Brazilian Journal of Operations & Production Management

ISSN 2237-8960 (Online)

ABEPRO ASSOCIAÇÃO BRASILEIRA DE ENGENHARIA DE PRODUÇÃO

RESEARCH PAPER

Agent-based dynamic scheduling model for product-driven production

João Thiago de Guimarães Anchieta e Araujo Campos[1], Jana Blumelova[1], Herman Augusto Lepikson[1], Francisco Gaudêncio Mendonça Freires[1]

[1] Industrial Engineering Program (PEI), Federal University of Bahia, Salvador, Brazil.

How to cite: Campos, J.T.G.A.A., Blumelova, J., Lepikson, H.A. et al. (2020), "Agent-based dynamic scheduling model for product-driven production", *Brazilian Journal of Operations & Production Management*, Vol. 17, No. 04, e20201075. https://doi.org/10.14488/BJOPM.2020.044

ABSTRACT

Goal: This research provides specific solution for dynamic scheduling of product-driven production with unique level of detail and original architecture.

Design / Methodology / Approach: Design process of scheduling problem-solving MAS is divided into three steps: agent encapsulation of entities participating in scheduling, including concept of agents and responsibilities they assume, system architecture and topology of the agents network, detailed design of decision scheme of individual agents.

Results: Production processes take place in dynamic environment and have to react to numerous real-time events, hence reschedule the production by a new design and implement agent-based model in order to solve dynamic flexible job shop scheduling problem in product-driven production environment.

Limitations of the investigation: Designed model counts with simple agents behaving on condition-action rules. These agents could be replaced by more sophisticated types of agents such as utility-based or learning agents. Also the implemented coordination mechanism ensuring global view on the scheduling problem is rather simple.

Practical implications: Via simulations of realistic production scenarios it was expected to prove an applicability of specific solution of how bringing the intelligence to the lower levels of control system may be used in dynamic scheduling. Based on elaboration of theoretical basis, principles were identified as suitable or widely used in design of such model.

Originality / Value: This research provides specific real-time architecture for a multi-agents dynamic scheduling of product-driven production with unique level of detail and scenarios analysis.

Keywords: Product-Driven Production; Dynamic Scheduling; Agent-Based Production Control; Flexible Job Shop Scheduling; Agent-Based Modelling.

INTRODUCTION

Production control is usually closely associated with production planning where the desirable outputs of products are set. Knowing the outputs, the ordering phase for resources is ensured by supply chain management. Once all the necessary resources are available they need to be allocated. It is a purpose of scheduling to optimize work and workloads in production process (Fanjul-Peyro et al., 2017; Laili et al., 2020; Helo et al., 2019). Scheduling problem in manufacturing has been historically approached as a static problem. Static schedules are fixed plans; they are optimal or near-optimal solutions of the scheduling problem. All the necessary

Financial support: None
Conflict of interest: The authors have no conflict of interest to declare
Corresponding author: joaothiagocampos@gmail.com
Received: 18 Aug 2020.
Approved: 26 Aug 2020.
Editor: Osvaldo L. G. Quelhas

This is an Open Access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

information is a priori known and it is assumed that proposed sequence of tasks is to be successfully executed. However, these schedules are not able to react to any changes.

Production control including the dynamic scheduling can be centralised or distributed. Historically the centralised approach, often hierarchical, has been prevailing in practice (Zhang et al., 2019). In the past a customer insisted on low cost, later a quality turned into a concern. When the cost and quality became generally reasonable, a speed of delivery started to be considered. Nowadays, the preferences of customers is customizability, then competitive industry now seeks more for robust and adaptable control system rather than optimal solution. Concepts such as Holonic Manufacturing Systems, Product-driven Control Systems, Intelligent Product, Intelligent Manufacturing Systems and Agent-based Manufacturing that have been proposed to design future manufacturing systems. High amount of research in these areas have been carried out lately (Borangiu et al., 2014; Ge et al., 2017; Derigent et al., 2020; Xiong and Fu, 2018).

In product-driven production control an inteligente, 'product in-progress' communicates with resources (typically the machines and workers), environment and other products. For instance it can configure the machines participate in production scheduling and help to properly react to any job-related or resource-related events. One of the approaches how to achieve such behaviour is to consider intelligent product as well as any resource or other physical or functional participating element an agent. An agent is defined as an entity that, by sensing and acting upon its environment, tries to fulfil a set of goals in a complex dynamic environment (Jürgen Müller, 1997). An agent is a hardware or (more usually) software-based computer system that has the following characteristics: autonomy, social ability, reactivity and pro-activeness (Moyaux et al., 2006). Negotiation and cooperation between agents are typical behaviour in multi-agent systems. Thus, this research proposes an agent-based dynamic scheduling with a real-time response of manufacturing failures. An archictecture is proposed and validated in a real product-driven environment.

