but known deviations from the most common procedure; however, a self-adaptable execution based on a strong plan is in line with what required by Industry 4.0 [12]. As shown in this article, the latter can address those exceptions too, leading towards higher success rates in assembly.

Reinforcement learning (RL) is a machine learning method [13] that deviates from the idea of training learning algorithms only on all the available data, by accepting the possibility that the training could continue over the execution phase, when the algorithm is applied. The application of such an algorithm to an unforeseen case generates new learning data. The interaction with the changing environment makes these algorithms more resilient; however, it is often the case that for environments that are well-modelled, such as experiments in physics, the RL algorithm – be that an ant colony, a particle swarm, a genetic algorithm, etc. – is applied to the simulated environment to generate an optimal solution [14,15]. This is indeed a good method because both the simulation and the RL algorithm can be run hundreds or thousands of times per second. Even though RL adaptable systems have been successfully developed for other manufacturing purposes [16], the approach is not yet feasible in assembly, as simulating the overall complexity of an assembly line and the interaction with humans is still

* Corresponding author.
E-mail address: andrea@degiorgio.info (A. de Giorgio).
[1] URL: https://andreadegiorgio.com

https://doi.org/10.1016/j.jmsy.2021.05.001
Received 8 December 2020; Received in revised form 7 April 2021; Accepted 1 May 2021
0278-6125/© 2021 The Author(s). Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

an open challenge, i.e. having entire factories that are digital and simulated. Today, ASP simulations can only rely on fixed parameters such as data extracted from the assembly CAD models or relevant insights in the topic. An example are all the criteria-based optimizations present in literature [17–20]. Luckily, not all the good aspects introduced by RL are lost in manufacturing. This article makes use of the basic step-by-step learning paradigm of RL and a statistical approach to prove that RL algorithms can be successfully applied in online ASP, at assembly time, when the number of assembly executions is limited to the possibilities of the real environment. The experiment is planned, developed and executed as an agile process, with incremental hypotheses. The experimental setup is built around a manufacturing course Tillverkningsteknik (MG1026) at KTH Royal Institute of Technology in which circa 150 students assemble a metal locomotive toy.

The approach of this article that introduces ASP variations at assembly time is in accordance with Marton's phenomenographic theory [21] that affirms that "there is no learning without discernment, and there is no discernment without variation". Both the assembly operators and the assembly environment introduce a great deal of variation that needs to be captured and exploited at the right time, i.e. during assembly, to generate better assembly sequences. Furthermore, the assembly system presented enforces the human-centered view of the operator 4.0 [22] for improvements in their physical ergonomics and the creation of useful mental models of the assembly for the operator, when these are not provided by the assembly design [23].

Among the preliminary results of the agile experiment, it emerges the need to digitalize the assembly-operator interaction at assembly time, which is solved by introducing digital twins of the operator's cognitive process and the assembly [24]. The gamification of the user experience has also proven a successful technique in manufacturing [25]. Thus, it can be integrated into an assembly guidance system (AGS). It follows that the article presents an agile-developed version of such AGS and the data collected over trial and error attempts to generate ASP online with it.

The aim of this article is to pave the way towards the introduction of online RL statistical techniques to model the ASP at runtime, instead of the common offline use of RL techniques for creating pre-determined and fixed ASP to run at assembly time.

This article is structured in the following way. Section 2 contains additional literature review on the topics touched by this introduction. Section 3 explains the research methodology applied to find innovative solutions to an old problem, i.e. ASP. Section 4 is the core of the paper and it is divided into several subsections corresponding to the various phases of the agile experiment and their partial results. Section 5 presents and discusses the overall results in applying RL for ASP. Section 6 presents the conclusions and outlines some future work.

2. Related works

Claeys et al. [26] introduce a generic model for managing context aware assembly instructions, i.e. the assembly instructions are pre-generated and stored to be retrieved when most useful, depending on the assembly environment. Their system, however, does not present learning capabilities at assembly execution time. In general, there are algorithms for solving and optimizing ASP problems based on known variations [17–20,27,28], computer aided geometric feasibility and optimization ASP algorithms [29–32]. None of these considers changing the assembly sequence at assembly time.

A literature review from Rashid et al. [33] reports several articles applying soft computing methods for ASP, including many that belong to the RL class, such as twenty-two using genetic algorithms, three using ant colony optimization, five using particle swarm optimization. There have been a few attempts to determine an optimal assembly sequence using reinforcement learning [34] and at least an attempt using deep reinforcement learning from Zhao et al. [35]; however, these methods focus on reinforcing policies, i.e. converging towards an optimal solution, on assembly states and conditions that are not directly acquired in an industrial environment but generated from a knowledge base. This approach has been flagged as partially wrong by Kaelbling et al. [36] for two main reasons: Firstly, because mapping an environment in advance, e.g. with a knowledge base, requires a huge effort than compared to acquiring data while operating in the environment. Secondly, because the environment is often subject to changes that can be better handled by an adaptable system that learns during its execution. Thus, the ability of an assembly planning system to learn during the assembly execution based on human decisions and real-time issues is fundamental. A work from Watanabe & Inada [37] seems to go in this direction, though it focuses on acquiring historical performance data from a robotic assembly and use reinforcement learning to improve the assembly task. As a confirmation that reinforcement learning is much more applied in robotics than in manufacturing problems, further joint robotic-assembly works are hereby reported: Yu et al. [38] proposed a case study using reinforcement learning to solve the scheduling problem in a human-robot collaborative assembly task. Martinez et al. [39] focused on reinforcement learning of robotic manipulations as part of an assembly task. All the reviewed scientific literature seems to neglect the possibility of applying reinforcement learning directly to human behavior during assembly tasks. An operation that holds the potential to elicit dynamic environmental knowledge and personal knowledge from the operators.

It is fair to say that a standard aspect left untouched by this article is the use of liaison matrices to represent assemblies [40–42]. The RL approach hereby presented is based on the aforementioned mathematical tool that has become common practice for computer-based assembly representation; however, an innovation comes from using the liaison matrix as a mathematical base for the RL statistical algorithms.

3. Research goal and methodology

The aim of this research is to introduce the use of reinforcement learning (RL) to produce an optimal assembly sequence plan during assembly execution. RL is a class of powerful machine learning (ML) algorithms able to learn during the execution of a process. They represent an alternative to the traditional learning methods where, firstly, a process is executed and its data collected, secondly, the process data is learned by an ML algorithm. The traditional ML methods are in line with and used by the common assembly sequence planning (ASP) strategies adopted by industry. Usually, an engineer plans the optimal ASP by studying the product features with some support software, often based on ML algorithms, and then the optimal ASP is implemented in form of assembly instruction manuals or any kind of non-adaptive guidance systems for the assembly execution. This approach (see Fig. 1) requires the assembly operators to give feedback only when it is too late to apply changes to the established ASP and a major change requires the original engineer to run the whole optimization process again with the new parameters acquired from the feedback, often after the review of the assembly design. Several operations that require time and effort, other than a distributed workload over ASP engineers, operators and other parties involved in the product design.

An ideal RL strategy could introduce an additional and faster feedback cycle (see Fig. 2) between the ASP generation and execution. The ability of RL algorithms to find optimal solutions while maintaining a degree of adaptability to new scenarios is exactly what can enable a new framework for ASP and become the ultimate goal of this research; however, there is no straightforward way to achieve this, as such a goal

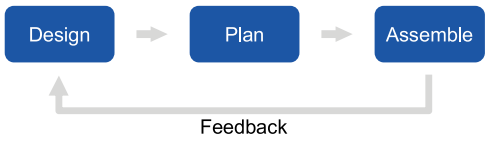

Fig. 1. Traditional feedback cycle in ASP.

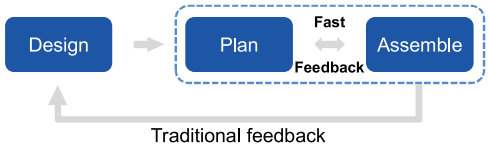

Fig. 2. Traditional and fast feedback cycles, when the ASP is generated at assembly time with RL.

requires to assess the effects of changing different aspects of traditional ASP and RL methods. RL algorithms are based on the possibility to simulate large number of executions in modelled environments, characteristic that is missing in the ASP process. The number of executions is limited to those happening in the real factory and the RL method needs to converge faster than its computational counterparts do. Note that the RL feedback cycle is faster because it can be executed at each assembly step. The traditional feedback flow is longer because it requires the entire assembly to be ended before that any data can be fed back to the assembly designer or to the assembly planner.

