

# Decentralised vs partially centralised self-organisation model for mobile robots in large structure assembly



Spartak Ljasenko\*, Pedro Ferreira, Laura Justham, Niels Lohse

Intelligent Automation Centre, The Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, UK

## ARTICLE INFO

### Article history:

Received 8 November 2017  
Received in revised form 17 July 2018  
Accepted 7 September 2018  
Available online 25 September 2018

### Keywords:

Self-organisation  
mobile robots  
Large structure assembly  
Multi-agent systems

## ABSTRACT

Currently, manufacturing companies are heavily investing into the automation of manufacturing processes. The push to improve productivity and efficiency is increasing the demand for more flexible and adaptable solutions than the currently common dedicated automation systems. In this paper, the planning problem for mobile robots in large structure assembly was addressed. Despite near-optimal results, the previously developed hybrid agent behaviour model was found to lack responsiveness and scalability. For that reason, an alternative, fully decentralised agent behaviour model was developed and compared to the hybrid one. Through simulated experiments, it was found that the decentralised agent behaviour model achieved much higher responsiveness; however, it required additional spare capacity to compensate for its decision-making imperfections.

© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Manufacturing companies are heavily investing into the automation of assembly processes. Efforts are aimed at increasing the utilisation of automated systems to improve the return of investment and save space on the shop floor. In the aerospace industry, this trend is particularly impacting the drilling and filling tasks because they account for 60% of the airframe assembly costs, 80% of work related injuries and 80% of product defects<sup>1</sup>. Currently deployed manufacturing systems tend to occupy large spaces on the shop floor yet lack use. Therefore, it is understandable that demand has been increasing for more flexible and versatile systems that can be shared between workstations.

In large structure assembly, there is typically a low throughput of products with large working contents [1]. Products are generally large enough to fit multiple manufacturing resources, such as robots, around them, which could increase the work rate. The main challenge in these systems is that products are completed in time. Therefore, the accurate scheduling of jobs is critical for these systems.

A group of European frontrunners, in academia and industry (EUROP), have compiled a Strategic Research Agenda (SRA) [2] for

the next few decades. It highlights the need for system components to use distributed control as opposed to the hierarchical control structures that have previously been employed. This is in agreement with the Industrie 4.0 initiative [3], where there is an intention to convert factories into smart environments by means of, and interlinking with, cyber-physical systems.

Another critical challenge for large structure assembly is how to deal with the product flow across the system. While conveyor belt systems are suitable for the transportation of bulk material for short to mid distances [4], the products in large structure assembly are usually very large, heavy and awkward to handle. For this reason, they cannot be transported via a conveyor belt system and, as such, these products are commonly transported by crane systems. One way of reducing the reliance on the crane system could be deploying a dynamic manufacturing system, comprised of mobile robots. In such a system, the machines move to the products instead of products to machines and the need to transport products around a shop floor can be reduced.

The fundamental differences between fixed automation systems and dynamic systems (consisting of mobile robots) in large structure assembly have already been assessed in [1]. It was concluded that mobile systems could overcome many limitations of fixed automation systems. In addition, given sufficient space, mobile systems could also increase the production rates of selected products by adding more resources to them. Therefore, if applied with appropriate scheduling models and assuming suitable physical capabilities, mobile systems should be better able to deliver products on time. Furthermore, mobile robots should also

\* Corresponding author.

E-mail addresses: [s.ljasenko@lboro.ac.uk](mailto:s.ljasenko@lboro.ac.uk) (S. Ljasenko), [p.ferreira@lboro.ac.uk](mailto:p.ferreira@lboro.ac.uk) (P. Ferreira), [l.justham@lboro.ac.uk](mailto:l.justham@lboro.ac.uk) (L. Justham), [n.lohse@lboro.ac.uk](mailto:n.lohse@lboro.ac.uk) (N. Lohse).

<sup>1</sup> Austin Weber. 2015. Assembly Automation Takes Off in Aerospace Industry. [ONLINE] Available at: <https://www.assemblymag.com/articles/92790-assembly-automation-takes-off-in-aerospace-industry>. [Accessed 13 July 2018]

respond more effectively to various disruptions, that would otherwise strongly affect a manufacturing system's productivity and business opportunities [5]. Examples of typical disruptions for production schedules include machine breakdowns, rush orders, order cancellations, scrap, rework and changes in due dates and times. The conclusions of [1] firstly highlighted the need to further advance mobile robot technologies to achieve higher positioning accuracy and structural stiffness, which will be critical for their wider implementation. Secondly, if the named advances are made, one would require the means to organise mobile robots in a complex and agile environment. From an operational perspective, one of the key challenges will be to maximise the resource utilisation and resilience benefits provided by dynamic mobile systems. This is addressed by distributed decision-making systems, which provide inherent responsiveness, adaptability and scalability.

This paper compares two fundamental agent behaviour models, which enable self-organisation of mobile robots within the context of large structure assembly. The emerging behaviour of a decentralised model is compared to that of the hybrid model's approach from [6]. In both models, all manufacturing resources and products are represented as separate agents. The behaviour models are required to allocate a number of mobile manufacturing resources to products, so as to get the jobs done by their due times. The objective is to characterise how effectively both models minimise the total weighted tardiness (TWT) of the system, which is the penalty imposed from missing the due times, as discussed in [7].

The structure of the paper is as follows: in Section 2, literature relating to this work is reviewed. In Section 3, the scheduling problem is formulated. The decentralised model is presented in Section 4 and compared to the already developed hybrid model in Section 5. The results of the work are discussed in Section 6 and the paper is concluded in Section 7.

## 2. Literature review

To deal with complex distributed systems, the concept of self-organisation has recently gained much interest in various fields. Examples include studies of tissue regeneration in biomaterials [8], magnetisation of ferromagnetic nanowires in nanotechnology [9] and individual pedestrian velocities within crowds in statistical mechanics [10]. In manufacturing, the emphasis has been on how systems can autonomously arrange themselves, to deliver manufactured products on time. Frei and Serugendo reviewed existing literature on self-organising assembly systems and provided a framework for the future of such systems [11]. They concluded that, although the research in this field was still in its early stages, the approach had great potential. The common aspect found across all the different fields where self-organisation is occurring, is that independent system 'particles' work together to bring about some form of order in the system as a whole.

While there are multiple sources for self-organisation models, only the hybrid model presented in [6] addresses the challenges of mobile robots in large structure assembly. This model uses a central Blackboard Agent (BA) to receive and process information about all other agents on the shop floor. It notifies Product Agents (PAs) if it predicts that any of them will be tardy (i.e. late in delivery). In those circumstances, the PAs negotiate with one-another with an overall aim of reducing TWT. Therefore, this model is partially centralised because the BA only processes information and does not make decisions for the PAs.

A multi-agent system for mobile robots was published by Giordani, et al. [12]. They presented a model where Task Agents send resource requests to a Task Coordination Agent. The mobile robots are then allocated to tasks by the Coordination Agent in a

way that minimises the movement distances of the mobile robots. It is a hybrid system, because there is a central element (the Coordinator Agent) and decentralised elements (the tasks and mobile robots). They consider only the minimisation of movement distances in order to minimise the makespan. However, sometimes a critical element for such systems is meeting the due times of individual products. This is of particular interest where final products are assembled out of subcomponents.

The application of mobile robots in the automotive industry has been researched by Michalos et al. [13,14]. In [14], they proposed a method for mobile robots to enable autonomous reconfiguration of production lines. In [13], they assessed the performance of production systems with mobile robots. They concluded that mobile robots reduce the time required to respond to a machine breakdown, as they require a shorter time for reconfiguring the workstation. As a result, the downtime of a mobile robot system is reduced and overall utilisation is increased. The disadvantages of mobile robots, as described in [14], are the challenges associated with accuracy and structural stiffness due to them not being fixed into the ground. Bundles of additional tooling and reinforcements are often required to carry out drilling and filling tasks (see the Kuka MRP in Fig. 1 or ElectroImpact's mobile robot [15]), to deal with the high standards of both automotive and aircraft assembly.

An analogy to this has been found in Autonomously Guided Vehicle (AGV) scheduling, where the emphasis is generally on material handling operations. For example, in [16], the effects of different pickup and dispatching rules on the performance of multiple-load AGVs were studied. Also, in [17], the authors applied the Ant Colony Algorithm for scheduling the transportation of jobs between machines. While not directly applicable, such approaches have proven and reiterated the responsiveness of distributed systems. Responsiveness is of paramount importance in environments of frequent changes and disruptions.

It can also be argued that there are some similar applications in swarm robotics. However, the field of swarm robotics is described as the study of how to coordinate large groups of relatively simple robots through the use of simple rules [18]. They are generally successful in covering a geographical area (i.e. [19]), or combining into various formations (i.e. [20]). However, they lack the level of sophistication necessary for solving work flow problems with objective functions.

The problem of centralised scheduling approaches is their inherent rigidity. This means that reconfiguration and other hardware changes usually require considerable modifications in the software. In addition to that, a single scheduling disruption may sometimes cause the whole manufacturing system to come to a halt due to high computational overheads [21]. Agent-based systems provide a natural solution to these by reducing the processing load on the central entity in hybrid systems, or by removing it completely in fully decentralised systems. Other reasons for adopting such a solution are: the heterogeneous environment, the need for versatility and scalability, the general trend of decentralisation via cyber-physical systems within the scope of Industrie 4.0 [3], the need to enable plug-and-produce (PNP) concepts [22] and elimination of the need to reprogram on a shop floor with potentially frequent perturbations.

The paradigm of multi-agent systems has not only been shown to work, but also to enable code reuse and to support the assembly of a variety of products with different requirements simultaneously. Effectively, a multi-agent system does not "feel" any change-overs of the shop floor [23], unlike the currently common rigid arrangements.

