

A Self-Organisation Model for Mobile Robots in Large Structure Assembly using Multi-Agent Systems

Spartak Ljasenko¹ • Niels Lohse¹ • Laura Justham¹ • Ivo Pereira² • Michael Jackson²

¹Wolfson School of Engineering, Loughborough University, Loughborough, UK
{S.Ljasenko, N.Lohse, L.Justham}@lboro.ac.uk

²EPSRC Centre for Innovative Manufacturing in Intelligent Automation, Loughborough, UK
{I.Pereira, M.R.Jackson}@lboro.ac.uk

Abstract. Mobile, self-organising robots are seen to be a possible solution to overcome the current limitations of fixed, dedicated automation systems particularly in the area of large structure assembly. Two of the key challenges for traditional dedicated automation systems in large structure assembly are considered to be the transportation of products and the adaptation of manufacturing processes to changes in requirements. In order to make dynamic, self-organising systems a reality, several challenges in the process dynamics and logistical control need to be solved. In this paper, we propose a Multi-Agent System (MAS) approach to self-organise mobile robots in large structure assembly. The model is based on fixed-priority pre-emptive scheduling and uses a blackboard agent as a central information source and to facilitate more common goal directed distributed negotiation and decision making between agents representing the different needs of products and available mobile resources (robots).

Keywords: Multi-agent systems, fixed-priority pre-emptive scheduling, large structure assembly

1 Introduction

The modern manufacturing industry is facing a number of challenges due to the global market's frequently fluctuating demands [1]. Traditional manufacturing systems are required to shift from mass production to mass customization [2]. The most common method of transporting products between manufacturing resources is by using conveyor belts [3]. This approach is not practical when the products are too

large, heavy or awkward to handle. In large structure assembly, products are commonly transported between manufacturing resources via cranes. This approach is very slow and expensive [4]. Moreover, fixed automation systems like the ElectroImpact E6000 [5] and HAWDE [6] have a fixed infrastructure which makes it difficult to change and adapt manufacturing processes. In our previous work [7], we showed that a mobile system is able to control product delivery times and adapt to fluctuations in demand better than a fixed automation system. By making manufacturing resources move to products as opposed to products to resources, the need to transport products is greatly reduced. Also, if the products are large enough, then the production rate may be scaled up by placing several mobile robots around one product. To our knowledge, no control model has been developed to facilitate scalable processes with the objective to minimise the Total Weighted Tardiness (TWT) of such a system.

Multi-Agent System (MAS) technology [8] is seen as a great tool for controlling various systems in real-time and dynamic environments. Because of the local problem solving, these systems should be able to deal with a high level of complexity, require less information exchange and respond quickly to unexpected events [9].

In this paper, we present a model to control the product flow in large structure assembly. This model can be considered somewhat centralised due to using a central blackboard agent as an information source. The blackboard agent does not control other agents. Instead, it helps them to exchange information. In comparison to a more decentralised system, on one hand such an agent increases the load on information exchange, but on the other it increases each agent's knowledge of the environment. As a result of that, the agents are able to make more informed decisions at the cost of requiring more communication between its entities [10].

The paper is organised as follows: in section 2, the fundamental structure of the mobile robot factory model underpinning this work is described. The development of a priority aging policy for our model is described in section 3. The architecture of the proposed MAS approach is described in section 4. A simulation model and some initial results are shown in section 5. Finally, section 6 draws conclusions from the initial testing of this approach and points out some directions for our further work.

2 Problem Description

The principle layout of the shop floor model underpinning this work is illustrated in **Fig. 1**. The fundamental assumption is that a number of products are fixed on static workstations and the mobile robots can freely travel on the floor between them. Each product therefore has a fixed location, capacity required work load, a due time and an associated tardiness cost. In the event of failing to meet the due time, the system is penalised with the tardiness cost of the tardy product multiplied by the tardiness time. The objective for our problem, minimising TWT, is shown in equation (1), where w_j is the tardiness cost, T_c is the completion time and T_d is the due time. $w_j = 0$, if the respective job j is completed on time.

$$\text{Min } \sum_{j \in J} w_j (T_c - T_d) \quad (1)$$

Many resources are allowed to work on the same product in order to achieve the required work rate. This is permitted, because the products are assumed to be large enough to enable a number of mobile robots to work simultaneously on them. The tasks are pre-emptive, meaning that mobile robots may temporarily pause one job in order to work on another one. Hence, the central challenge this shop floor model is to find best schedule for both products and mobile robots that minimise TWT for a given work load.