There are five aspects of traditional RL that have to be mapped to ASP: environment, agent, states, rewards and actions (see Fig. 3). The environment is clearly the assembly process. One would rather think of the assembly station as a production environment, but be aware that this computer science terminology refers to the process that is modelled as an ML algorithm rather than a real environment where the process is executed. The agent is somehow a bit more complex to define. An agent can be both the operator executing the assembly and the assembly guidance system taking a certain decision for the operator. States and actions are connected to how an assembly plan can be represented in form of computational knowledge. Finally, rewards are connected to the optimization strategy and are of two kinds: the feelings perceived by operators in satisfactory assembly steps and the numerical score attributed to the execution of certain actions in answer to certain states. The former reward is perhaps the most complex aspect to model, especially when the optimality of a step becomes subjective because of the choice of the particular operator executing the assembly. A more objective way to consider the rewards is by looking at aggregated choices from different operators.

As mentioned earlier, there is a duality of the agent in the real world as human operator and as its digital twin embedded in the AGS. A human is digitally represented by the inputs and outputs of a set of sensors and interfaces enabling for a direct translation of the human perception, intention and actions to the simulated environment where RL is enforced. This constitutes a central point for this research. Namely, exploring how assembly knowledge is transferred from an AGS to an operator and *vice versa*.

The overall system is presented in Fig. 4. An operator O_i produces an assembly plan P_i during the assembly Ai. The current plan and the past ones P1, …, Pi are used for online RL of the AGS instructions to the next operators.

All the aspects of this system, previously described, become sub-goals for this research. Therefore, an agile approach is required to explore and meet as many sub-goals as possible and pave the way towards the ultimate goal that is using RL in ASP. A series of experiments are planned and executed one at a time before knowing what the next one will be. Each experiment acts as an observation environment, useful to produce new research hypotheses and test them within the next experiment. This adaptive methodology has an advantage when the intermediate goals are clear, but there is no clear understanding about the overall process to investigate and there does not exist an established experiment to directly achieve and test the ultimate goal.

Fig. 3. Traditional RL architecture.

Each experiment is carried out with the following cycle of operations (see Fig. 5):

- analysis of results from prior experiments and next research hypotheses generation;
- setup and test of new experimental equipment;
- assembly under experimental conditions;
- collection of results.

Each experiment cycle is performed several times. The agile principle allows testing an experimental setup for errors and possible issues, other than collecting significant statistical data before proceeding to the next experiment; however, for clarity, results presented in this paper do not mention how many times a cycle is executed prior to the obtainment of definitive results. Issues encountered in some of the test runs might be presented as general results of one particular experiment and omitted in the others. This does not indicate a discontinuity of issues, but rather a shift in the focus of the experiments.

4. Experimental setups and partial results

The experimental setup consists of an assembly station for a metal toy locomotive, as shown in Fig. 6. The assembly station is minimal, as it is part of a university course and not an industrial line. There is a wide table, with assembly tools and components. A group of students, further referred as novice operators, performs the assembly task. The peculiarity of this assembly station is that the product is stably invariant, while the operators change at every experiment. The academic course Tillverkningsteknik (MG1026) at KTH Royal Institute of Technology is structured so that circa 150 students take part every semester to several applied tasks. One of these involves circa 50 locomotive assemblies, done in groups of one to four students. The same students who perform the final assembly produce the locomotive components during the course; however, the course objective is teaching how to manufacture the parts, rather than assembling them. Thus, altering the assembly does not alter the learning outcomes of the course and this allows the experimental setups to be independent from any educational needs.

Each experiment is, as much as possible for this article, reported as a standalone execution. This is explained in the scientific methodology section; however, the overall scientific progress is part of a unique experimental setup that evolves in an agile way towards the interesting findings. Thus, sometimes the lines are a bit blurred and some common details are only explained in the section corresponding to the experiment where they are mainly relevant.

4.1. Setup of the first experiment

The first experiment consists of placing a camera over the assembly station where the metal toy locomotive is assembled. The novice operators follow the assembly instructions provided by three paper sheets present on the table. The instructions consist of an exploded view of the assembly, a list of screws and their manufacturing data and labels associated with a certain CAD component in the drawings and an operation list that goes as follows:

- Assemble the boiler and front cover.
- Mount the boiler on the frame. Use dome nut and chimney as nuts. Do not overtighten.
- Insert the roof into the cabin. The roof is a black plastic plug.
- Mount the cabin to the boiler.
- Mount a wheel on each axle. Push the shoulders into the frame.
- Fit the remaining wheels.

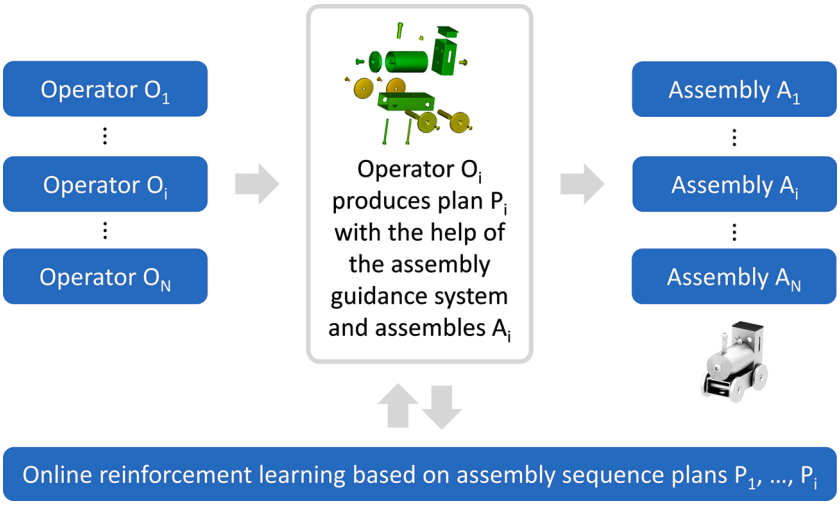

Fig. 4. System overview.

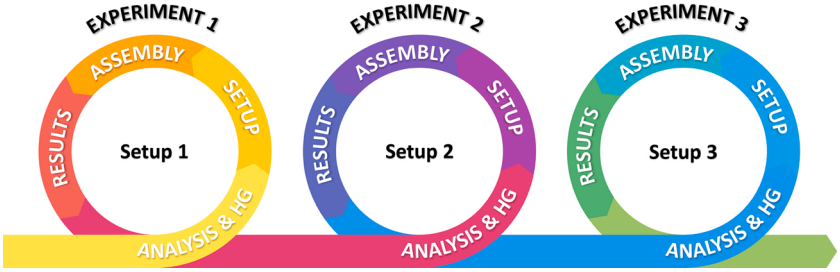

Fig. 5. Agile approach to experiments. Each experiment consists of a set of steps, from analysis of previous results and research hypotheses generation (HG), to the preparation of a next experimental setup, assembly and collection of results.

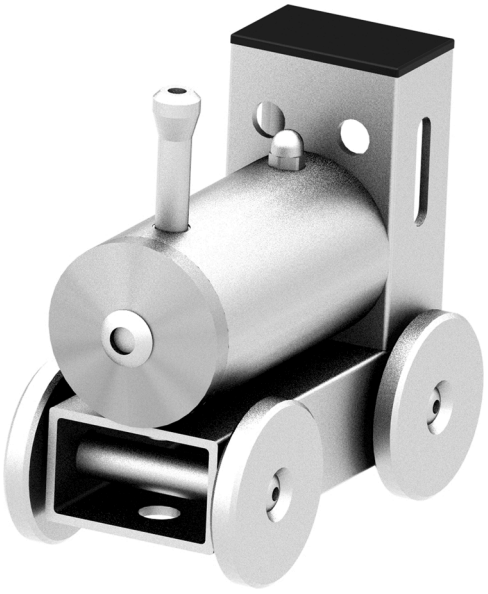

Fig. 6. A rendering of the locomotive from its CAD parts.

In this experiment, the observing researcher does not interact at all with the operators. Nine videos are thus recorded and analyzed.

4.2. Results of the first experiment

This setup consists of the original course assembly task and it is always successfully completed by the students/operators, with the optional assistance of the course instructors.

Nine assembly videos, recorded with this setup, are watched several times to qualitatively and quantitatively grasp several aspects of the assembly scenario, as seen from different operators. This is in line with Marton's phenomenographic approach [21]. Among the observations, a few issues that are relevant to the development of a human-machine interaction system are identified.

Issue 1. Parallel assembly. Having 2–5 operators per assembly station means that at some point a bit of parallel assembly is inevitable. Further experiments need to handle this issue in order to capture a proper assembly sequence.

Issue 2. Product quality. A few times during an assembly task, a subassembly operation is suspended because a component has not been properly machined and needs some further adjustments. The total assembly time should exclude eventual delays due to quality issues. If quality issues arise, there are no other manufactured components that can be used to replace the originals. Thus, a quality check needs to be run prior to assembly.