In the field of multi-agent systems, the notion of product intelligence has been gaining increasing interest in the past couple of decades. In the industrial context, it has been used to describe the linking of an order or product to the information and rules

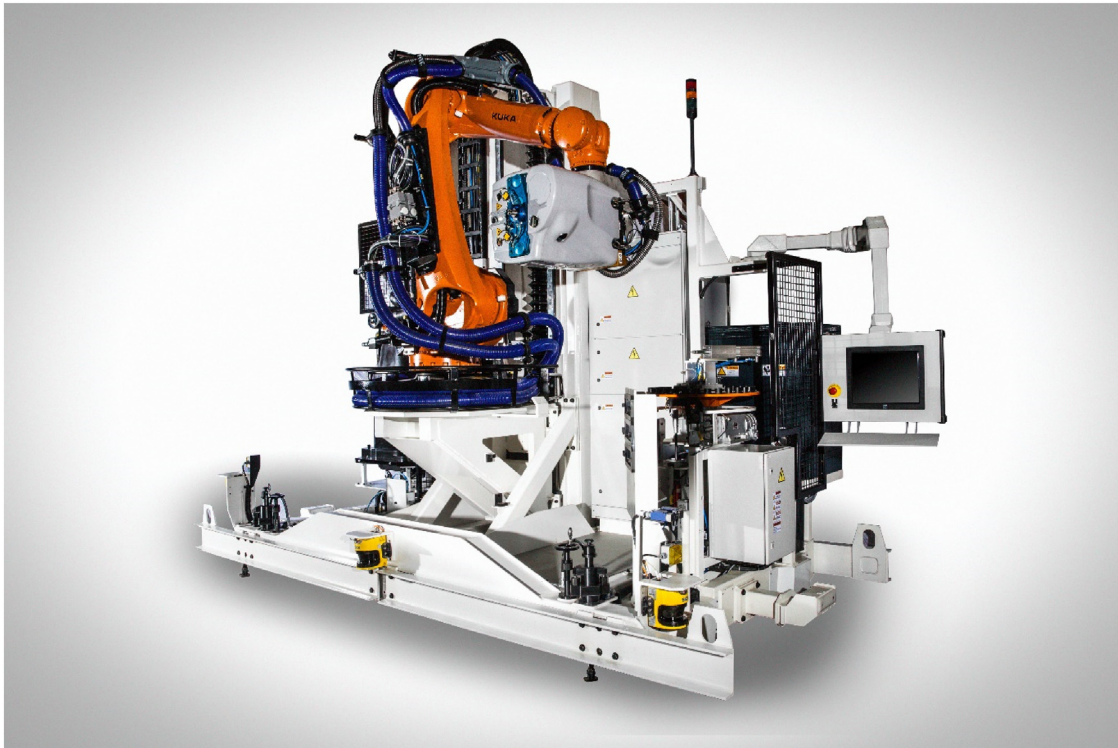


Fig. 1. Mobile Robotic Platform and multifunction robot end effector are used for drilling and fastening wing assemblies (Courtesy of Kuka Systems Aerospace).

governing the way that it is intended to be produced, stored or transported [24]. A notable experience in implementing product intelligence has been published in [25]. There, washing machine assemblies were carried on pallets. Each pallet was represented as a product agent (PA) and equipped with an RFID tag, which provided information to a communication module. The PA then successfully communicated with its surrounding agents (most importantly, the resource agents (RAs)) for allocation and execution of necessary operations. This work has provided the basic agent types required for any such system, as can be seen in [26].

It is generally accepted that centralised structures can achieve near-optimal schedules, given enough computational resource. However, these have a weak response to changes on the shop floor [27], which is characterised by high computational overheads. In [28], it is shown that even with a single machine, the scheduling problem with the objective of minimising tardiness and earliness is an NP-hard problem. This means that from an algorithmic perspective, problems of this type would most likely result in a polynomial increase in computational overheads when the input size (i.e. number of agents, time steps, etc.) is increased. Conversely, decentralised systems respond much better to changes and disturbances. However, this is due to only having partial knowledge of the environment. As a result, their decisions may not be optimal for the manufacturing system as a whole [29].

Clearly, there is no doubt about the importance of self-organisation and distributed systems in the manufacturing context. However, there is a lack of studies that consider which agent behaviour models would be best suited to control large mobile robot systems. The hybrid model presented in [6] achieves very good schedules, however it lacks responsiveness to disruptions and adaptability to new requirements. There is the potential that alternative agent behaviour models can overcome these disadvantages. Therefore, there is scope for two knowledge contributions in this context. Firstly, to develop an alternative

decentralised model for the same scheduling problem, and secondly, to assess its performance in relation to the hybrid model, given a set of relevant industrial scenarios.

### 3. Problem formulation

In order to formulate the problem being considered within this paper, a representative shop floor and scenario was developed. Additionally, the hybrid self-organisation model, which has already been proposed, needed to be defined. This section provides: a listing of the notations used for describing the problem, the key performance indicators, the presentation of a typical scenario on the shop floor, the expression of the constraints for the problem and a brief description of the hybrid self-organisation model [6] that was used for the comparison study.

#### 3.1. Notation

The following notations were used throughout this paper

- $b_j$  – bank roll, the amount of credits in the bank of the PA that is responsible for job  $J_j$
- $C_j$  – completion time of job  $J_j$
- $C_{t,j}$  – tardiness cost of job  $J_j$
- $C_c$  – current contract
- $d$  – the distance between an RA that was offered credits and the offering PA's location
- $G_b$  – bid gap
- $i$  – a workstation  $WS$  column index
- $I_{gb}$  – bid gap increment
- $J_{1..n}$  – a set of jobs (represented by PAs)
- $L_j$  – time required to load job  $J_j$  on any workstation  $WS_{i \dots 1,2}$
- $l_j$  – start of loading of a product  $J_j$  to a workstation  $WS_{i \dots 1,2}$
- $l'_j$  – end of loading of a product  $J_j$  to a workstation  $WS_{i \dots 1,2}$
- $M_{1..m}$  – set of mobile resources
- $m$  – number of mobile resources

- n – number of products
- $p_j$  – processing time of job  $J_j$
- $P_m$  – moving penalty factor
- $S_j$  – start time of processing job  $J_j$
- $t_{c,j}$  – completion time of job  $J_j$
- $t_{d,j}$  – due time of job  $J_j$
- $t_{i,j}$  – launch time of job  $J_j$
- $U_j$  – time required to unload job  $J_j$  from any workstation
- $WS_{i \dots 1,2}$ 
  - $u_j$  – start time of removal of job  $J_j$  from a workstation  $WS_{i \dots 1,2}$
  - $u_j'$  – end time of removal of job  $J_j$  from a workstation  $WS_{i \dots 1,2}$
- TWT – total weighted tardiness
- $V_o$  – value of credit offering
- $WC_j$  – working content of job  $J_j$ . This is a measure of how much work must be processed in job  $J_j$  by mobile robots. Each working RA processes one unit of  $WC_j$  per time step
- $WS_{i \dots 1,2}$  – a workstation with column index  $i$  and a row index of 1 or 2

3.2. Key performance indicators

In addition to presenting the decentralised self-organisation model, the aim of this work was to compare and assess two fundamentally different behaviour models for mobile robots. The purpose was to establish which models provide ideal solutions for scheduling in different scenarios. To achieve a conclusion, two key performance indicators were used that reflect the needs of such a system:

- 1) The total weighted tardiness (TWT): This is a measure of how efficiently a model plans its processing of products with respect to due times and tardiness costs. Generally, there are negative consequences when a product is completed later than its due time. In this work, each product has been allocated a tardiness cost. This counts as a penalty for every unit of time that the

completion of the product has gone past the due time. Therefore, the TWT is a sum of all weighted tardiness costs and is calculated as follows:

$$TWT = \sum (C_{t,j} * (t_{c,j} - t_{d,j})) \tag{1}$$

- 2) Computational effort for rescheduling. When any change or disruption occurs on the shop floor, the self-organisation models should respond with the best possible solution in the shortest possible time. As described in [29], the two fundamentally different behaviour models are expected to perform very differently in such circumstances. For the hybrid model, it is measured in seconds taken to compile a schedule. For the decentralised model, a baseline is defined as the time taken (in seconds) for the longest time step in the simulation. This is because, instead of planning forward, the decentralised model makes decisions through sealed-bid auctions at each time step. Therefore, a disruption can only have an effect on a single negotiation round, where each agent makes its typical decisions just like in any other round. To link the computational effort to the TWT calculations, the time step term had to be defined. Thus, a time step is a second in the schedule of the hybrid model and as a round of bidding in the decentralised model. Firstly, the interest in this comparison is the validation of a negligible negative impact on the decentralised model. Secondly, it is in determining how much the delay affects the performance of the hybrid model with respect to the objective function of minimising TWT.

3.3. Job shop model

To formulate the problem and its complexity, it is important to establish a typical scenario for large structure assembly. Fig. 2 depicts a typical aerospace shop floor enhanced with mobile robots. Similar to many outputs from established literature, the

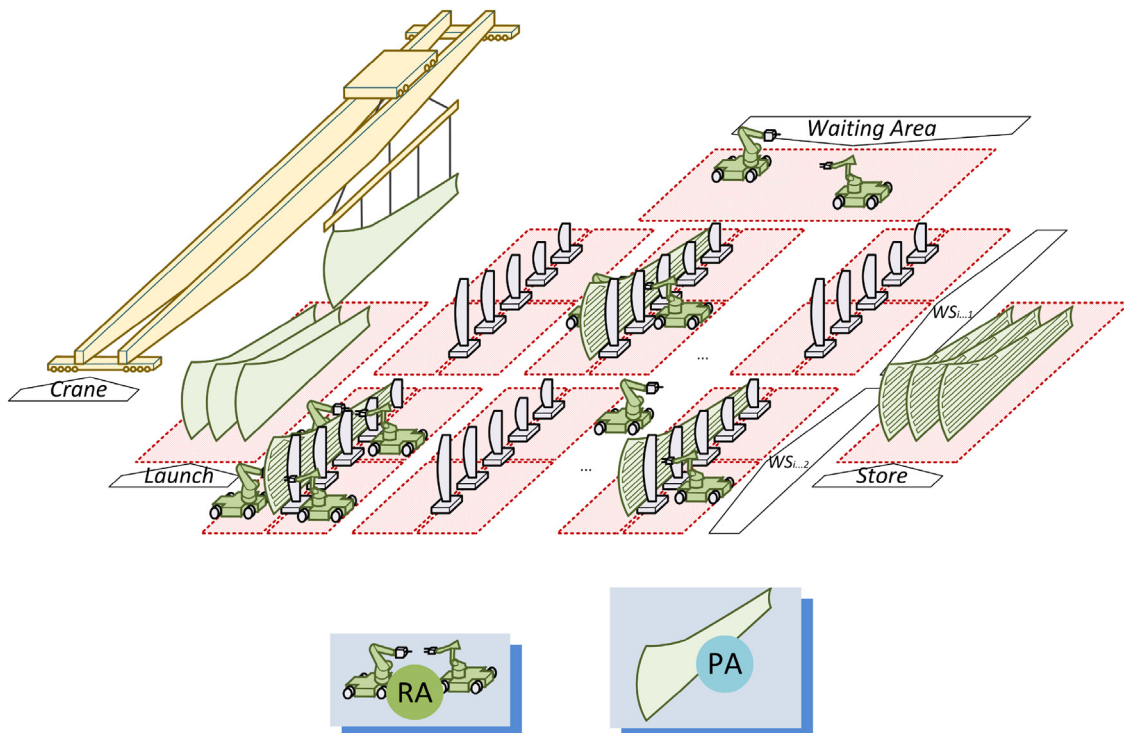


Fig. 2. The principal shop floor layout for the mobile automation system.