Agent-based dynamic scheduling in manufacturing is a specific approach to the specific part of distributed production control. In product-driven production control, the intelligent product demonstrates some identical features as the agent by definition. It interacts with its environment and it is able to make local decisions or participate in global decision-making process. Also it operates in complex dynamic environment. Thus, it seems to be highly suitable to represent an intelligent product by an agent.

Then, the main goal of this research is to design and implement an agent-based model in order to solve dynamic scheduling problem in product-driven production environment. Simulations of realistic production scenarios shall present specific solution of how bringing the intelligence to the lower levels of control system may be used in dynamic scheduling.

METHOD

Design process of scheduling problem-solving MAS is divided into three steps:

- agent encapsulation of entities participating in scheduling, including concept of agents and responsibilities they assume;
- system architecture and topology of the agents network;
- detailed design of decision scheme of individual agents.

Agent Encapsulation

Recalling three basic types of agents in production control according to PROSA - product, resource, order, staff and architecture -, following entities participating in product-driven production scheduling were identified and are considered further:

a) Product agent related – database of alternative process plans;

For proper scheduling where tasks are performed on products-in-progress in correct order by correct resource with correct specifications a product model is needed. Product model basically says how to fabricate certain product with basic informations as task performed, order of tasks, specifications and resources. Alternative process plans are stored in database of alternative process plans (DAPP).

b) Resource agent related;

Resource Agent (MA) is responsible for correct performing of tasks by resource as well as for participation in scheduling process. Each resource in production is represented via one RA. Due to two essential spaces of dynamic scheduling, RA consists of two basic parts – scheduling and execution agent.

Execution Agent of RA (REA) is responsible for control of the respective resource to perform tasks on products-in-progress correctly.

Scheduling Agent of RA (RSA) manages local schedule of the resource. For scheduling purposes RSA also needs to be notified of resource-related events by REA and inform about it concerned agents.

c) Order agent related – intelligent products (products-in-progress).

Intelligent product agent (IPA) is core object of production control (including scheduling) in product-driven production. Previously defined agents (DAPPA, RA) are permanently part of the system. IPA is created when an order of customer is recognized by manufacturing system (order arrival). Consequently, IPA is discarded when the order is delivered. Intelligent products as main control entities are responsible for scheduling as well as execution. RA is subordinated to IPA both in scheduling and physical process execution. Therefore, IPA consists of two basic parts – scheduling and execution agent.

Scheduling Agent of IPA (IPSA) is main scheduling entity. It breakdowns job into tasks. For correct decomposition of job into tasks it needs product model stored in DAPP. It is responsible for assignment of resources to tasks and also it participates in global level of assignment of resources to tasks. Execution Agent of IPA (IPEA) is needed for control of physical process execution. It is responsible for ensuring that tasks are performed on products-in-progress correctly, in correct order and on time.

Product state model mentioned in previous paragraphs is information on tasks to be performed in order to deliver job. Product state model is up-to-date information on job.

Architecture

Architecture described follows to the greatest possible extent work of Borangiu et al. (2014). Their architecture is suitable because it is based on holonic principles, uses PROSA and deals with dynamic scheduling in product-driven production.

Arrival and scheduling parts of architecture described are illustrated via swim lane diagram in Figure 1.

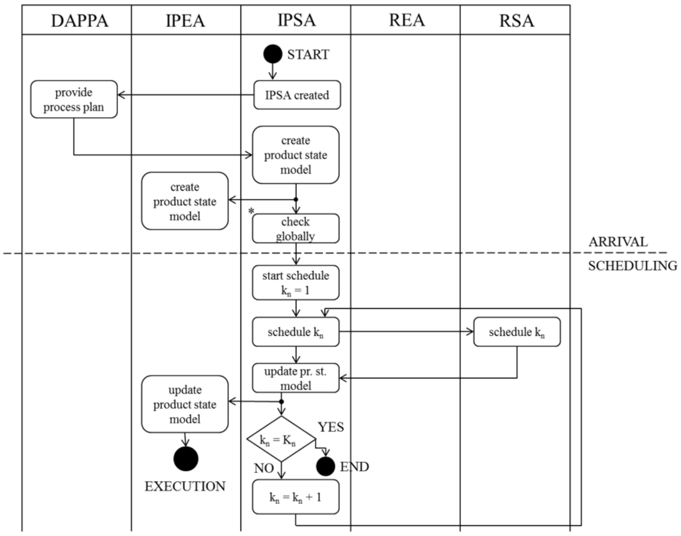

Figure 1. Architecture - arrival and scheduling parts.

IPEA routes product-in-progress to scheduled resource and requests tasks to be performed. REA verifies with its respective RSA whether the request is in compliance with its local schedule. When all the tasks have been successfully performed, IPA proceeds to delivery part. This includes waiting for other jobs from the same order and discard of IPA agent. Execution and delivery parts of architecture described in this paragraph are illustrated via swim lane diagram in Figure 2.