Issue 3. Camera perspective. About half the videos are recorded from a side perspective and another half are recorded from a top perspective. Both camera positions present advantages. A side perspective works best for showing the action from a human perspective. A top view is better to avoid occlusions due to objects or humans in the scene. Ideally, a camera can be positioned 45 degrees towards the table, halfway between top and side view.

Issue 4. Operators' intentions. Verbal communication is used to exchange intentions among operators while performing the assembly; however, it is hard to define when a certain intention arises, before it is communicated to the other operators and/or it is concretized into an assembly action.

4.3. Hypotheses generation and setup of the second experiment

In order to create more variability, according to Marton's proposal [21], in the second experiment the operators are asked to perform the assembly with the sole exploded view, without reading the written instructions. Course assistants are asked to refrain from helping the students and assistance is only provided in case of major problems, such as quality issues.

The research hypotheses for this experiment are the following:

Hypothesis 1. When lacking the assembly instructions, the operators have to apply their own understanding of the assembly task, consisting of limited knowledge due to prior personal experience or education.

Hypothesis 2. The assembly process can be successfully completed without the assembly instructions.

The experiment should be useful to learn what kind of knowledge an AGS needs to provide in order to obtain a successful assembly of the product.

4.4. Results of the second experiment

The results of this experiment are of a qualitative nature and based on a number of experiments that can give intuitive answers to the two hypotheses made. About ten assemblies are executed without providing the written instructions to the operators, but the sole exploded view and screws/nuts tables. All the operators indeed tried to perform the assembly, even with missing instructions. In a few cases, the intervention of the researcher has prevented the sub assembly of parts that would have jeopardized the completion of the assembly. Thus hypothesis 1 (H1) is confirmed but hypothesis 2 (H2) is false because at least one exception was found, i.e. an assembly was not properly completed without instructions. The operators would use their knowledge in spite of the missing written instructions, as in H1, but the completion of the assembly relies on the ability of the observing researcher to steer the operators away from the wrong assembly sequences once the intention of the operators is manifested. This last point is a key issue for the generation of new hypotheses and their testing.

It follows, from a positive H1 and a negative H2, that an AGS is needed. The guidance system has to prevent the operators from doing something wrong while allowing them to exploit their personal knowledge of the assembly. Ideally, for optimal communication, such a system is only required to provide instructions based on the manifested intentions of the operators. Two engineering techniques are introduced to the experimental setup and used to the develop a guidance system for this purpose: liaison matrices and soft constraints. Liaison matrices are a way to mathematically encode if two assembly components are to be assembled together. In particular, an adjacency matrix lists all the neighbor components and the liaison matrix does that; but it also excludes components that do not have a stable assembly operation between them. The limitation to this matrix is that it is not possible to define if more than two liaison components are part of the same subassembly or separate ones.

In order to encode the subassembly order, there are two main approaches. One considers the assembly steps and defines which components belong to each step. The other is based on precedence constraints. For each component to be assembled a precedence matrix indicates which other components must have been assembled before. An operator start assembling the components that have no requirements and proceeds with those that are allowed by the assembled components, until the assembly is finished. These two approaches are supported by several ASP methods that rely more on one or the other. For example, AND/OR graphs or any winning assembly sequences from ASP based on genetic algorithms show all the alternatives available for each step, while the assembly precedence graphs constrain the sequences to those that can satisfy the precedence constraints. In any case, the aim of ASP is to generate as many sequences that are feasible and select the best sequence for the final assembly and relative instructions. For a guidance system to allow an operator to freely select the next assembly step, while checking that such step is not preventing the execution of the whole assembly, the method that provides the more adaptable solution is an AND/OR graph which lists all the possible solutions; however, an AND/OR graph is not easy to produce for every assembly and embed into matrices for automatic execution in an AGS. Methods that come up with few assembly plans are not generic enough to evaluate assemblies that are freely defined by an operator. Instead, precedence constraints are widely used. Because precedence matrices are easy to define and deploy for many assemblies. The only issue with precedence constraints is that they do not leave much choice to the operators unless they can be violated. Thus, soft constraints are preferable for the next experimental setups, i.e. some precedence constraints that are not mandatory. The starting point is testing the ability of the operators to complete the assembly without introducing any soft or hard constraints.

An AGS – in its version 1 (v1) - is developed as a touch screen interface that shows the exploded locomotive assembly and allows an operator to pick two by two components that have a liaison, i.e. a value of one between them in the liaison matrix (see Figs. 7 and 8). The resulting interface is presented in Fig. 9. The soft constraints are imposed visually, in form of a green/gray map (see Fig. 10) that shows components that can be picked up. The exploded view blinks with the green/gray map to show that components are selectable. Once a component is selected this becomes blue. Two selected components are automatically confirmed as assembled in real time. An "undo" button appears to cancel the latest operation. See Fig. 9 for more details.

The AGS records all the assembly operations in a. mat file containing the assembly transitions or actions (from previous component to next component) specified over the liaison matrix as increasing values from 1 to N. This file is easily importable in Matlab®, where all the results are collected and analyzed. At any time, the. mat file contains all the information from the previous successful assemblies and it is reloaded and updated by the AGS for each new successful assembly. Thus, each assembly is guided by the partial results obtained by all the previous assemblies plus the values for the current one. This enables a reinforcement learning approach to detect and enforce soft constraints. It becomes the objective of the following experiments to record through the AGS and statistically analyze in Matlab® the assembly behavior of the operators.

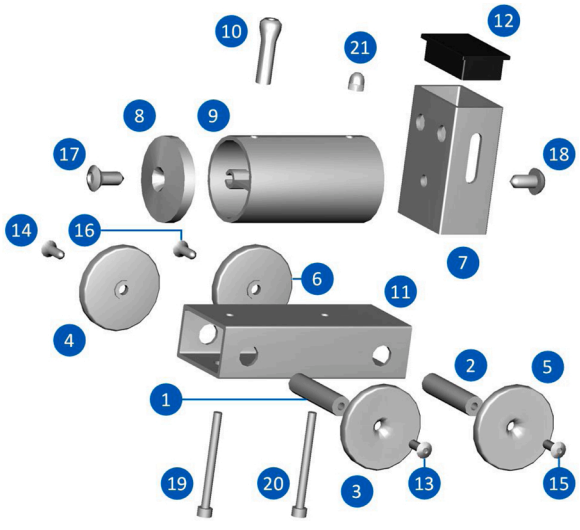

Fig. 7. Component IDs for the locomotive assembly.

Component	Axel A	Axel B	Wheel A	Wheel B	Wheel C	Wheel D	Cabin	Nose cone	Boiler	Cimney	Bottom frame	Roof	Philips screw (wheel A)	Philips screw (wheel B)	Philips screw (wheel C)	Philips screw (wheel D)	Threaded screw (forehead)	Threaded screw (cabin side)	Hexagonal screw A	Hexagonal screw B	Dome nut
ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1			1	1							1		1	1							
2					1	1					1				1	1					
3	1												1								
4	1													1							
5		1													1						
6		1														1					
7									1			1						1			
8									1								1				
9							1	1		1	1						1	1	1	1	1
10									1										1		
11	1	1							1										1	1	
12							1														
13	1		1																		
14	1			1																	
15		1			1																
16		1				1															
17								1	1												
18							1		1												
19									1	1	1										
20									1		1										1
21									1											1	

Fig. 8. Liaison matrix for the metal locomotive assembly.

4.5. Hypotheses generation and setup of the third experiment

Given the results of the second experiment and the development of the AGS v1, hypotheses 3, 4 and 5 (H3, H4 and H5) are made:

Hypothesis 3. The AGS provides the needed assembly guidance to complete the assembly when the assembly instructions are missing.

Hypothesis 4. The AGS solves the parallel assembly issue 1 by forcing the assembler to plan and execute operations serially.

Hypothesis 5. The AGS solves issue 4 by framing an operator's intention before they can execute it.

The AGS v1 is deployed (see Fig. 9) for tests over another round of assemblies. At this stage, the hypotheses to be tested are H3, H4 and H5, all of them about the AGS. Thus, the setup is the same as in the second experiment, but the researcher does not interact with the operators for other reasons than to explain how to use the AGS itself or when fatal assembly operations are done.