jobs in this paper are also represented by PAs and manufacturing resources by RAs. Both of these agent types can be thought of as embedded processing and communicating modules, which work with their own sets of objectives.

Based on this scenario, the elements of the system and their expected relations can be clearly defined. Products  $J_{l..m}$  from the set  $J_m$ , with working content  $WC_j$  of several hours of single-machine processing, are loaded to workstations  $WS_{i..1,2}$ . Once loaded, mobile resources  $M_{l..m}$  may move to them and start processing the products. The completion time  $C_j$  establishes the time the processing finishes. The products are then unloaded from the system.

The mobile robots can move freely to any required workstation  $WS_{i..1,2}$  at any time. Due to the sizes of the products, several mobile robots can fit next to each other, which can increase the work rate on any given product. To simplify the current problem, a maximum of four mobile resources were allowed work on any product. However, it is also important to note that the assumption of more than one resource being able to work on any product would normally increase the probability of blockages due to negotiation conflicts. This problem can arise when too many RAs agree to process the same PA(s). However, the PAs in the models initially only accept as many RAs as they require to be completed on time. Priority is given to the RAs that are closest. Where there is an equal distance, a first-come-first-served policy is used. These policies prevent negotiation conflict bottlenecks, as the PAs are always able to make the final decision.

It has been assumed that the challenges of localisation and structural stiffness will not have any impact in these scenarios. Therefore the mobile robots are assumed to be able to reliably carry out the tasks. Additionally, loading or unloading a job  $J_j$  is considered to take the same time for any workstation  $WS_{i..1,2}$  on the shop floor. There are two reasons for this: Firstly, the loading time for the fastening and vertical movement of products, performed by the crane system, would be much greater than the horizontal movements. The only difference in real scenarios is the horizontal distance between workstations. All products must be lifted to the same height to be transported and then lowered to a suitable height and accurately fastened to their jigs. Thus, given that the vertical movement is expected to take the majority of time and the horizontal variance is more case specific, one can argue that neglecting horizontal movement is a fair assumption. Secondly, the focus of this paper is on the behaviour of the mobile manufacturing system, not on the product supply mechanism. Thus, the performance of the crane system is not critical for the purpose of this work. The final assumption is to consider the mobile robots' moving times as instantaneous. This is supported by [1] where the same scenario showed that the moving times of mobile robots are very small in relation to the time taken to process the products and should therefore be included in the spare capacity.

The following constraints apply to the model:

$$S_j \geq 0, l_j \geq 0, u_j \geq 0, l_j' \geq 0, u_j' \geq 0, t_{l,j} \geq 0 \forall j \in J \quad (1)$$

$$|C_{j, \min}| = t_{l,j} + L_j + p_j \forall j \in J \quad (2)$$

$$S_j \geq l_j' \forall j \in J \quad (3)$$

$$C_j = S_j + p_j \forall j \in J \quad (4)$$

$$u_j \geq C_j \forall j \in J \quad (5)$$

$$m_j \max = m = 4 \quad (6)$$

$$(l_j, l_j') = ]u_j, u_j'], (l_j, l_j') = ]l_{j+1}, l_{j+1}'[ \forall j \in J \quad (7)$$

$$(u_j, u_j') = ]l_j, l_j'], (u_j, u_j') = ]u_{j+1}, u_{j+1}'[ \forall j \in J \quad (8)$$

The first constraint (1) ensures that no activity ( $S_j, l_j, u_j, l_j', u_j', t_{l,j}$ ) can take place before the simulation begins. The second constraint (2) specifies that the earliest possible completion time of any job  $C_j$ ,  $\min$ , is the sum of the time taken to load ( $L_j$ ) and process ( $p_j$ ) a job  $J_j$  after it was launched at time  $t_{l,j}$ . The unloading time is not considered for the TWT calculations, as that is dependent on the crane system's availability. The due time for RAs,  $t_{d,j}$ , is set without considering the unloading as well. Constraint (3) defines that a job can only start being processed at time  $S_j$  after it has finished loading to a workstation at time  $l_j'$ . The completion time  $C_j$  in constraint (4) is the sum of the starting time  $S_j$  added to the processing time  $p_j$  for each agent. Under constraint (5), for each job, unloading may only be started at time  $u_j$  when the processing on that product has been finished at time  $C_j$ . Constraint (6) ensures that the maximum number of resources  $m_j \max$  that can be allocated to processing a single job, in the scenario presented, is 4. The crane system's availability is defined under constraint (7). It establishes that between the start  $l_j$  and finish  $l_j'$  of loading job  $j$ , there can be no unloading ( $u_j, u_j'$ ) or loading of other jobs ( $l_{j+1}, l_{j+1}'$ ) and vice-versa under constraint (8).

This section has provided an overview of the expected operation of a system and has established its boundary conditions. However, it does not establish how one can plan for such an environment. The next sections present the behaviour models that were designed to do that.

### 3.4. Hybrid self-organisation model

The hybrid self-organisation model considered for this work is the one presented in [6]. It is based on the exponential priority aging policy (PAP). An overview of the model is presented in Fig. 3. It establishes that the product agents (PAs) must send their details (location, due time, tardiness cost and working content) to the blackboard agent (BA), at the start of each simulation and after each disruption. By applying the PAP, the BA knows the priority ranking order of all PAs at all times.

Because the PAs order RAs based on their priority ranks, the BA can predict how many RAs each PA will occupy at any moment, hence establishing their predicted completion times. Whenever a PA is predicted to be tardy by the sole application of the PAP, the BA sends a notifying message to the respective PA. That PA then seeks to exchange resources with other PAs, in such a way that would result in the lowest TWT.

In the most common representation, agents are considered to be individual entities that work in their own interests. This approach is most popular in fully decentralised architectures and e-commerce applications (i.e. [30,31]). However, in manufacturing, the emphasis has been on cooperative behaviour in the interests of global objectives. This is most evident in holonic manufacturing systems (i.e. [29,32]). Thus, PAs in this model are cooperative, like in [12].

### 4. Decentralised self-organisation model

In aerospace manufacturing applications, drilling and filling tasks commonly require manipulators to act from both sides of the work piece. This means that mobile resources are required to work in pairs to complete the task, and are consequently modelled as such in the self-organisation models. Similarly, in the hybrid

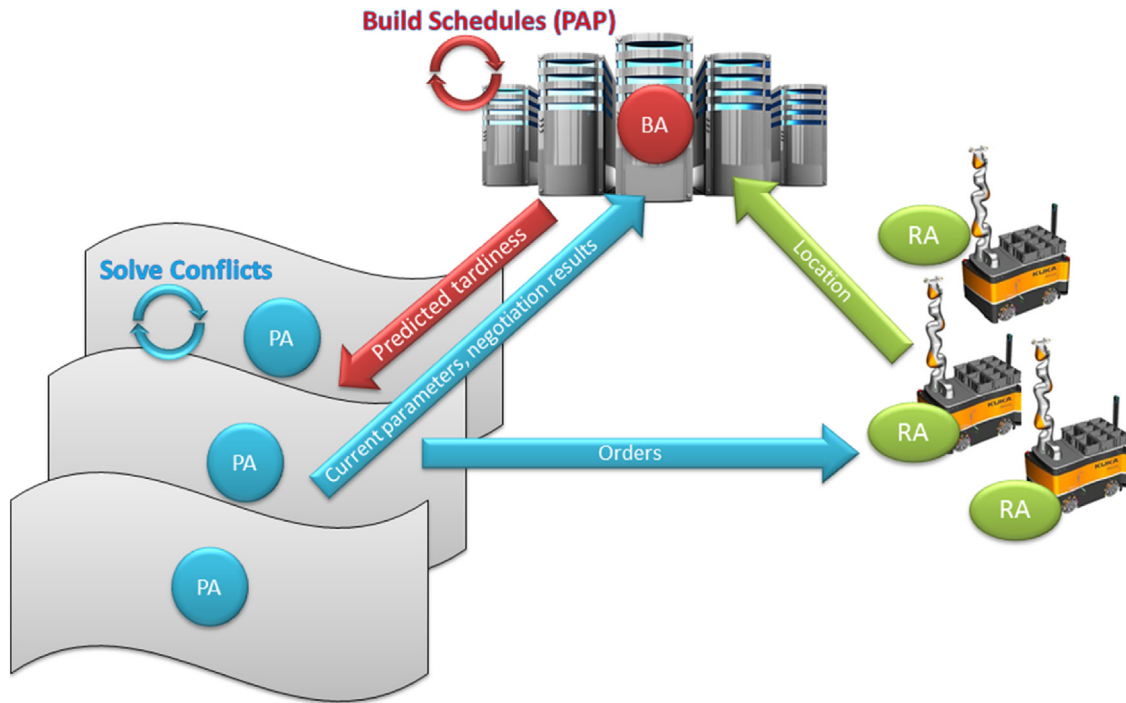


Fig. 3. The overview of the hybrid model's structure.

model, a mobile resource is represented by a single resource agent (RA) that can communicate with other agents.