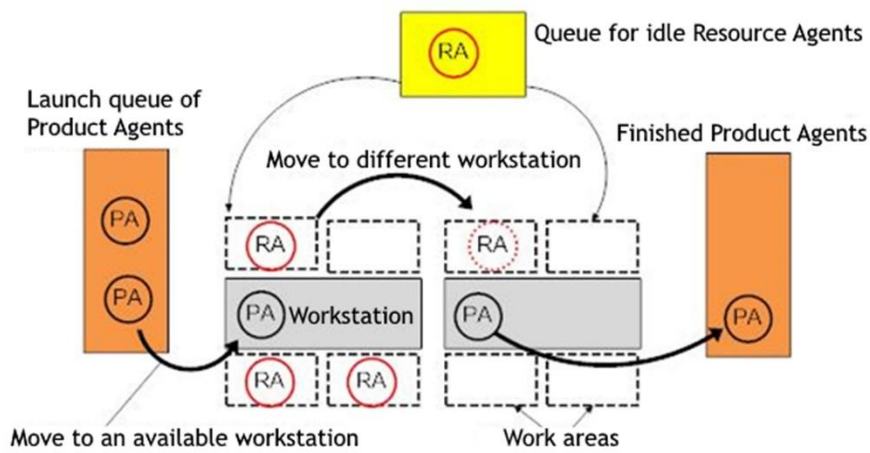


Fig. 1: Illustration of the principle shop floor organisation for mobile robot based assembly

In this paper, a MAS-based approach for the self-organisation of mobile robots in large structure assembly is proposed. The motivation behind this approach is to help mobile robots better cope with the complexity of deciding how to distribute themselves among products in fixed locations. Finding the optimal schedule is a NP hard problem even for relatively static environments as is the case in large structure assembly situations [11]. Following well-established architectural patterns for MAS in manufacturing [12], the objectives of each product instance have been represented through Product Agents (PA) and the capabilities of each mobile agent as Resource Agents (RA).

In order to achieve effective self-organisation between the agents, the decision making policies for the agent types as well as their communication protocols need to be defined. The scheduling policy that most closely matches the presented problem is fixed-priority pre-emptive scheduling that is commonly used in task scheduling for operating systems [13]. There, tasks are allocated to resources based on their priorities. A common issue with this is the starvation of low priority products when high priority products are constantly launched. In order to not starve low priority products, priority aging has been introduced. In our problem, the products must also meet set due times and therefore an appropriate priority aging policy is of high importance. The most commonly used priority aging policies are very basic and do

not consider any due times or tardiness costs. Therefore, a more sophisticated priority aging policy was required for this model.

The objective of this paper was, therefore, to modify the fixed-priority pre-emptive scheduling model to incorporate scalability and minimise TWT.

3 Development of the Priority Aging Policy

The integration of the priority aging policy into the task scheduling approach is the first key component of this paper. It is required to ensure that no low-priority products are starved. This is a common problem in fixed-priority pre-emptive scheduling where low priority tasks never get processed due to constantly arriving higher priority tasks. The challenge in this problem is not simply to avoid starving a product, but also to meet its due time. On some occasions, the arriving product work load requirements can be greater than the manufacturing capacity of the mobile system and it is therefore inevitable that some products will be tardy. To our knowledge, no suitable priority aging policy for our model exists, therefore we propose and analyse our own ones.

The first analysed priority aging policy was the linear policy (2) and the second one was the exponential one (3). In the two shown equations, $P(t)$ is priority in time, C_t is tardiness cost, t is the current time and d_t is the due time. If the due time has passed, the priority equals to the tardiness cost.

$$P(t) = Ct * \left(\frac{t}{dt}\right) \quad (2)$$

$$P(t) = Ct \frac{t}{dt} \quad (3)$$

4 Model Structure

The model uses three types of agents: product (PA), resource (RA) and blackboard (BA).