4.6. Results of the third experiment

As said before, several assemblies are executed without providing the written instructions to the operators, but the sole exploded view and screws/nuts tables. The results of this experiment verify H3, because all the operators in 20 trials understood how to successfully complete the assembly, though in few executions they still required a little external help to spot the existence of some constraints. The use of an AGS also confirms the validity of H4, because no parallel assembly attempts arose when the operators had to stick to selection and execution steps with the AGS. It becomes rather difficult to understand if H5 can be confirmed or not, as the researcher's observations captured another related issue preventing the demonstration of H5. Even if the AGS can elicit the intention of the operators, not all the operators are comfortable with the idea of selecting components to be assembled in two by two selections. In almost all cases, the explanation of the researcher that the system only takes two components at a time was not accepted or accepted with unfavorable comments. The problem found is formulated as:

Issue 5. Subassemblies. Every operator expresses their assembly intention in form of new subassemblies made of several components, rather than adding one new component to the current assembly.

4.7. Setup of the fourth experiment

The results from the third experiments lead to issue 5 that can be addressed by adjusting the AGS structure in a way that reflects multi-component subassembly selections. Thus, a new AGS version 2 (v2) is developed.

In this second version of the AGS, some soft constraints are

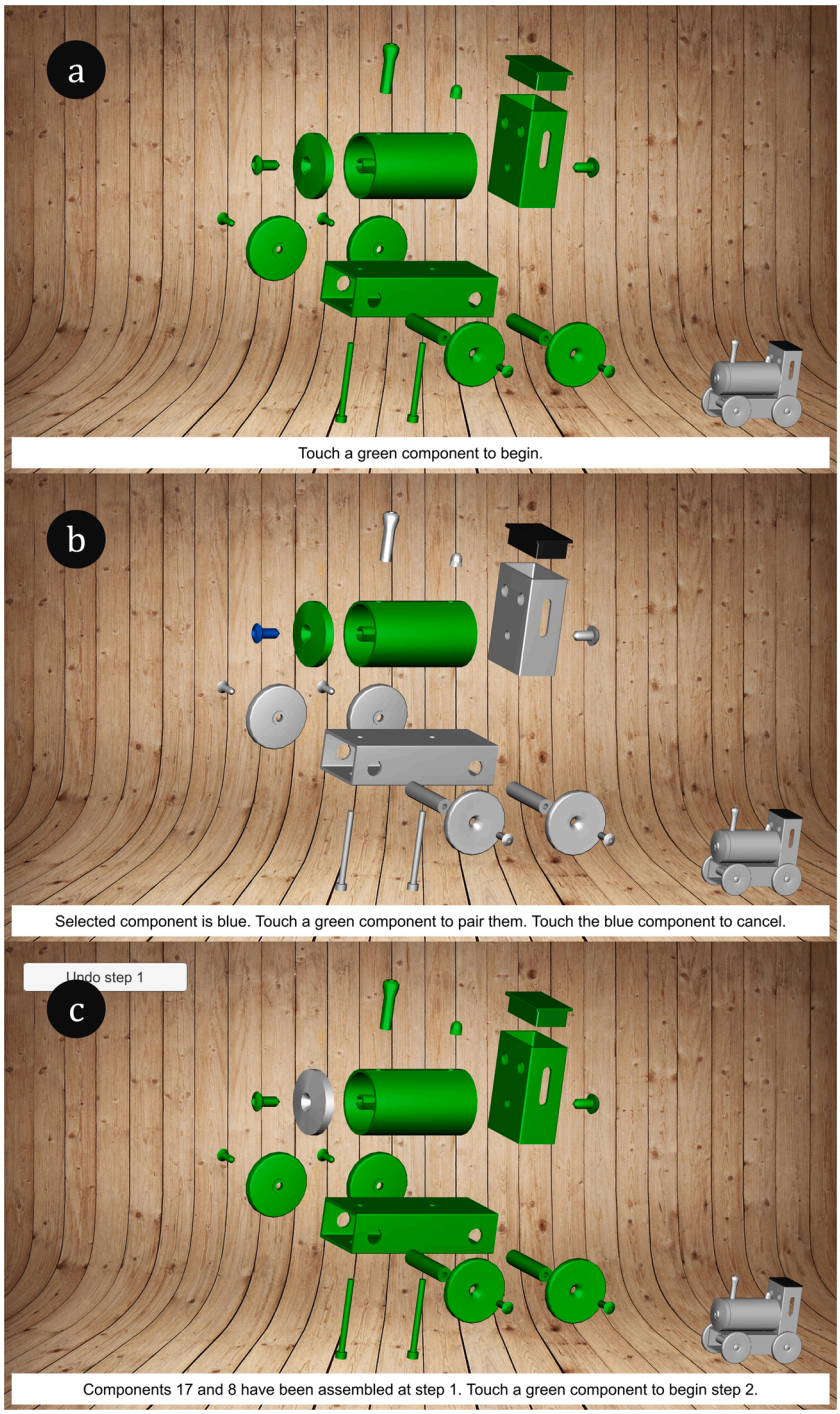

Fig. 9. Assembly guidance system v1 with digital twin of the locomotive assembly. (a) The green/gray map allows selecting all the components that have available liaisons. (b) Component 17 "threaded screw" is selected. (c) Component 8 "front cover" is selected and the subassembly {8,17} is automatically considered done. Components 8 has become gray in the green/gray map because there are no other liaisons left for it, i.e. they cannot be further assembled. While component 17 still has one liaison left with component 9 "boiler". On the top-left corner, a button allows to undo the last assembly step. On the bottom-right corner, the assembled locomotive is shown. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

introduced. Initially based on the ASP presented in the paper assembly instructions, the constraints are imposed visually in form of a colored map (see Fig. 10 for the colormap and Fig. 11 for the AGS v2 interface) and updated after the execution of each assembly. This map shows components that are to be picked up later in time as more yellow than components that are to be picked up sooner in time, which are greener. The component colors are updated by recording the most frequent order of the selected components. Thus, they encode an ASP sequence in a less explicit – or fuzzy – language, using a simple action or transition frequency matrix.

The transition frequency matrix is based on the liaison matrix and it is expressed component by component with a range of increasing values from one to N over the liaisons used. Liaisons are set from the beginning to values -1. The matrix is, of course, symmetrical.

While the colored ap does not solve a particular issue, it has the aim of reinforcing the idea that a standard ASP exists before that an operator can analyze the assembly state and make their decision. It is a method to elicit knowledge from the human operators while they are using the AGS.

On the AGS interface, at first, the entire colored map blinks. After the selection of a first component that becomes blue, all the liaison-related component blink with the color map, while the rest of non-liaison-related components are gray and unselectable. On the bottom-left corner, an assembled locomotive rotates to give a preview of the final objective of the assembly process. On the top-left corner, a button allows to undo the last assembly operation. On the top-right corner, the selected

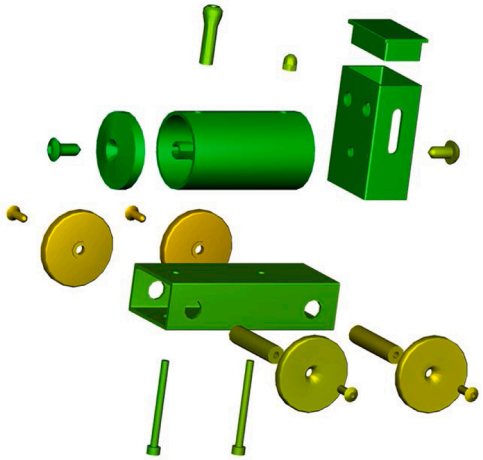

Fig. 10. Colored map for the locomotive assembly. Green components should be assembled earlier than yellow components. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

subassembly is shown. The button "assemble components" confirms the subassembly operation done at the real station. When this is pressed, the selected assembly is moved to the bottom-right corner, where all the assembled components are shown as a preview of the current assembly state. A white ribbon on the bottom of the AGS interface describes the current state and the possible operations. See Fig. 11 for more details.

The aim of the subsequent experiment is to test that the AGS v2 does not generate issues that are similar to issue 5. The experimental setup is kept as before, with the sole change of the guidance system deployed in its version 2 and the issue 5 test is formulated as a new hypothesis 6 (H6):

Hypothesis 6. If every operator can express their assembly intention in form of new subassemblies made of several components, rather than adding one new component to the current assembly, issue 5 disappears.

4.8. Results of the fourth experiment

This round of assemblies verifies H6 because it quantitatively shows that operators can think in a quite wide range of subassemblies (see Table 1) and subsequent assembly sequences. It also shows that when the optimization criteria are not made explicit, people tend to follow any of their own ideas. Which can be formulated as:

Issue 6. Optimization criteria. Operators with soft constraints and no instructions tend to favor their own personal optimization criteria, which are not explicit.

This experiment confirms that, despite H6 stands, H5 is false. H5 was hard to prove or falsify with the previous experimental data, but this time it is falsified by the fact that there are no limitations to the intentions that an operator could have. Thus, the AGS will always be limited by the programmer's understanding of the operators' way of thinking.