Also, each product agent (PA) represents the actual product on the shop floor. PAs are launched into the system when the represented product has been loaded onto a workstation  $WS_{i \dots 1,2}$ . The overview of this model's structure is shown in Fig. 4.

In the decentralised model, there is no central unit for coordination. As opposed to the cooperative nature in the hybrid

self-organisation model, each agent in this model is strictly following their own interests. The general flowchart for the self-organisation behaviour of this model is shown in Fig. 5.

An important consideration to avoid bottlenecks is that agents do not wait indefinitely for answers to messages. The agent behaviours, therefore, include timeouts in case a message has failed to reach its addressee, or it is taking too long to receive. Additionally, to keep track of the decision times a timing agent (TA)

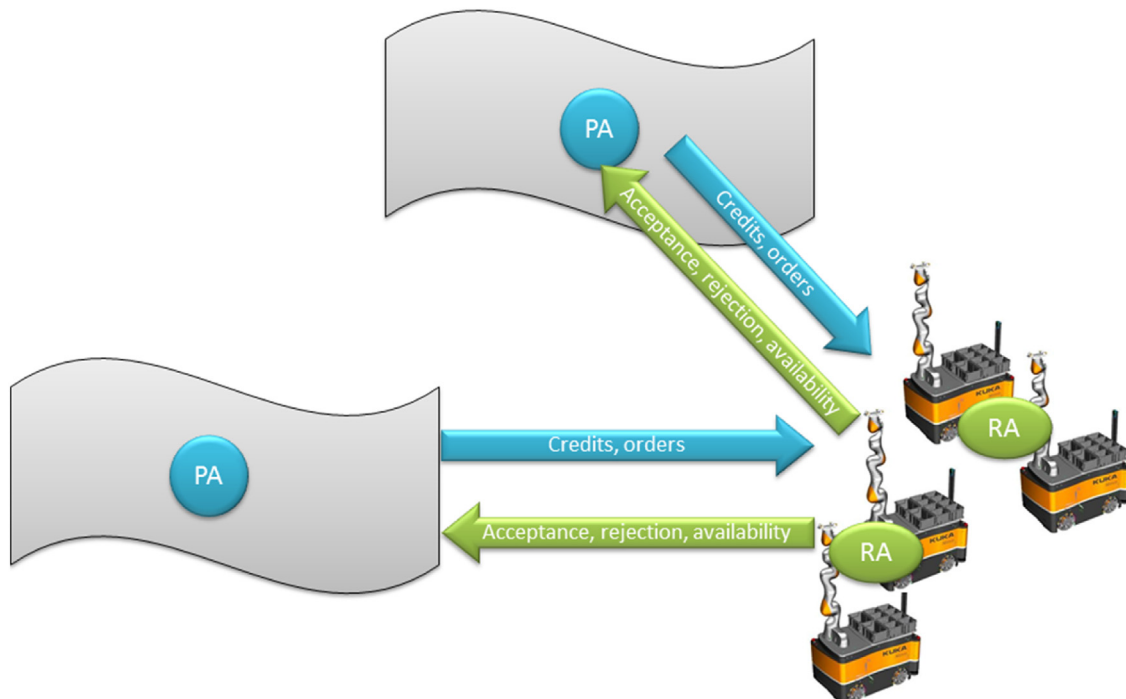


Fig. 4. The overview of the decentralised model's structure.

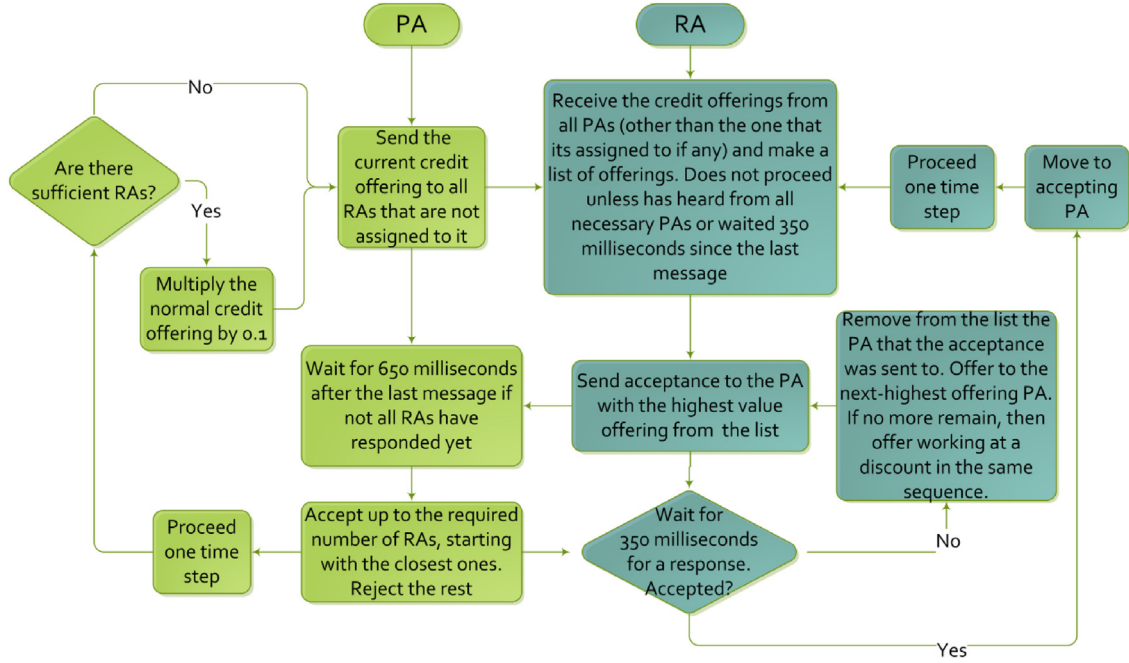


Fig. 5. The general flowchart for the self-organisation behaviour in the decentralised model.

is proposed, which is notified by all agents when negotiations are finished (each round). Once the TA has received the notifications from all agents, or it has waited for the timeout duration of 650 ms, it signals that the time has moved on one time step (bidding round) and negotiation may start again. In the test runs it was determined that the shown times were sufficient for the purposes.

When a PA is launched, it is given credits based on Eq. (2). The initial bank roll  $b_j$  of a job  $J_j$  is the product of multiplying its working content  $WC_j$  and tardiness cost  $C_{t,j}$ . This way, the bargaining power of each PA is linearly proportional to their penalty of being tardy in a real manufacturing environment.

$$b_j = WC_j * C_{t,j} \quad (2)$$

At each bidding round, each PA calculates how many credits it is offering to RAs per time step based on Eq. 3. The fraction of bank roll  $b_j$  divided by the working content  $WC_j$  is the maximum possible credit offering that the PA can make per unit of its working content. The bid gap  $G_b$  is a spending strategy variable for lowering the credit offerings. This was designed with the intention to save credits for the later stages of production. With any success of attracting RAs with high bid gaps, the maximum possible offer becomes higher in the later stages, subsequently making the PA more competitive. As seen in the concept, which is explained and justified in [6], this enables products to start low credit offerings and gradually increase them over time. It was shown that such a method is effective because the most time-pressured PAs bid most aggressively. This is because it is required to increase the chances of attracting the necessary RAs to be completed on time.

$$C_o = \frac{b_j}{WC_j} * (1 - G_b) \quad (3)$$

The bid gap variable defines the spending strategies of PAs. It starts with a pre-set value and remains unchanged if the PA has sufficient RAs assigned to it. However, when the PA has insufficient RAs, it reduces the bid gap by the bid gap increment  $I_{gb}$  at each round of bidding. Thus, each product starts by offering credits that are lowered by the bid gap and gradually increase them if necessary.

The bid gap increment  $I_{gb}$  is calculated as shown in Eq. (4). The increment is designed to gradually increase the credit offerings of PAs if they have insufficient resources. In this model it was designed to reduce the bid gap to 0 by the point when there is 20% of time left to due time if the PA has not received sufficient RAs at any instance. This threshold value was deemed to be a good starting point for testing the expected behaviour of PAs. Even in the most pessimistic of scenarios, each PA would still achieve the maximum credit offerings before its due time. While the proposed value enables the intended behaviour for this study, one could define a set of experiments to determine the most appropriate threshold value. However, this was outside the scope of this work, as the chosen threshold was sufficient to establish the trend in the behaviour.

$$I_{gb} = \frac{G_b}{(t_{d,j} - t_{l,j}) * 0.8} \quad (4)$$

When a PA has sufficient RAs assigned to it, it continues requesting for RAs. However, the credit offering is only 0.1 of the normal value. This was introduced to ensure that free RAs are utilised and jobs get completed before due time where possible.

The behaviour of RAs is straightforward. They actively listen to all offers and compare them to their current contracts. The offered value of a contract is calculated by deducting the movement penalty from the offered credit amount as established in Eq. (5). When the offered value of a contract  $V_o$  is higher than the current contract  $C_c$ , the offer is accepted.

$$V_o = C_c - (d * P_t) \quad (5)$$

The factors that vary in the decentralised model are the spending strategy of the PAs and the moving penalty factors of RAs. The behaviours of the RAs and PAs reflect the individual interests of both agents. PAs aim to get processed by spending as few credits as possible, while RAs aim to earn as much as possible.