The BA is the first agent to be launched in the simulations. It is followed shortly by the PAs and RAs. The RAs send a message to the BA that includes their locations. The PAs send a message to the BA that includes their due times, location, tardiness costs and capacity. From this information, the BA is able to compile an initial schedule based on the priority aging of each product. The main purpose of the BA is to identify conflicts in schedules and notify the potentially tardy PAs about it. It is recognised that without the BA, it would be difficult, if not impossible, for the PAs to have sufficient knowledge to solve scheduling conflicts efficiently.

After the notifications are sent to the PAs, the BA listens to further messages from PAs. The further messages can be from newly launched PAs or changes to the schedule from existing ones.

The PAs send their location, due time, tardiness cost and capacity to the BA as soon as they are launched. They then listen for messages from the BA or other PAs. A message from the BA means that this particular product will be tardy unless the schedule is altered. This triggers the PA to negotiate a better schedule with other PAs. The PAs have a master-slave relationship with RAs. This means that the PAs do not need to negotiate with the resources. The negotiation only occurs among PAs once the initial schedule has been completed and tardy products identified.

Based on the priority ranking at any moment in time, the highest ranking products have the right to order just enough resources as is necessary to meet the due time. As a result of this, the lowest ranked products will always be starved if there are insufficient resources.

The RAs have a straightforward behaviour. Firstly, they notify the BA of their location and secondly they listen for orders from PAs. As the slave in the master-slave relationship with PAs, the RAs follow orders from products.

So far, the structure of the model and each agent's purpose has been described. Below, we describe all the interaction protocols that are used in the model.

Product – Blackboard

Product agents have two reasons to communicate with the blackboard agent. Firstly, all product agents send their locations, due times, tardiness costs and capacities to the blackboard.

Secondly, when the blackboard agent identifies that a product agent is tardy; it sends it a notification about it.

Product – Resource

This interaction is straightforward because product agents have a master-slave relationship with resource agents. Once product agents have agreed which resource agents each one will be occupying, they send an order to their resource agents to move to products and start working.

Product – Product

This interaction is the second key component in this model. As no priority aging policy can be expected to achieve an optimal result (due to the vast range of possible scenarios), this interaction serves as a corrective measure. This interaction is triggered by a notification from the BA. The intention of the protocol is to change the schedule in a way that reduces total weighted tardiness in the whole system.

When a product agent is notified about expected tardiness, it sends a message to all product agents that have due times after it. The message contains the product's due time, location, ID, expected tardiness cost and resource shortage. The shortage is the capacity that is not met by the due time in the initial schedule. The responding product agent compares the tardiness cost of the requesting agent to its own if the transaction is to be accepted. If the total weighted tardiness is lower as a result of accepting, then the responding agent sends an accepting message with the numerical value of how much the interaction will reduce the total weighted tardiness.

When several responses are received by the requesting agent, it means there is more than one favourable transaction available. In such a situation, the requesting agent accepts the first most beneficial response.

The reason why the request is not sent to product agents with earlier due times is because they only request as much as is required and any loss of resources would cause the whole product to be tardy for a much greater time than is reasonable.

5 Simulation and Initial Results

In order to analyse how various priority aging policies respond in different circumstances, a shop floor was modelled in NetLogo (version 5.2.1 [14]). As inputs to this model, there were four products with due times, capacities and tardiness costs. The simulations then followed the flowchart shown in **Fig. 4** and compared the results with different priority aging policies. Before launch, a product's priority was set to zero. The priority remained equal to the tardiness cost of a product if it was not completed by its due time.

The linear priority aging policy was found to produce poor results, as shown in **Fig. 2**. The basic nature of this aging policy resulted in frequent conflicts even during solving of some relatively simple problems. Products with low tardiness costs were often denied sufficient resources near their due times by products with higher tardiness costs and later due times. This problem occurred much more rarely with the exponential priority aging policy. In this policy, the priority of each product is increasing exponentially until its due time. This gives an advantage to products with low tardiness costs and earlier due times.