4.9. Hypotheses generation and setup of the fifth experiment

All the previous experiments have created a basis of hypotheses and issues that leads to a final experiment. This experimental setup makes again use of the successful AGS v2. The researcher illustrates its use by explicitly asking the operators to pick subassemblies that would be stable after the assembly operation (see Table 1), i.e. when the components will hold together by gravity, friction or anything else than the operators' ability to keep them together. This setup aims at verifying the following:

Hypothesis 7. If optimization criteria are given, the choice of subassemblies converges towards specific choices.

In particular, this experiment should quantitatively verify hypothesis 7 (H7) in terms of stability of the chosen subassemblies. It should also validate all the previously proposed solutions to tackle the issues found. For this purpose, a whole class of circa 150 students is dedicated to one final large experiment, able to generate a statistically relevant number of assembly executions with the proposed AGS in its final version v2 and stability criteria.

4.10. Results of the fifth experiment

An entire class of students corresponds to 47 assembly groups and relative locomotive assemblies. For each assembly, the selected subassemblies and subassembly sequences are shown in Fig. 12. If these results are compared with those of the previous experiment, see Table 1, the choice of any subassemblies this time is limited to stable ones, thus numerically confirming H7.

The statistical results offer insights on what are the most common subassemblies in both the fourth (limited to the stable ones) and fifth experiments, namely: {8,9,17}, {7,12}, {7,9,18}, {1,3,13}, {2,5,15}, {1,4,11,14}, {2,6,11,16}, {9,11,20,21} and {9,10,11,19}. The observation of such common subassemblies in different order from Fig. 12 suggests that statistics of what subassembly at which step can also be extracted from the data and highlighted. The operation leads to Fig. 13, a statistical assembly step graph. Which is also a novel contribution to literature introduced by this article. This graph encodes the most significant subassembly/step choices to complete the assembly and it represents a hybrid form between a general AND/OR graph and a fully defined assembly sequence.

The order of selection of all the subassemblies is analyzed in Matlab® and it generates two relevant results shown in Figs. 14 and 15. The first one is a statistically reinforced subassembly transition matrix (Fig. 14) that is composed of all the values corresponding to a transition from one subassembly (column) to another (row). The diagonal shows the total count of each subassembly and it is in accordance with the gray boxes displaying the same information in Fig. 13. In orange and red, the matrix values that are respectively above two-third and one-third of the diagonal numbers for the row, used as conventional thresholds to highlight the information contained. A matrix similar to this, but listing each component instead of the subassemblies is the one used to generate the colormap of Fig. 10. This suggests how the same operation could be done by visually letting the operator chose an entire sequence of subassemblies. The second result is a statistically reinforced subassembly precedence matrix (Fig. 15) that is generated by collecting all the values corresponding to a transition from any subassembly (column) to another (row), at any step. In other words, the past use of a subassembly is the value displayed by the column for each selected subassembly (row). If the column value is zero it means that the corresponding subassembly has never been assembled before the one indicated by the row. If the same conventional threshold as before is set for this matrix, considering two thirds of the diagonal number as a qualifying value, the cells above it are colored in red. They constitute precedence constraints that can be enforced to obtain an optimal ASP from the collective operators' knowledge elicited by the AGS with a statistical reinforcement learning process.

It is important to outline that the precedence/transition matrices in Figs. 14 and 15 did not numerically drive the assembly process in this research; however, they can be used to generate further colormaps that implicitly leave the choice to the operators, i.e. by showing colors instead of numbers. A similar operation was done on the statistical transitions applied over the liaison matrix to generate the AGS colormap shown in this article. Alternatively, in the application of online RL algorithms, these thresholds can be tuned up as hyperparameters for the RL algorithms to make an informed choice instead of leaving it up to the

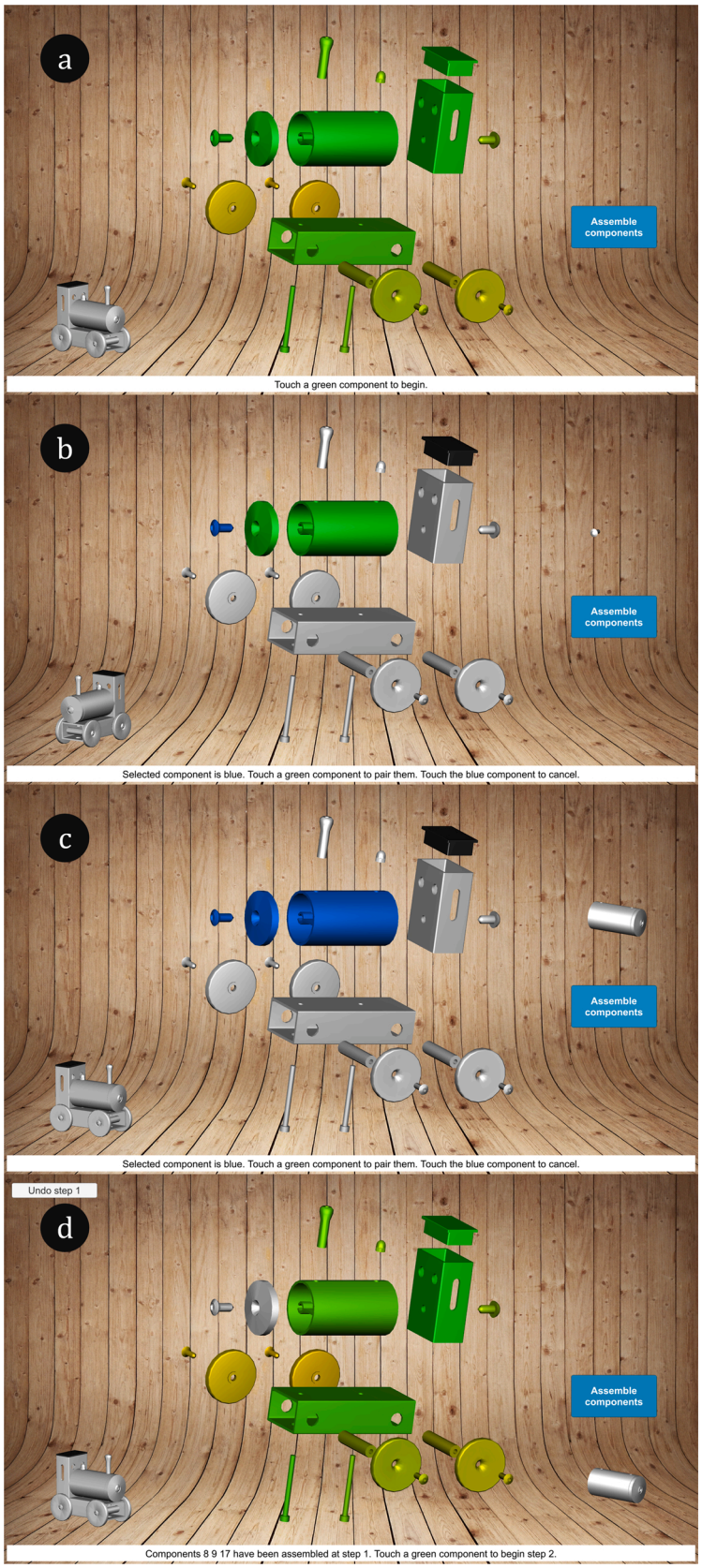

Fig. 11. Assembly guidance system v2 with digital twin of the locomotive assembly. On the bottom-left corner, an assembled locomotive rotates to give a preview of the final objective. On the top-left corner, a button allows to undo the last assembly operation. On the top-right corner, the selected subassembly is shown. On the bottom-right corner, all the assembled components are shown as a preview of the current assembly state. The button "assemble components" confirms the subassembly operation done at the real station. (a) The colored map allows selecting all the components. (b) Component 17 "threaded screw" is selected. (c) Subassembly {8,9,17} is selected. (d) Subassembly {8,9,17} is assembled and components 8 and 17 have become gray in the colored map because they cannot be further assembled.

Table 1
Frequency and stability of subassemblies selected with the assembly guidance system v2 in the fourth experiment (without the stability criteria) and in the fifth experiment (with the stability criteria).