In order to understand the impact of different spending strategies, one should vary the starting bid gaps. Therefore, three bid gaps were used to test different spending strategies in the representative scenarios. One aggressive bid gap with little leeway at the end; one

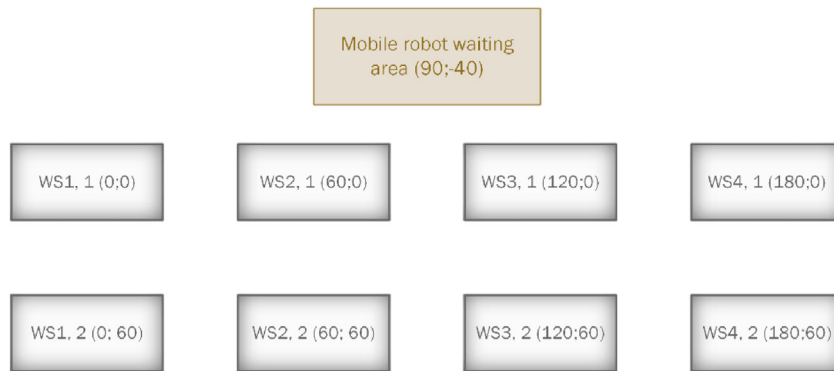


Fig. 6. The workstation layout for the experiments.

conservative bid gap with more leeway; and one balanced bid gap with a pre-defined value between them were used.

Similarly to the three spending strategies for PAs, there were three moving penalty factors considered for the mobile system. These factors were also intended to have low, balanced and high values. Effectively, these penalties dampen the ease of outbidding competitive PAs and reduce the amount of movement by the mobile robots.

#### 4.1. Experiments and results

In experiment 1, the decentralised model at different spending strategies and moving penalty factors is compared to the hybrid model in the assessment of minimising TWT in different scenarios. In experiment 2, the computational overheads of the models are compared. As discussed under the decentralised model's section, the varied factors are the spending strategy of the PAs and the moving penalty factors of RAs. The hybrid model needed no adjustments, because it was shown in [6] that it was optimised already.

The experiments were carried out on a computer with an i3-5020U processor (dual core, 2.2 GHz), 64bit Windows 10 OS and in the JADE agent development environment (version 4.5.0). JADE was configured to use 2048MB of heap memory.

To show how the models apply to multiple machines, a small number of four RAs (equivalent to 8 mobile robots) were deployed to process PAs. There were 8 workstations (2 per resource). Each RA worked at a rate of 1 unit of working content per second, i.e. if all four RAs processed the same product for 1,000 s, then there would be 4,000 s of work done.

The system configuration, which is typical for large structure assembly, introduces a limitation on the launch time, as this depends on the availability of the crane system (CS). The CS uses first-come-first-served logic to load products on each of the 8 workstations. When there are no available workstations to load products on, the CS unloads a randomly chosen workstation with a completed product. In order to prevent the CS from being a supply bottleneck in the system, the time to load and unload products was set to 4,000 s each. This way, the 4 RAs need 10,000 s on average to process any product and the CS need 8,000 s to load and unload any product. As a result, the CS can always supply products faster than they can get processed.

The workstations were laid out as shown in Fig. 6. The distances between workstations were scaled to represent those that would typically be seen in large structure assembly. Currently, one of the largest manufactured products that need a great amount of drilling and filling is the Airbus S380<sup>2</sup> aircraft. Judging by its shape and size, the wing panels are approximated as 40 m long. The RAs consider

the distances between workstations in a straight line and therefore they accommodate for turns and traffic. The gap between adjacent workstations was set to be 60 m. Certainly, the layout might affect the results, as it clearly has an effect on the moving distances of mobile robots. However, investigating that was not the purpose of this work. Moreover, based on the conclusions of [1], the effect was considered negligible.

To ensure a supply bottleneck at the start was not observed and to establish a scenario with some work-in-progress, it is assumed that half of all workstations  $WS_{i \dots 1,2}$  are loaded. The crane system CS then loads new products to available workstations  $WS_{i \dots 1,2}$  and unloads completed products. The production will stop when the last product  $J_j$  is completed.

The different factors of the decentralised model are denominated by DXXYY, where XX stands for the initial bid gap  $G_b$  and YY for the moving penalty  $P_m$ . i.e. D0810 has an initial bid gap of 0.8 and a moving penalty of 1.0.

##### 4.1.1. Experiment 1

This experiment consisted of four sub-experiments, each one representing a specific scenario. In each one there were 20 products with a working content of 40,000 s each. They were launched in a predetermined order and had predetermined properties (working content, due time and tardiness cost). This setup reflects the order in which products are usually launched in the aircraft manufacturing industry, as there are long-standing orders that can be estimated to a good extent in advance. Problems with such an approach can occur when there has been a disruption of any kind. That can result in the reduction of available resources, new due times and changes in priority of certain products.

Scenarios with abundant then sufficient resources with various other complicated conditions were created for both self-organisation models. The general specifications for this experiment are shown in Table 1 and more detailed settings are presented in the Appendix. Sub-experiment 1a was designed to test whether both models, including every variation, could finish the products without tardiness. The reason for this was to confirm the findings in [1], that with sufficient spare capacity, any

Table 1  
The matrix of the first experiments descriptions.

Sub-experiment	Spare Capacity	Additional Challenging Factors
1a	5%	None
1b	0%	None
1c	0%	Every other PA's tardiness cost is halved
1d	0%	Irregular intervals between product launch and due times. Impossible to have TWT = 0.

<sup>2</sup> Airbus. 2018. Technology. [ONLINE] Available at: <https://www.airbus.com/aircraft/passenger-aircraft/a380-family/technology.html>. [Accessed 13 July 2018]



sensibly designed agent behaviour model can achieve a TWT = 0 in the given problem. The flow of products was steady in the sense that every next product was launched with a later due time than the previous ones.

Sub-experiment 1b had the same flow of products, however with no spare capacity. It was designed so that mathematically any deviation or error would cause tardiness of a product. The reason for this sub-experiment was twofold; to confirm that the hybrid model achieved optimal results and to assess how the sub-optimal variations of the decentralised model compared to one-another.

In sub-experiment 1c, all the settings of 1b, other than the tardiness costs, remained the same. Every other PA's tardiness cost was halved. The interest in this sub-experiment was to test the hybrid model's optimisation. Additionally, it was used to assess how the variations of the decentralised model could handle the differences in tardiness costs.

In the final sub-experiment, 1d, all settings from sub-experiment 1b, other than the due times, remained the same. In this case, some products with very rushed due times were designed to be launched so that they would disrupt the natural order of product completions. It was also designed in such a way that mathematically it would be impossible to complete all products on time.

The results for this experiment are shown in Fig. 7. In every sub-experiment the hybrid model achieved optimal results, as expected. The decentralised model's results in sub-experiment 1a confirm the expectations that, despite sub-optimality, the decentralised model achieved TWT=0 at all its behaviour variations when there is 5% spare capacity.

From sub-experiment 1b onwards, the decentralised model gained some TWT at all variations. The results show that the results consistently worsen when the moving penalty factor and initial bid gap are increased. In order to investigate the combinations of moving penalty factors and spending strategies on the TWT in the given sub-experiments, the results were plotted in 3D bar charts below. On these charts, the spending strategy became more aggressive when moving towards the right, ergo, decreasing the variable. The moving penalty factor increases when moving backwards and the TWT increases upwards.

The results of the decentralised model at different settings in sub-experiment 1b are compared in Fig. 8. The increasing moving penalty factor steadily increases the TWT at the two conservative spending strategies (0.8 and 0.5). However, at the aggressive spending strategy (0.2), the highest TWT is gained at the lowest moving penalty factor. More conservative spending strategies lower the resultant TWT. The TWT gained by the most aggressive spending strategy is significantly higher than at the more conservative settings. Lowering the starting bid gaps seems to cause an exponential increase in the resultant TWT. The results

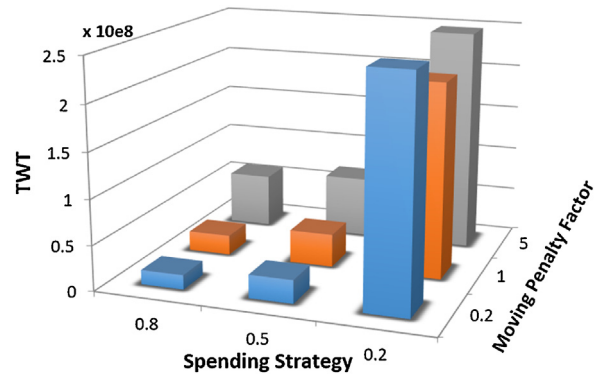


Fig. 8. The decentralised model's results in sub-experiment 1b.

show that when there is no spare capacity, then aggressively spending credits at early stages is clearly not in the interest of PAs.

The same graph plotted for sub-experiment 1c is shown in Fig. 9. Similar trends to 1b are produced; however there is a much smaller spread in the results. The most aggressive spending strategy (0.2) only adds a small proportion of TWT on top of the balanced (0.5) spending strategy. The effects of the moving penalty factor are less clear in this sub-experiment and are generally smaller than the effects of changing the spending strategy.

The bar chart for the final sub-experiment is shown in Fig. 10. The results are very similar to those achieved in sub-experiment 1b: small increase in TWT with increasing moving penalty factors; and a sudden increase of TWT for the most aggressive spending strategy. There is a clear increase of TWT as the starting bid gap of the spending strategy decreases and moving penalty increases.

Throughout experiment 1, the spending strategy of the PAs clearly had a much greater effect on TWT than the moving penalty factor, with a clear relationship showing the increase. The results at more conservative spending strategies (0.8) in each sub-experiment were almost always the best out of the 3 challenging sub-experiments. The TWT was consistently highest at the most aggressive one (0.2). Varying the moving penalty factor, however, was not as predictable. Increasing it usually increased the TWT, but not as consistently as making the spending strategy more aggressive.

Out of the challenging sub-experiments, 1c had the smallest spread in results. This is because the challenge of large differences in tardiness costs between PAs is very difficult to handle for the decentralised model. The PAs with lower tardiness costs and consequently lower credits in the bank, got squeezed out by the wealthier PAs with higher tardiness costs. Therefore, there was very little that the model's parameters could improve.

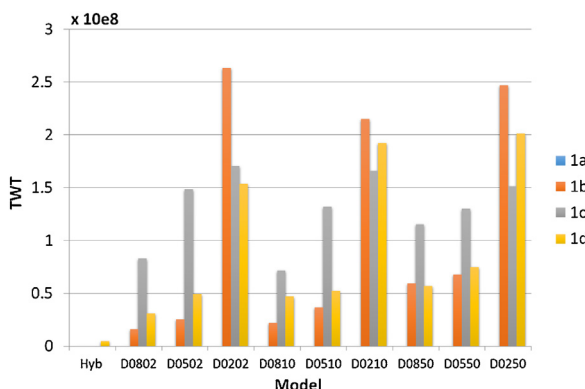


Fig. 7. The results for experiment 1.