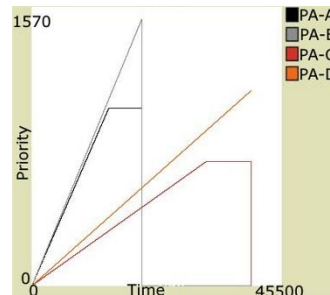


Fig. 2: The priority in time plot for the exponential priority aging policy

The exponential priority aging policy meets the intention of prioritizing products closer to their due times. The sharp increase in priority close to the due time caused products with low tardiness costs to reach high priority rankings and therefore increase the chances of receiving the necessary resources as seen in **Fig. 3a**. Further priority aging policies were considered, but they produced either poor or inconsistent results. The only difficulties arise when multiple products have close due times, as seen in **Fig. 3b**. In such cases, the product with the higher tardiness cost and later due time may cause the earlier product to starve. Alternatively, occasionally products with earlier due times and lower tardiness costs caused products with later due times and higher tardiness costs to be tardy.

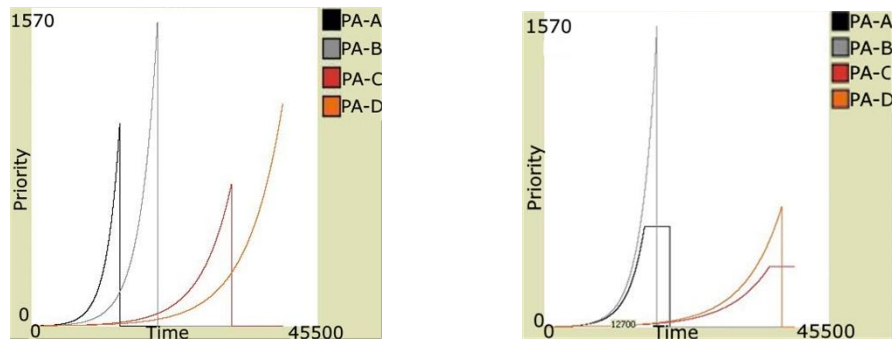


Fig. 3: The priority in time plot for the exponential policy in a) easy and b) difficult environments