Subassembly	Freq. 4th exp.	Freq. 5th exp.	Stability
{8,9,17}	15	47	Stable
{7,12}	15	47	Stable
{7,9,18}	14	47	Stable
{1,3,13}	13	35	Stable
{2,5,15}	12	35	Stable
{1,4,11,14}	9	33	Stable
{2,6,11,16}	8	33	Stable
{10,19}	8	0	Unstable
{9,11,20,21}	8	47	Stable
{9,10,11,19}	7	47	Stable
{11,19,20}	6	0	Unstable
{2,6,16}	6	3	Stable
{2,11}	5	0	Unstable
{9,19}	5	0	Unstable
{20,21}	5	0	Unstable
{1,11}	4	0	Unstable
{1,4,14}	4	4	Stable
{9,19,20}	3	0	Unstable
{5,15}	2	0	Unstable
{9,20}	2	0	Unstable
{9,20,21}	2	0	Unstable
{11,19}	2	0	Unstable
{1,2,11}	1	4	Stable
{1,3,11,13}	0	2	Stable
{1,3,4,11,13,14}	1	8	Stable
{1,3,4,13,14}	0	2	Stable
{2,5,11,15}	1	1	Stable
{2,5,6,11,15,16}	0	9	Stable
{2,5,6,15,16}	0	2	Stable

operators.

5. General discussion of results

Advancing from the first to the fifth agile experiments, the given paper instructions, illustrating a predefined fixed ASP, have been replaced by an adaptive AGS that learns an optimal ASP from the operators. By comparing the ASP before and after, it can be seen from Table 2 that the instruction order is slightly changed. This is due to the understanding level of the operators and the proper codification of their true intentions provided by the AGS interface. For instance, separate instructions such as "Push the shoulders into the frame", relative to subassembly {1,2,11}, and "Fit the remaining wheels", relative to subassemblies {1,4,14} and {2,6,16}, is replaced by a unique operation described by subassemblies {1,4,11,14} and {2,6,11,16}. This is because fitting the axel into the frame comes more natural for the operator when the operation is directly completed with the addition of the remaining wheel to it. In both cases, namely with paper instructions or AGS, the assembly is successful. Thus, the software approach objectively allows to structure and document the ASP process, together with the mental process of the operators, without interfering with them.

Among the many issues encountered, all were solved either immediately or by a following experiment. In particular, issues 1–4 are tackled at the beginning, providing the fundamental choices leading to the AGS, and issues 5 and 6 allow improving the AGS from its version 1 to version 2. The working hypotheses generated by this process are all qualitatively or quantitatively verified or falsified and overall show that controlling the experimental design and its variables in such a high complex assembly task is not only possible, but also fruitful. The statistical results collected at the end of the fifth experiment provide a statistical basis to apply RL algorithms at assembly time, basing the optimization function not on preexisting criteria but on the informed decisions of knowledgeable operators in the era of Industry 4.0. The statistical step frequency graph (see Fig. 13) and the statistical hard and soft precedence constraints (see Figs. 14 and 15) generated by this work

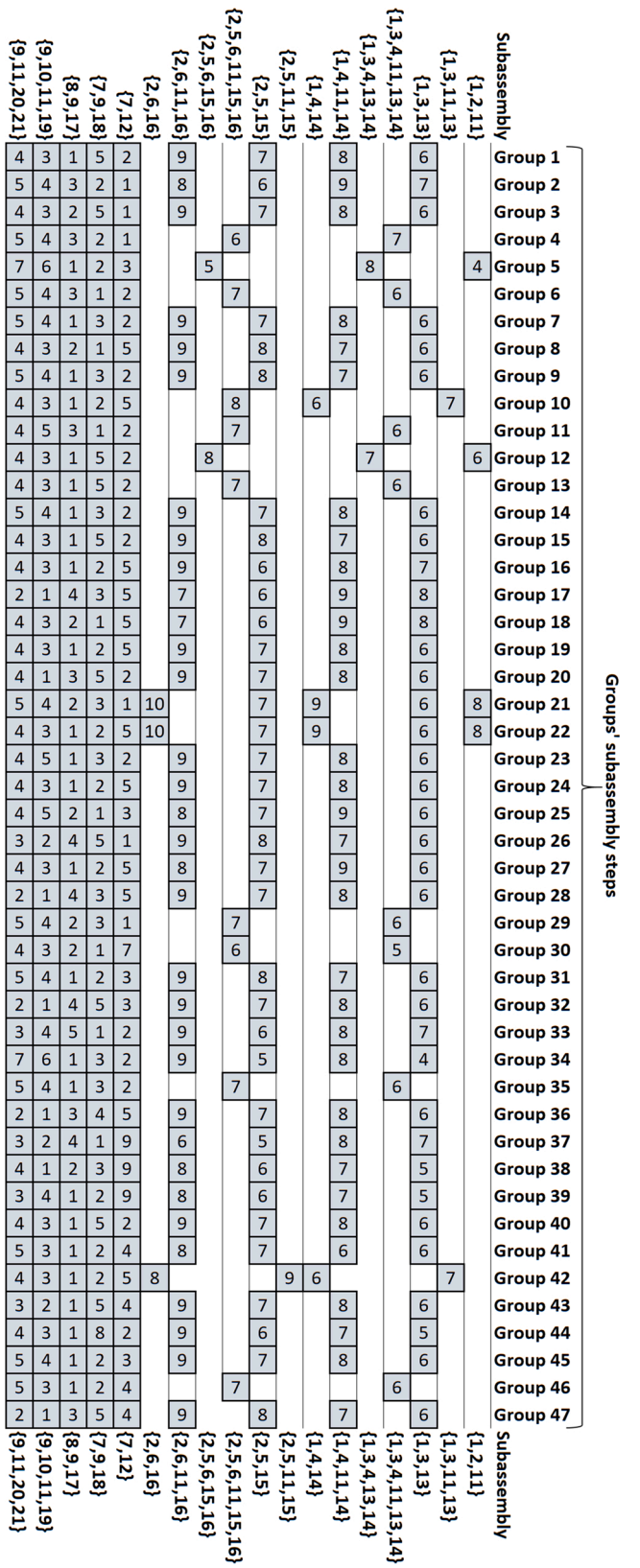

Fig. 12. Subassembly steps for each group.

are in line with the outcomes of previous techniques, with the sole difference that they are dynamically generated at assembly execution time, as online RL methods would require.

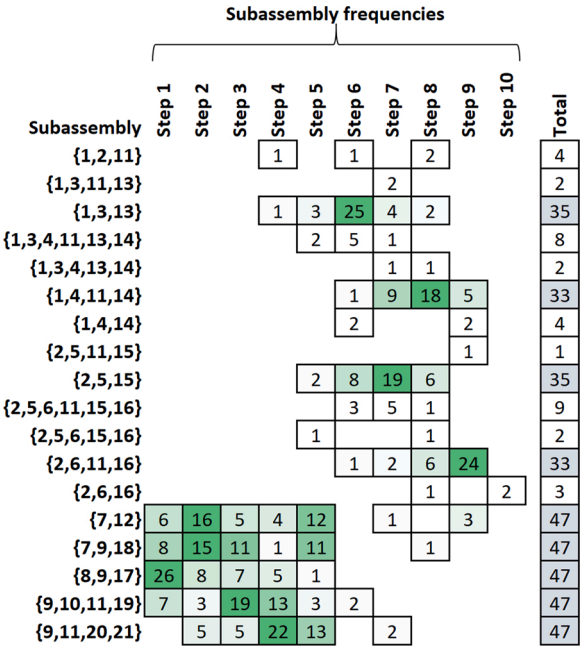

Fig. 13. Subassembly step frequencies. The total can be read both as the sum of the step frequencies and the total frequency of each subassembly. On the side column, in gray, the subassemblies with the greatest frequencies, and in green, the relative step counts, with darker green for higher frequencies. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

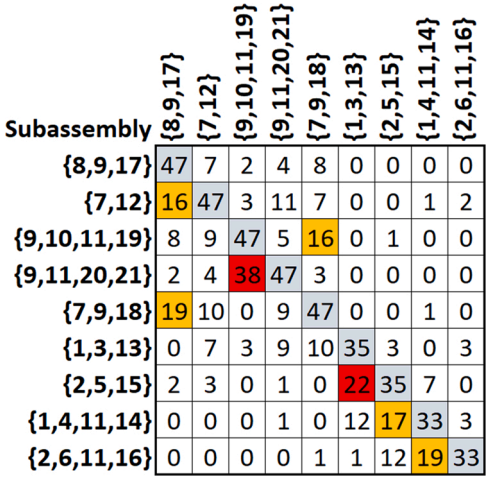

Fig. 14. Statistically reinforced subassembly transition matrix. Total times that each column subassembly is picked right before the row subassembly. The diagonal shows the overall subassembly usage. In orange (middle value) and red (high value), the hard constraints. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

6. Conclusions and future work

This research shows that ASP can be done with statistical RL techniques with a step optimization approach at assembly time. The ASP optimization policies are determined and driven by the competence of the Industry 4.0 skilled operators. Computers, in particular AGS, have to be the interface between the digital world that has computational power to support informed decisions, and the humans who operate in the real world. This is a new approach for an expert operator that can fully interact with the ASP algorithm, or rather be part of it, and drive the ASP optimization function, based on their personal experience on the