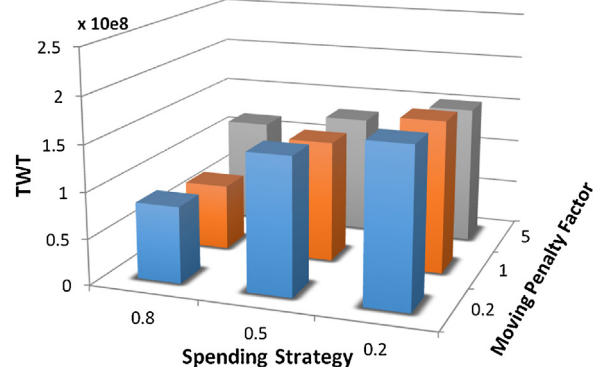


Fig. 9. The decentralised model's results in sub-experiment 1c.

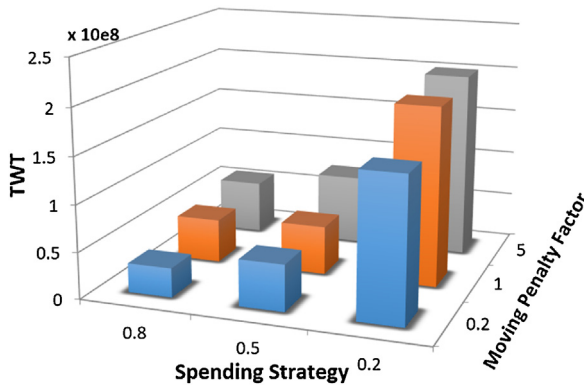


Fig. 10. The decentralised model's results in sub-experiment 1d.

Conversely, sub-experiments 1b and 1d allowed for better results from models with conservative spending strategies. In these sub-experiments, each PA had the same tardiness cost. Clearly, aggressively spending in the early stages was counterproductive for PAs, because that left them with low credits at the later stages where there was a significantly higher TWT.

The variance was restricted in this work, because the model was designed to be executed in a synchronous turn base manner. There was also no noise in the experimental setup. Thus, all decisions were taken at the exact same time and processing the same parameters. Therefore, it was sufficient to run each experiment only once.

Experiment 2 links to experiment 1 by measuring the rescheduling computational effort for both models. The time taken to reschedule is then included in the schedule as a penalty to simulate the effect of a disruption. Thus, the responsiveness of both models is included in assessing their performance with relation to minimising TWT.

4.1.2. Experiment 2

In this experiment, the rescheduling computational effort of both models was measured. The purpose of this experiment was to highlight the penalty of the hybrid model due to the rescheduling effort in relation to the decentralised model. The hybrid model would be much less effective without planning forward, because the BA would not know whether to notify PAs about predicted tardiness. Therefore, the hybrid model's performance in relation to the objective function of minimising TWT is dependent on its responsiveness to disruptions.

On the contrary, the decentralised model is a very versatile self-organisation model where agents take fast and straightforward decisions at each round of bidding. By design, the decentralised

model is not affected by increasing the frequency of disruptions that may arise in the manufacturing process. This is because the agents in this model do exactly the same at each round of bidding. Thus, its' response time was considered as the baseline for the comparison against that of the hybrid model's.

The assessment of this is important because it becomes possible to estimate the total computational effort and its effect in a real manufacturing system. This can be especially useful if the frequency of disruptions can be estimated.

This experiment was designed to assess the responsiveness of both models. It consisted of two sub-experiments: In sub-experiment 2a, the effect of varying the working content of products was assessed and in sub-experiment 2b, the effect of varying the number of products was assessed. These are the two variables that affect the size of the schedules and consequently the time it takes to process them. Other variables, such as tardiness costs, launch times and due times were of no interest in this experiment, because they would not affect the processing time. It must be noted, however, that there was no time pressure for the PAs to be completed. Therefore, the hybrid model did not need to trigger the swapping negotiations to then reschedule with swapped resources.

As per the first experiment, there were four mobile resources deployed. It is recognised that, in a realistic environment with frequent disruptions, the computational overheads would increase each time there was a disruption of any type. However, in this experiment, the scenarios were limited to a single hypothetical disruption.

Before carrying out this sub-experiment, testing was done to determine the limitations of the specified hardware and heap memory. Based on that, the experiment was bound at a maximum of 90 products and working contents of 40,000s each. Thus, sub-experiment 2a was carried out with 90 products and sub-experiment 2b was carried out with a working content of 40,000s per job. The results are shown in Fig. 11. For the hybrid model, a change in the working content was linearly proportional to the required computational effort when rescheduling. Whereas, an increase in the number of products resulted in an exponentially proportional increase in the computational overheads.

To establish a comparison, the decentralised model was run through this sub-experiment as well. The longest round of bidding took 1.5 s to process. As such, this value provided a baseline for the model's rescheduling computational effort. This is because it performs the same actions at each round of bidding and is thus unaffected by disruptions on the shop floor.

These results reveal the key characteristics of the behaviour models. For the hybrid model, increasing the planning horizon increases the computational overheads linearly. Whereas increasing the number of products increases the computational overheads exponentially. In both cases, the decentralised model responds in the

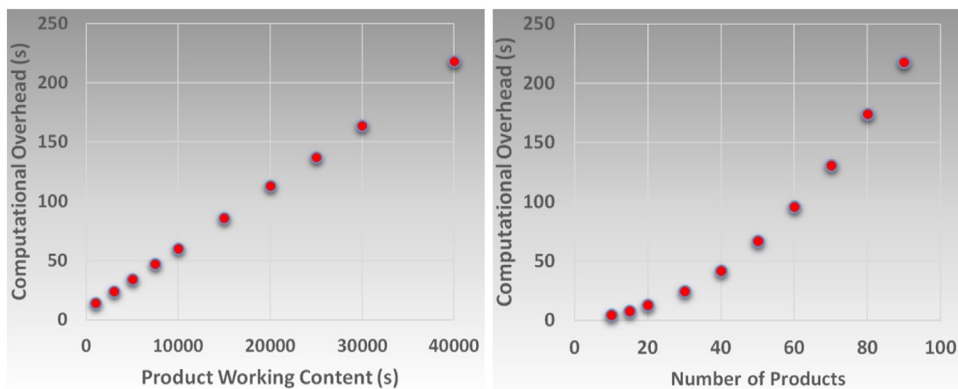


Fig. 11. The results for sub-experiments 2a (left) and 2b (right).

same manner regardless of altering the abovementioned variables. Thus, the feasibility of the hybrid model is strictly dependent on those variables and on the frequency of disruptions.

At the equivalent setting of the first experiment ( $n = 20$ ,  $WC_j \dots j = 40,000$ ), the hybrid model required approximately 13 s to process the schedule. Hypothetically, if the production process was halted for that duration, then the results for the hybrid model in the first experiment would look as represented by “Hyb\*” in Fig. 12. It represents a very extreme case of a hypothetical scenario, where the disruption occurred shortly after the initial schedule was compiled. This way, the rescheduling proportion of the schedule is highest and the highest number of products is affected by it. It is shown that, with 5% spare capacity, it still does not result in any TWT. With 0% spare capacity; a negligible amount of TWT is generated in relation to the decentralised model. Thus, the advantage of the decentralised model’s responsiveness is not of substantial value in the given scenarios with a single disruption.

## 5. Discussion

The hybrid model consistently achieved the best and optimal results in the given simulations in relation to minimising TWT. This result was expected, because the model is optimised for that purpose. The performance is achieved due to having a single entity (BA) in the system that receives global knowledge of the whole environment. However, for the same reason, it must process a large amount of information and notify PAs of tardiness when necessary. As shown in experiment 2, this can be very computationally demanding to do. Furthermore, if the supplied schedule is not optimal and PAs signal that they have agreed to swap resources, the time required increases further. The hybrid model’s computational overheads reached approximately 233 s when processing 90 products with working contents of 40,000 units each.

The processing time for compiling the specific schedule may be dramatically decreased by using a more powerful computer or cloud computing services instead of the specified computer. However, the scenarios considered as part of this work are only a single stage of the assembly process. With added complexity in the scheduling problem, the computational effort for the hybrid model would increase further along the trend lines.

An argument in favour of the hybrid model is that the system may not necessarily need to stop after a disruption has occurred. In some cases it may even be better to proceed with a sub-optimal schedule for the duration. This would result in higher utilisation and lower TWT than completely halting the system. Such an approach could be suitable for environments with a low frequency of disruptions.

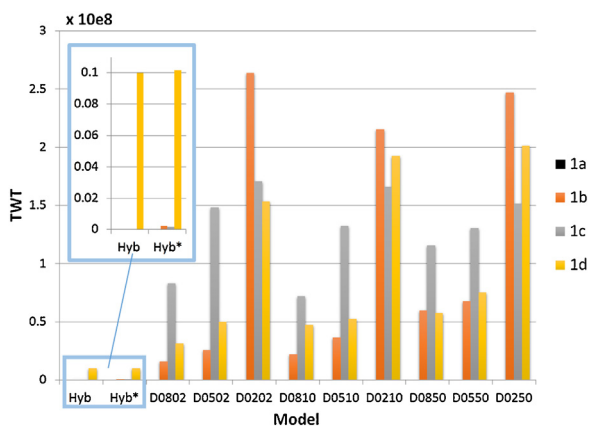


Fig. 12. The results for the first experiments with added tardiness due to computational overheads.

However, if disruptions are frequent, it is possible that new disruptions occur during the time when the hybrid model is still responding to the previous one. As a result, there would be little to no sense in using the hybrid model at all. Thus, in addition to the challenges of extending the planning horizon and increasing the number of entities, the hybrid model is also limited by the frequency of disruptions.