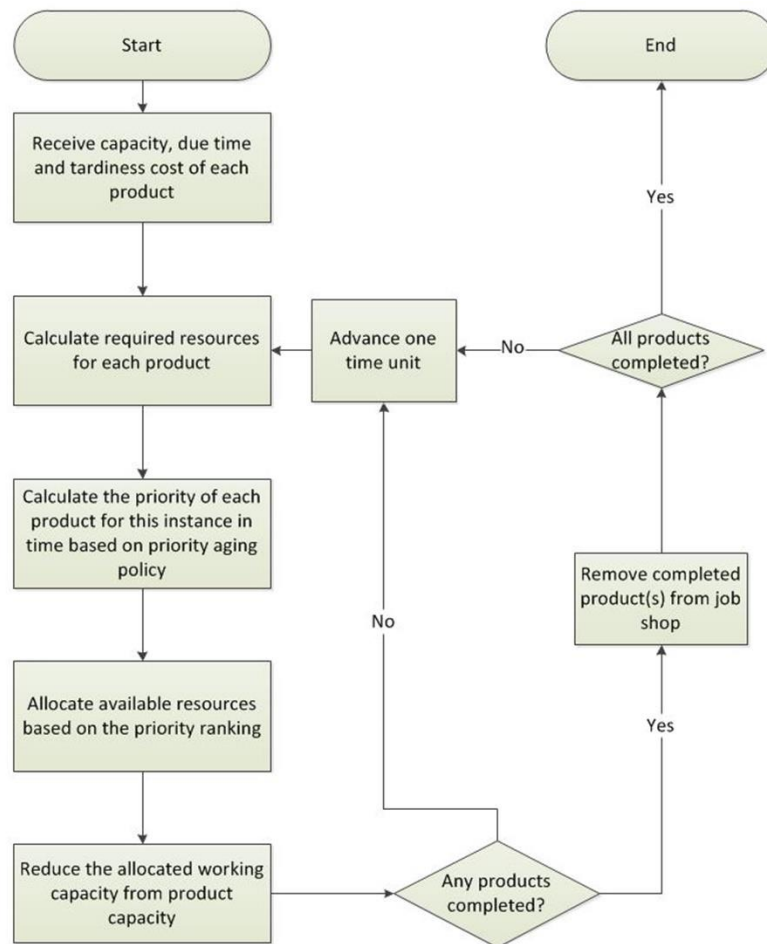


Fig. 4: The flowchart for the priority aging simulation

6 Conclusions and Further Work

In this paper a novel heterarchical scheduling approach using MAS was proposed. The model is intended for mobile robots to minimize TWT in large structure assembly. The novel contribution of the proposed approach is the scheduling for products that may have several resources processing them. The mobile robots are treated as resources that can be dynamically allocated to any of the existing products in order to complete them on time. Where completion is not possible, the lowest possible TWT is achieved. Although MAS were used, the proposed agent organisation is only partially distributed due to the use of a central blackboard agent (BA). The response of this model to disruptions and increasing the number of entities will be assessed after further testing. In our further work, we will compare this model to other MAS models.

Acknowledgments. The authors acknowledge support from the EPSRC Centre for Innovative Manufacturing in Intelligent Automation, in undertaking this research work under grant reference number EP/IO33467/1.

References

- [1] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution," *Comput. Ind.*, vol. 66, pp. 99–111, 2015.
- [2] P. Leitão and F. Restivo, "ADACOR: A holonic architecture for agile and adaptive manufacturing control," *Comput. Ind.*, vol. 57, pp. 121–130, 2006.
- [3] S. Zhang and X. Xia, "Optimal control of operation efficiency of belt conveyor systems," *Appl. Energy*, vol. 87, no. 6, pp. 1929–1937, 2010.
- [4] Z. Yuguang, X. Kai, and S. Dongyan, "Computer-Aided Design Assembly unit partitioning for hull structure in shipbuilding," *Comput. Des.*, vol. 45, no. 12, pp. 1630–1638, 2013.
- [5] D. Remley, J. Rediger, P. Haworth, and R. Holden, "Slug Rivet Machine Installs 16 Rivets Per Minute Drill-Rivet-Shave," 2009.
- [6] S. Smith, I. Moore, and T. Jackson, "2004-01-2806 HAWDE Five Axis Wing Surface Drilling Machine," *Assembly*, no. 724, 2004.
- [7] S. Ljasenko, N. Lohse, and L. Justham, "A Comparison of the Manufacturing Resilience between Fixed Automation Systems and Mobile Robots in Large Structure Assembly," in *49th CIRP Conference on Manufacturing Systems*, 2016, vol. 0.
- [8] M. Wooldridge, *An Introduction to MultiAgent Systems*. 2002.
- [9] M. Mes, M. van der Heijden, and J. van Hillegersberg, "Design choices for agent-based control of AGVs in the dough making process," *Decis. Support Syst.*, vol. 44, no. 4, pp. 983–999, 2008.
- [10] D. D. Corkill, "Blackboard and Multi-Agent Systems & the Future," *Collab. Softw.*, January 2003.
- [11] L. Wan and J. Yuan, "Single-machine scheduling to minimize the total earliness and tardiness is strongly NP-hard," *Oper. Res. Lett.*, vol. 41, no. 4, pp. 363–365, 2013.
- [12] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 979–991, 2009.
- [13] C. Leet, J. Hahn, Y. Seot, C. Y. Parkq, M. Lee, and C. S. Kim, "Enhanced Analysis of Cache-related Preemption Delay in Fixed-priority Preemptive Scheduling *," *IEEE Trans. Comput.*, pp. 187–198, 1997.
- [14] U. Wilensky, "NetLogo Home Page," 2016. [Online]. Available: <https://ccl.northwestern.edu/netlogo/>. [Accessed: 13-Jul-2016].