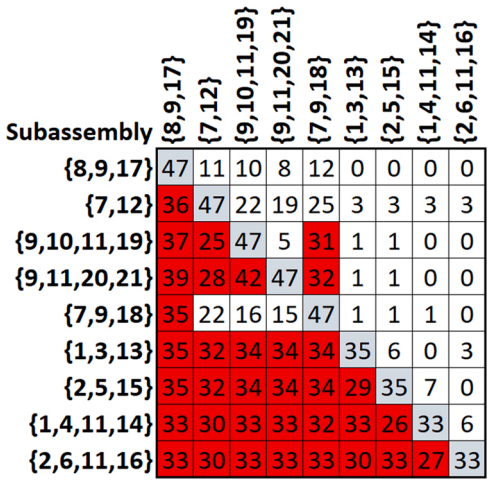

Fig. 15. Statistically reinforced subassembly precedence matrix. Total times that each column subassembly is picked at any step before the row subassembly. The diagonal shows the overall subassembly usage. In red, the hard constraints. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

Table 2
Comparison of assembly instructions between the initial paper instruction ASP from the course instructor and the AGS-reinforced ASP recorded in the fifth experiment.

Instructions	Relative subassembly	Order	
		Instr.	AGS
Assemble the boiler and front cover	{8,9,17}	1	1
Mount the boiler on the frame	{9,10,11,19} and {9,11,20,21}	2	4 and 5
Insert the roof into the cabin	{7,12}	3	2 or 3
Mount the cabin to the boiler	{7,9,18}	4	2 or 3
Mount a wheel on each axle	{1,3,13} and {2,5,15}	5	6
Push the shoulders into the frame	{1,2,11}	6	Almost never
Fit the remaining wheels	{1,4,14} and {2,6,16}	7	Almost never
Push the shoulders into the frame and fit the remaining wheels	{1,4,11,14} and {2,6,11,16}	No (already done in 6 and 7)	7

assembly lines and of an inevitably complex world. The advantage is to be able to tackle any kind of unknown problems before they can arise and add a great deal of adaptiveness and resilience to the industrial operations.

This research sheds light upon how an important peculiarity of RL algorithms is not exploited in industrial processes. This is the exploitation vs exploration pattern – ironically human-inspired – corresponding to asking operators' to either stick to their instructions or take initiative. This research has partly proven that, especially in the context of Industry 4.0, giving options to knowledgeable operators is the way forward to truly take advantage of the online capabilities of RL algorithms in industrial applications, in particular with ASP. Moreover, the detailed approach allows coexistence with more elaborate schemes in which the assembly processes are defined through ontological work [43] or for semi-automated operations requiring high adaptability and self-configuration [44]. This is an added advantage as the production designers can create assembly systems that are gradually automated from manual to fully automated.

The introduction of Marton's variational approach to the ASP, together with the developed AGS, allows adding another couple of components of the Bloom's Taxonomy [45] to the learning experience.

Namely, a trial and error or mechanical operation (depending on the operator's knowledge) is translated in quality time for learning when the initial task of simply *applying* the operations described by the assembly instructions is replaced by the need for the operators to *analyze* the assembly state, report it to the AGS and use it to *evaluate* the best strategy to operate. A change that highlights both the pedagogical success for such an application in the context of the manufacturing course that provides the use case described in this article, and the improved manufacturing outcome that is foreseen in real industrial environments, where operators can learn quickly and efficiently the assembly operations while working on a fully functional production line.

The use of statistical RL methods provides insights on the type of knowledge that artificial intelligence systems can accumulate when interacting with human operators. One is the step frequency diagram, which shows the best statistical assembly sequence that emerges from the wisdom of a crowd of operators. Another is the precedence matrix that can be constructed by applying a conventional threshold to the statistical values obtained by counting the transitions among the elements of the liaison matrix. All these mathematical tools can be used as building blocks for RL algorithms that are meant to drive an online ASP.

Future work should primarily focus on testing the AGS developed with this research on real assembly lines in industry. Firstly, because the results obtained in the experimental setup of this article need to be validated on several different assemblies and, secondly, because of the eventual limitations coming from the didactical assembly that has been chosen for this research that does not allow to optimize other industrial criteria, e.g. assembly execution time. A second line of research could be directed to understanding the reasons behind an operator's choice of assembly sequence. While RL is generally proven to converge towards optimal solutions, it does not explain why a solution in ASP might be optimal. This is against the current line of thought in ASP, but it is in line with the direction taken by deep reinforcement learning, where unsupervised algorithms solve problems without the necessary human understandability.

Another focus of future work is to integrate the AGS into the operator's equipment, with technology such as head mounted devices for augmented reality or any other devices that are operated by multimodal input such as sensors and cameras, other than the operator's speech, gaze, hand gestures or movements in the assembly station.

While online RL algorithms are good optimization tools, they are meant to be used for learning from small data, instead of big data. The latter approach is possible when the source of variation provides plenty of data. Such is the case for the offline approach to RL presented in the literature review of this article. An assembly of customized products might produce only a limited variation of data, therefore the statistical approach with online RL algorithms seems coherent with the industrial requirements but it needs to be verified for its consistency on small assembly data. Can a few attempts to a correct assembly elicit the majority of issues? This is indeed a potential limitation and an important perspective to be addressed in future work.

As this article is meant to pave the way to the application of online statistical RL algorithms to ASP, a major limitation consists in not applying and testing any specific and well-known RL algorithms to online ASP for benchmarking purposes. This has to be done once it is assessed the validity of such approach, which remains the main aim of the research presented in this article and its future work. Since the statistical convergence aspect of online RL has been assessed as promising for ASP, it is indeed a needed future work to test standard RL algorithms and list their pros and cons towards a full use of online RL ASP methods. In particular, strategies about how to give rewards to correct ASP or how and when to apply the exploitation vs exploration strategy, for example if the operator's knowledge can be evaluated before assembly and a threshold can be based on the result of this assessment.

Declaration of competing interest

The authors report no declarations of interest.

Acknowledgements

The authors wish to thank Mats Bejhem, lecturer for the manufacturing course Tillverkningsteknik (MG1026) at KTH Royal Institute of Technology presented as a case study for this paper, for his availability and excellent contributions to the success of this research. A special mention to Jan Stamer and Mikael Johansson for their help in preparing the experimental setups. This research is funded by KTH Royal Institute of Technology in Stockholm, Sweden.