Unavoidable problems for the hybrid model would occur if the manufacturing system was scaled up excessively or the code needed frequent and significant changes. With excessive upscaling, the required computational effort would eventually become too large for efficient operation.

The other issue with the hybrid model would be the coding challenge. It is accepted that behaviour models with centralised architectures have a greater volume and complexity of code than those with fully decentralised architectures. Therefore, expanding the hybrid model’s code further is demanding in two ways: the software engineering effort and the hardware that processes it.

The decentralised model achieved sub-optimal results in situations where there was no spare capacity designed into the product flow. Where there was 5% spare capacity, the model handled the experiment at every setting without gaining any TWT. Considering that in the North American automotive industry, machines typically operate at efficiency levels of 60–70% [33], the setup in sub-experiment 1a was not too optimistic.

Further in the first experiment, the models were given tight due times (0% spare capacity) in various scenarios. As opposed to the hybrid model, the decentralised model did not achieve optimal results in these. A relatively regular pattern could be identified in the results. The model consistently performed better when the PAs were initially set to offer smaller credits at their bidding rounds. Thus, the PAs saved credits for the later stages at the expense of having lower odds of attracting RAs at the start. This worked well for two reasons: Firstly, PAs had high bargaining power when close to their due times, and secondly, newly launched PAs could not compete with the ones close to finishing a job. Conversely, PAs that offered high amounts of credits from early stages onwards were much less competitive nearer the due time when new PAs were already being launched. This finding is in agreement with the work that the hybrid model was based on in [6]. Furthermore, competing PAs were further obstructed by the moving penalty factor that the RAs had to consider before moving from one PA to another. Very large proportions of TWT for the decentralised model, with aggressive spending strategies, resulted from when PAs had already missed their due times and could not outbid other PAs for the remaining few resources.

The presented decentralised model is only one out of a vast range of possible models that could be developed for the given purpose. It was also tested at only 9 different setups, meaning that it is unlikely that it performed to its best ability in the given scenarios. The results, however, confirmed a very important point from [1]: In steady situations with sufficient spare manufacturing capacity, the self-organisation models needn’t be complicated at all. Without spare capacity, the decentralised model gained some TWT at every setting and scenario. This indicates that the given model must have some spare capacity in the system to compensate for imperfections. In the future, it would be interesting to determine the exact amount at which the model started gaining TWT.

The power of the decentralised architecture would be amplified in a larger and perhaps more realistic manufacturing system. Such a system can include different stages of assembly and different kinds of skills for mobile robots. The illustration in Fig. 12 represents a possible shop floor layout for such a system. In a more complex variant of this problem, a mix of products with different skill and timing requirements would be launched in designated areas on the shop floor. Firstly, this layout adds optimisation complexity, because now the individual skills of RAs and their requirements for PAs will have to be considered. Secondly, the larger job shop layout would

have more agents on it. In such a layout, it would be possible to vary the local spare capacities and eliminate bottlenecks in the areas by transferring resources between them (Fig. 13).

Knowing that the computational effort is exponentially proportional to the increase in the number of agents, the rescheduling time for the hybrid model in this case would further increase by a large amount. From the experiments in this paper, it is difficult to estimate how significant it would be for any specific system, as there can be many sizes and variations to it. Nevertheless, despite the sub-optimality of the decision-making, the decentralised model would continue with its normal behaviour and high responsiveness. The only things that could make the decentralised model respond slower would be the additional code and messaging required for negotiations. The messaging could be reduced by introducing more localised messaging, so that very distant (and therefore highly penalised) RAs would not even receive messages. Based on the decentralised model's results, it is fair to assume that adding agents would cause a negligible reduction in responsiveness. Therefore, the decentralised system has a natural advantage over the hybrid model in terms of computational effort, dealing with complexity and time required to respond to changes.

Because the processing times in large structure assembly are very long in comparison to the computational efforts shown in this paper, the models can also take a pre-negotiating approach (similar to [34]). The advantage would be the fact that, at each instance, the agents have either already negotiated or are currently negotiating on the next step(s), in effect eliminating the wait between predictable events. However, in such a setup, both models would not have an immediate response for an unpredictable event and would still need

to negotiate/schedule as was done in this paper. Furthermore, the high computational effort of the hybrid model could potentially make this infeasible due to the necessary time and hardware costs.

## 6. Conclusions

In this paper, a novel decentralised self-organisation behaviour model for mobile robots in large structure assembly was presented. The model was then compared to the previously developed hybrid model in a range of experiments.

The experiments confirmed the natural advantages and disadvantages of both model architectures. The hybrid model achieved optimal scheduling results at the expense of higher computational effort. The decentralised model, on the other hand, did not achieve the optimal results in challenging scenarios. However, it achieved 0 TWT with 5% spare capacity at all behaviour settings. It also showed that it constantly experiences low computational loads regardless of the environment. Such behaviours are typical for these types of systems. Therefore, the work in this paper confirms existing theory from this field.

The decentralised model is a very versatile and adaptable model that does not get impeded by computational effort. It is well-suited for environments where there are frequent modifications and scaling on the shop floor. Its weakness is the sub-optimality, which requires additional capital investment in order to have some spare capacity in the system to make up for the lower efficiency of resource utilisation.

Therefore, it can be concluded that if a mobile manufacturing system is not large enough to cause computational issues, nor will it

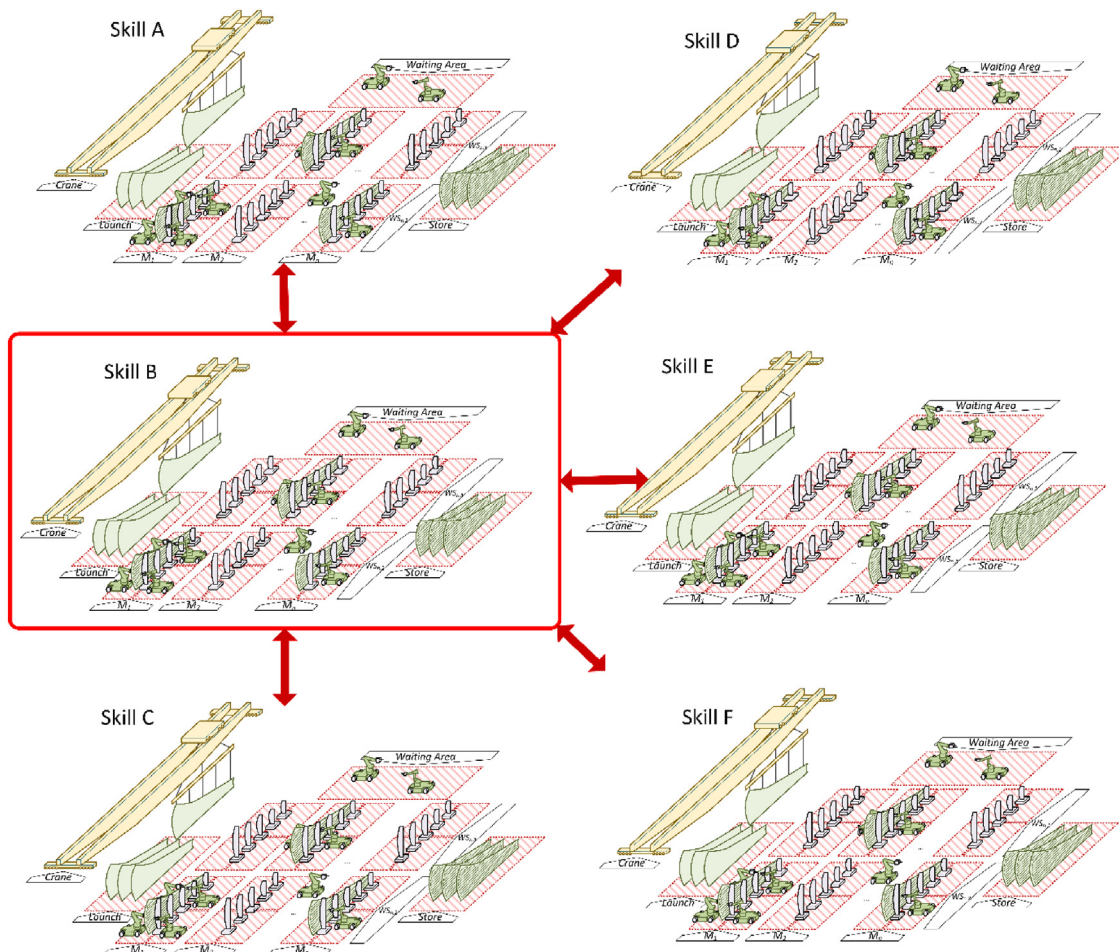


Fig. 13. A sample expanded job shop layout.

need many changes in its lifetime, the hybrid model is the better option. This is due to a more efficient utilisation of existing resources. However, if the system is very large or is expected to grow, then it could be worth using the decentralised model. Whilst the decentralised model may require a small proportion of additional capital investment in the beginning, it is highly likely to overcome many problems later in the manufacturing system's life cycle.

For further work, it is proposed that the decentralised model is tested with moving times included in the simulation. It would be interesting to investigate the impact of moving time regardless of its small proportion. Depending on the results, it could then be reasonable to set high moving penalty factors to discourage excessive moving.

Also, it is inconclusive how much spare capacity would be the minimum for the decentralised model to achieve  $TWT=0$  in the given scenarios. As 5% was clearly more than enough and 0% was insufficient, it would be important to evaluate this more rigorously and to find a better estimation.

Furthermore, it will make sense to analyse how the models perform in noisy environments. Noise can be added in the form of asynchronous messaging or randomised experimental variables, for example. Such an analysis would require a much higher sample size for experiments. Knowing the results would enable the prediction of the behaviour factors that achieve the best results for the decentralised model.

Finally, a case study from a representative manufacturing environment should be put together and used for testing the models. Eventually, the scenarios should expand to a multi-stage manufacturing shop floor that could have thousands of agents on it with different skills and requirements. The interest would firstly be in determining at which complexity the hybrid model would become unusable. Secondly, it would be in identifying whether the decentralised model would continue performing to the same standards.