References

[1] Lu Y, Xu X, Wang L. Smart manufacturing process and system automation – a critical review of the standards and envisioned scenarios. J Manuf Syst 2020;56:312–25. https://doi.org/10.1016/j.jmsy.2020.06.010.
[2] Cohen Y, Faccio M, Galizia FG, Mora C, Pilati F. Assembly system configuration through Industry 4.0 principles: the expected change in the actual paradigms. IFAC-PapersOnLine 2017;50:14958–63. https://doi.org/10.1016/j.ifacol.2017.08.2550.
[3] Pellicciari M, Andrisano AO, Leali F, Vergnano A. Engineering method for adaptive manufacturing systems design. Int J Interact Des Manuf 2009;3:81–91. https://doi.org/10.1007/s12008-009-0065-9.
[4] Wang L, Keshavarzmanesh S, Feng H-Y. Design of adaptive function blocks for dynamic assembly planning and control. Int J Ind Manuf Syst Eng 2008;27:45–51. https://doi.org/10.1016/J.JMSY.2008.06.003.
[5] Wang L, Keshavarzmanesh S, Feng HY, Buchal RO. Assembly process planning and its future in collaborative manufacturing: a review. Int J Adv Manuf Technol 2009;41:132–44. https://doi.org/10.1007/s00170-008-1458-9.
[6] Cohen Y, Naseraldin H, Chaudhuri A, Pilati F. Assembly systems in Industry 4.0 era: a road map to understand assembly 4.0. Int J Adv Manuf Technol 2019;105:4037–54. https://doi.org/10.1007/s00170-019-04203-1.
[7] Onori M, Neves P, Akillioglu H, Maffei A, Hofmann A, Siltala N. Dealing with the unpredictable: an evolvable robotic assembly cell. Enabling manuf. compet. econ. sustain. Berlin Heidelberg: Springer; 2012. p. 160–5. https://doi.org/10.1007/978-3-642-23860-4_26.
[8] Maffei A, Onori M, Neves P, Barata J. Evolvable production systems: mechatronic production equipment with evolutionary control. IFIP Adv Inf Commun Technol 2010;314:133–42. https://doi.org/10.1007/978-3-642-11628-5_14.
[9] Schmitt R, Permin E, Kerkhoff J, Plutz M, Böckmann MG. Enhancing resiliency in production facilities through cyber physical systems. Cham: Springer; 2017. p. 287–313. https://doi.org/10.1007/978-3-319-42559-7_11.
[10] Alharbi O. Industry 4.0 operators: core knowledge and skills. Adv Sci Technol Eng Syst 2020;5:177–83. https://doi.org/10.25046/aj050421.
[11] Kaasinen E, Schmalfuß F, Özturk C, Aromaa S, Boubekeur M, Heilala J, et al. Empowering and engaging industrial workers with operator 4.0 solutions. Comput Ind Eng 2020;139:105678. https://doi.org/10.1016/j.cie.2019.01.052.
[12] Alcácer V, Cruz-Machado V. Scanning the industry 4.0: a literature review on technologies for manufacturing systems. Eng Sci Technol Int J 2019;22:899–919. https://doi.org/10.1016/j.jestch.2019.01.006.
[13] Sutton RS, Barto AG. Reinforcement learning: an introduction. MIT press; 2018.
[14] Kuhnle A, Kaiser JP, Theiß F, Stricker N, Lanza G. Designing an adaptive production control system using reinforcement learning. J Intell Manuf 2020:1–22. https://doi.org/10.1007/s10845-020-01612-y.
[15] Hubbs CD, Li C, Sahinidis NV, Grossmann IE, Wassick JM. A deep reinforcement learning approach for chemical production scheduling. Comput Chem Eng 2020;141:106982. https://doi.org/10.1016/j.compchemeng.2020.106982.
[16] Kim YG, Lee S, Son J, Bae H, Do Chung B. Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system. Int J Ind Manuf Syst Eng 2020;57:440–50. https://doi.org/10.1016/j.jmsy.2020.11.004.
[17] Su Y, Mao H, Tang X. Algorithms for solving assembly sequence planning problems. Neural Comput Appl 2020:1–10. https://doi.org/10.1007/s00521-020-05048-6.
[18] Shoval S, Efatmaneshnik M, Ryan MJ. Assembly sequence planning for processes with heterogeneous reliabilities. Int J Prod Res 2017;55:2806–28. https://doi.org/10.1080/00207543.2016.1213449.
[19] Chen RS, Lu KY, Tai PH. Optimizing assembly planning through a three-stage integrated approach. Int J Prod Econ 2004;88:243–56. https://doi.org/10.1016/S0925-5273(03)00187-7.
[20] Wang Y, Tian D. A weighted assembly precedence graph for assembly sequence planning. Int J Adv Manuf Technol 2016;83:99–115. https://doi.org/10.1007/s00170-015-7565-5.
[21] Marton F, Trigwell K. Variatio est mater studiorum. High Educ Res Dev 2000;19:381–95. https://doi.org/10.1080/07294360020021455.
[22] Gualtieri L, Palomba I, Merati FA, Rauch E, Vidoni R. Design of human-centered collaborative assembly workstations for the improvement of operators' physical

ergonomics and production efficiency: a case study. Sustainability 2020;12:3606. https://doi.org/10.3390/su12093606.
[23] Parmentier DD, Van Acker BB, Detand J, Saldien J. Design for assembly meaning: a framework for designers to design products that support operator cognition during the assembly process. Cogn Technol Work 2020;22:615–32. https://doi.org/10.1007/s10111-019-00588-x.
[24] Qi Q, Tao F, Hu T, Anwer N, Liu A, Wei Y, et al. Enabling technologies and tools for digital twin. J Manuf Syst 2019. https://doi.org/10.1016/j.jmsy.2019.10.001.
[25] Ulmer J, Braun S, Cheng CT, Dowey S, Wollert J. Human-centered gamification framework for manufacturing systems. In: Procedia CIRP; 2020. p. 670–5. https://doi.org/10.1016/j.procir.2020.04.076.
[26] Claeys A, Hoedt S, Van Landeghem H, Cottyn J. Generic model for managing context-aware assembly instructions. IFAC-PapersOnLine 2016;49:1181–6. https://doi.org/10.1016/J.IFACOL.2016.07.666.
[27] Moussa M, ElMaraghy H. Master assembly network for alternative assembly sequences. J Manuf Syst 2019;51:17–28. https://doi.org/10.1016/j.jmsy.2019.02.001.
[28] Zhang N, Liu Z, Qiu C, Tan J. A novel assembly sequence design mechanism for assembly sequence planning. 2020 IEEE 7th Int. Conf. Ind. Eng. Appl. ICIEA 2020, Institute of Electrical and Electronics Engineers Inc. 2020:10–4. https://doi.org/10.1109/ICIEA49774.2020.9102101.
[29] Su Q. Computer aided geometric feasible assembly sequence planning and optimizing. Int J Adv Manuf Technol 2007;33:48–57. https://doi.org/10.1007/s00170-006-0447-0.
[30] Trigui M, BenHadj R, Aifaoui N. An interoperability CAD assembly sequence plan approach. Int J Adv Manuf Technol 2015;79:1465–76. https://doi.org/10.1007/s00170-015-6855-2.
[31] Ben Hadj R, Trigui M, Aifaoui N. Toward an integrated CAD assembly sequence planning solution. Proc Inst Mech Eng Part C J Mech Eng Sci 2015;229:2987–3001. https://doi.org/10.1177/0954406214564412.
[32] Li Z, Wang J, Anwar MS, Zheng Z. An efficient method for generating assembly precedence constraints on 3D models based on a block sequence structure. CAD Comput Aided Des 2020;118:102773. https://doi.org/10.1016/j.cad.2019.102773.
[33] Rashid MFF, Hutabarat W, Tiwari A. A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches. Int J Adv Manuf Technol 2012;59:335–49. https://doi.org/10.1007/s00170-011-3499-8.
[34] Lowe G, Shirinzadeh B. Dynamic assembly sequence selection using reinforcement learning. Proc. - IEEE Int. Conf. Robot. Autom., Institute of Electrical and Electronics Engineers Inc. 2004:2633–8. https://doi.org/10.1109/robot.2004.1307458.
[35] Zhao M, Guo X, Zhang X, Fang Y, Ou Y. ASPW-DRL: assembly sequence planning for workpieces via a deep reinforcement learning approach. Assem Autom 2019;40:65–75. https://doi.org/10.1108/AA-11-2018-0211.
[36] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: a survey. J Artif Intell Res 1996;4:237–85. https://doi.org/10.1613/jair.301.
[37] Watanabe K, Inada S. Search algorithm of the assembly sequence of products by using past learning results. Int J Prod Econ 2020;226:107615. https://doi.org/10.1016/j.ijpe.2020.107615.
[38] Yu T, Huang J, Chang Q. Mastering the working sequence in human-robot collaborative assembly based on reinforcement learning. IEEE Access 2020;8:163868–77. https://doi.org/10.1109/access.2020.3021904.
[39] Martinez D, Alenya G, Jimenez P, Torras C, Rossmann J, Wantia N, et al. Active learning of manipulation sequences. Proc. - IEEE Int. Conf. Robot. Autom., Institute of Electrical and Electronics Engineers Inc. 2014:5671–8. https://doi.org/10.1109/ICRA.2014.6907693.
[40] Xing Y, Chen G, Lai X, Jin S, Zhou J. Assembly sequence planning of automobile body components based on liaison graph. Assem Autom 2007;27:157–64. https://doi.org/10.1108/01445150710733423.
[41] Lai HY, Huang CT. A systematic approach for automatic assembly sequence plan generation. Int J Adv Manuf Technol 2004;24:752–63. https://doi.org/10.1007/s00170-003-1760-5.
[42] Raju Bahubalendruni MVA, Biswal BB. Computer aid for automatic liaisons extraction from cad based robotic assembly. 2014 IEEE 8th Int. Conf. Intell. Syst. Control Green Challenges Smart Solut. ISCO 2014 - Proc., Institute of Electrical and Electronics Engineers Inc. 2014:42–5. https://doi.org/10.1109/ISCO.2014.7103915.
[43] Lohse N, Hirani H, Ratchev S. Equipment ontology for modular reconfigurable assembly systems. Int J Flex Manuf Syst 2005;17:301–14. https://doi.org/10.1007/s10696-006-9030-0.
[44] Khabbazi MR, Wikander J, Onori M, Maffei A. Object-oriented design of product assembly feature data requirements in advanced assembly planning. Assem Autom 2018;38:97–112. https://doi.org/10.1108/AA-07-2016-084.
[45] Krathwohl DR. A revision of bloom's taxonomy: an overview. Theory Pract 2002;41:212–8. https://doi.org/10.1207/s15430421tip4104_2.