## Acknowledgement

The authors acknowledge support from the EPSRC Centre for Innovative Manufacturing in Intelligent Automation, in undertaking this research work under grant reference number EP/IO33467/.

## Appendix A.

### Tables A1–A4

**Table A1**

The experimental settings for sub-experiment 1a.

Product Number	Due time	Tardiness Cost	Working Content (s)
1	40,000	1000	40,000
2	40,000	1000	40,000
3	40,000	1000	40,000
4	50,000	1000	40,000
5	60,000	1000	40,000
6	70,000	1000	40,000
7	80,000	1000	40,000
8	90,000	1000	40,000
9	100,000	1000	40,000
10	110,000	1000	40,000
11	120,000	1000	40,000
12	130,000	1000	40,000
13	140,000	1000	40,000
14	150,000	1000	40,000
15	160,000	1000	40,000
16	170,000	1000	40,000
17	180,000	1000	40,000
18	190,000	1000	40,000
19	200,000	1000	40,000
20	210,000	1000	40,000

**Table A2**

The experimental settings for sub-experiment 1b.

Product number	Due time	Tardiness Cost	Working Content (s)
1	40,000	1000	40,000
2	40,000	1000	40,000
3	40,000	1000	40,000
4	40,000	1000	40,000
5	50,000	1000	40,000
6	60,000	1000	40,000
7	70,000	1000	40,000
8	80,000	1000	40,000
9	90,000	1000	40,000
10	100,000	1000	40,000
11	110,000	1000	40,000
12	120,000	1000	40,000
13	130,000	1000	40,000
14	140,000	1000	40,000
15	150,000	1000	40,000
16	160,000	1000	40,000
17	170,000	1000	40,000
18	180,000	1000	40,000
19	190,000	1000	40,000
20	200,000	1000	40,000

**Table A3**

The experimental settings for sub-experiment 1c.

Product number	Due time	Tardiness Cost	Working Content (s)
1	40,000	1000	40,000
2	40,000	500	40,000
3	40,000	1000	40,000
4	40,000	500	40,000
5	50,000	1000	40,000
6	60,000	500	40,000
7	70,000	1000	40,000
8	80,000	500	40,000
9	90,000	1000	40,000
10	100,000	500	40,000
11	110,000	1000	40,000
12	120,000	500	40,000
13	130,000	1000	40,000
14	140,000	500	40,000
15	150,000	1000	40,000
16	160,000	500	40,000
17	170,000	1000	40,000
18	180,000	500	40,000
19	190,000	1000	40,000
20	200,000	500	40,000

**Table A4**

The experimental settings for sub-experiment 1d.

Product number	Due time	Tardiness Cost	Working Content (s)
1	40,000	1000	40,000
2	40,000	1000	40,000
3	39,000	1000	40,000
4	40,000	1000	40,000
5	50,000	1000	40,000
6	60,000	1000	40,000
7	90,000	1000	40,000
8	95,000	1000	40,000
9	80,000	1000	40,000
10	100,000	1000	40,000
11	120,000	1000	40,000
12	120,000	1000	40,000
13	110,000	1000	40,000
14	150,000	1000	40,000
15	140,000	1000	40,000
16	160,000	1000	40,000
17	180,000	1000	40,000
18	170,000	1000	40,000
19	190,000	1000	40,000
20	200,000	1000	40,000

## References

- [1] S. Ljasenko, N. Lohse, L. Justham, A comparison of the manufacturing resilience between fixed automation systems and mobile robots in large structure assembly, 49th CIRP Conference on Manufacturing Systems, (2016) vol. 0.
- [2] EUROP, "Robotic Visions to 2020 and Beyond – the Strategic Research Agenda for Robotics in Europe.", (2009) .
- [3] H. Kagermann, W. Wahlster, J. Helbig, "Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.01," no. April. p., (2013) , pp. 82.
- [4] S. Zhang, X. Xia, Optimal control of operation efficiency of belt conveyor systems, *Appl. Energy* 87 (no. 6) (2010) 1929–1937.
- [5] P. Leitão, J. Barata, An agent-based disturbance handling architecture in manufacturing control, *Ifac Proc. Vol. 40* (no. 3) (2007) 50–55.
- [6] S. Ljasenko, N. Lohse, L. Justham, I. Pereira, A self-organisation model for mobile robots in large structure assembly using multi-agent systems, *SOHOMA'16 Workshop*, (2016) , pp. 1–8.
- [7] Y. Liu, H. Dong, N. Lohse, S. Petrovic, N. Gindy, An investigation into minimising total energy consumption and total weighted tardiness in job shops, *J. Clean. Prod.* 65 (2014) 87–96.
- [8] Z.I. Foraida, T. Kamalidinov, D.A. Nelson, M. Larsen, J. Castracane, Elastin-PLGA hybrid electrospun nanofiber scaffolds for salivary epithelial cell self-organization and polarization, *Acta Biomater.* 62 (2017) 116–127.
- [9] A.A. Ivanov, V.A. Orlov, Self-organization of the magnetization in ferromagnetic nanowires, *J. Magn. Magn. Mater.* 440 (November) (2017) 217–220 2016.
- [10] M. Saberi, K. Aghabayk, A. Sobhani, Spatial fluctuations of pedestrian velocities in bidirectional streams: exploring the effects of self-organization, *Phys. A Stat. Mech. Appl.* 434 (2015) 120–128.
- [11] R. Frei, G.D.M. Serugendo, Self-organizing assembly systems, *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* 41 (no. 6) (2011) 885–897.
- [12] S. Giordani, M. Lujak, F. Martinelli, A distributed multi-agent production planning and scheduling framework for mobile robots, *Comput. Ind. Eng.* 64 (January 1) (2013) 19–30.
- [13] G. Michalos, N. Kousi, S. Makris, G. Chryssolouris, Performance assessment of production systems with mobile robots, *Procedia CIRP* 41 (2016) 195–200.
- [14] G. Michalos, K. Kaltsoukalas, P. Aivaliotis, P. Sipsas, A. Sardelis, G. Chryssolouris, Design and simulation of assembly systems with mobile robots, *CIRP Ann. Manuf. Technol.* 63 (no. 1) (2014) 181–184.
- [15] T. Gray, D. Orf, G. Adams, Mobile Automated Robotic Drilling, Inspection, and Fastening, (2013) , pp. 1–7.
- [16] Y.-C. Ho, S.-H. Chien, A simulation study on the performance of task-determination rules and delivery-dispatching rules for multiple-load AGVs, *Int. J. Prod. Res.* 44 (no. 20) (2006) 4193–4222.
- [17] M. Saidi-Mehrabad, S. Dehnavi-Arani, F. Evazabadian, V. Mahmoodian, An Ant Colony Algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs, *Comput. Ind. Eng.* 86 (2015) 2–13.
- [18] F. Navarro-garcia, A. Serapio-palacios, P. Ugalde-silva, G. Tapia-pastrana, An Introduction to swarm robotics, *ISRN Robot.* 2013 (2013).
- [19] S. Papatheodorou, A. Tzes, Y. Stergiopoulos, Collaborative visual area coverage, *Rob. Auton. Syst.* 92 (2017) 126–138.
- [20] M.H.A. Majid, M.R. Arshad, Hydrodynamic effect on V-shape pattern formation of swarm autonomous surface vehicles (ASVs), *Procedia Comput. Sci.* 76 (no. Iris) (2015) 186–191.
- [21] W. Shen, D.H. Norrie, Agent-based systems for intelligent manufacturing: a state-of-the-art survey, *Knowl. Inf. Syst.* 1 (2013) 129–156.
- [22] T. Arai, Y. Aiyama, Y. Maeda, M. Sugi, J. Ota, "Agile assembly system by 'plug and produce,'" *CIRP Ann. Manuf. Technol.* 49 (January 1) (2000) 1–4.
- [23] P. Leitão, Agent-based distributed manufacturing control: a state-of-the-art survey, *Eng. Appl. Artif. Intell.* 22 (no. 7) (2009) 979–991.
- [24] D. McFarlane, V. Giannikas, A.C.Y. Wong, M. Harrison, Product intelligence in industrial control: theory and practice, *Annu. Rev. Control* 37 (no. 1) (2013) 69–88.
- [25] P. Leitão, N. Rodrigues, J. Barbosa, C. Turrin, A. Pagani, Intelligent products: the grace experience, *Control Eng. Pract.* 42 (2015) 95–105.
- [26] M. Onori, A. Maffei, F. Durand, P. Engineering, K.T.H. Royal, "The IDEAS plug & produce system, International Conference on Advanced Manufacturing Engineering and Technologies The, (2013) .
- [27] M. Mes, M. van der Heijden, A. van Harten, Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems, *Eur. J. Oper. Res.* 181 (no. 1) (2007) 59–75.
- [28] L. Wan, J. Yuan, Single-machine scheduling to minimize the total earliness and tardiness is strongly NP-hard, *Oper. Res. Lett.* 41 (no. 4) (2013) 363–365.
- [29] P. Leitão, F. Restivo, ADACOR: a holonic architecture for agile and adaptive manufacturing control, *Comput. Ind.* 57 (2006) 121–130.
- [30] N. Gatti, F. Di Giunta, S. Marino, Alternating-offers bargaining with one-sided uncertain deadlines: an efficient algorithm, *Artif. Intell.* 172 (no. 8–9) (2008) 1119–1157.
- [31] S.S. Fatima, M. Wooldridge, N.R. Jennings, An agenda-based framework for multi-issue negotiation, *Artif. Intell.* 152 (no. 1) (2004) 1–45.
- [32] A. Giret, V. Botti, Engineering holonic manufacturing systems, *Comput. Ind.* 60 (no. 6) (2009) 428–440.
- [33] X. Gu, X. Jin, J. Ni, Y. Koren, Manufacturing system design for resilience, *Procedia Cirp* 36 (2015) 135–140.
- [34] P. Ferreira, S. Doltsinis, A. Anagnostopoulos, F. Pascoa, N. Lohse, A performance evaluation of industrial agents, *IECON 2013 - 39th Annu. Conf. IEEE Ind. Electron. Soc.*, (2013) , pp. 7404–7409.