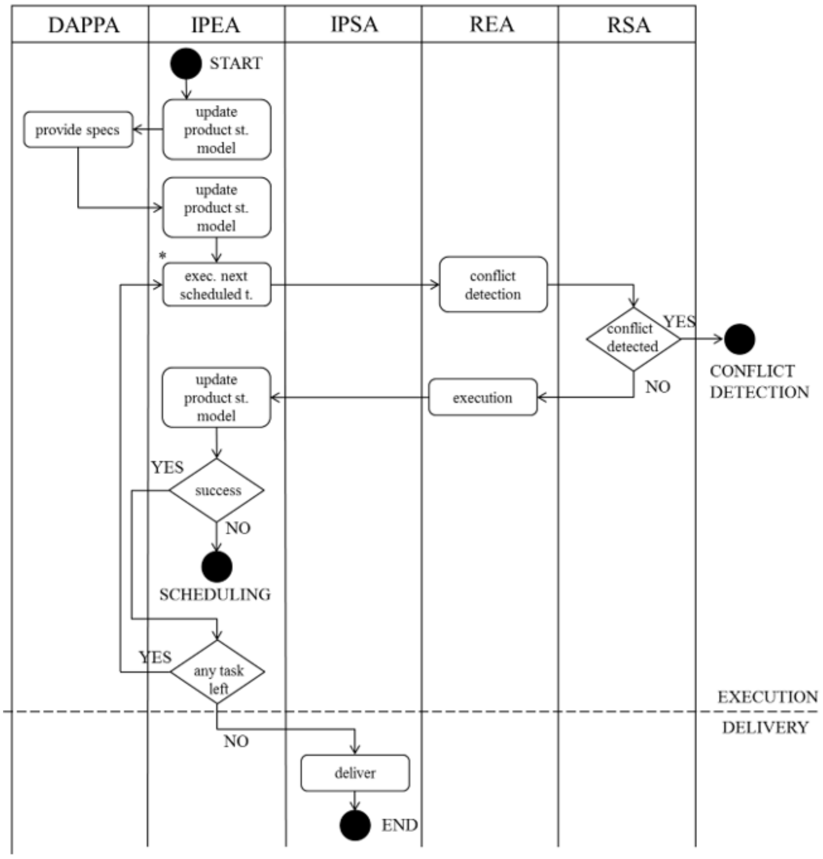

Figure 2. Architecture - execution and delivery parts.

However, when RSA detects conflict with its local schedule, it informs REA and IPSA about the situation. REA in this case must not start an execution. IPSA is responsible for resolving this conflict. Once the conflict is resolved, execution continues. Execution conflict detection part of architecture described in this paragraph is illustrated via swim lane diagram in Figure 3.

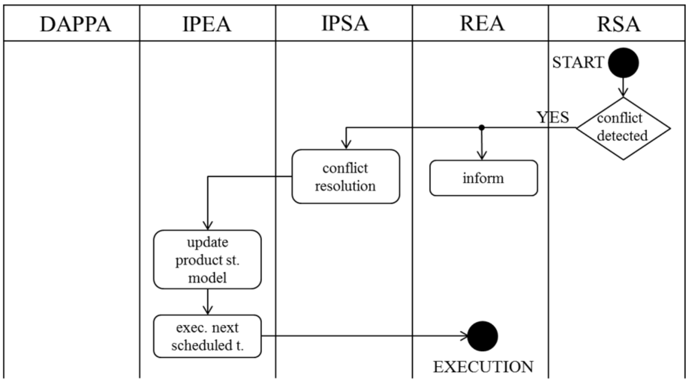

Figure 3. Architecture - execution con ict detection part.

Decision Scheme of an Agent

Agents proposed behave based on condition-action rule. Their rules are strongly related to previously defined interaction protocols.

DISCUSSION AND RESULTS

Modeled product scenario

Model Factory is an educational environment located in SENAI CIMATEC, Salvador, Brazil, built with cooperation of McKinsey Company. This mimic factory aims to capture all aspects of real factory environment in smaller scale and more controlled way which allows for better mastering of lean concepts.

In modelled production scenario, products from product family of pneumatic cylinders are produced. Produced pneumatic cylinders differ in length (stroke), diameter (bore) and male/female threaded piston rod end for mounting.

Pneumatic cylinder consists of cylinder barrel (body), cylinder head, cylinder cap (base), piston, piston rod with screw thread (male or female), piston, seal glands and seals. In order to simplify validation scenario, fabrication of only cylinder body and piston rods is considered. There are two raw materials used: tube for body production and bars for rod production.

There are three workstations considered in this production process – sawing, turning and cleaning. Firstly, both body and rod are cut to desired length at sawing workstation. Then rod continues to turning workstation where relevant thread is created. Both body and rod must be cleaned after machining process (sawing and turning).

In modelled production scenario, products from product family of pneumatic cylinders are produced. Produced pneumatic cylinders differ in length (stroke), diameter (bore) and male/female threaded piston rod end for mounting. All possible products are listed in Table 1. This simplification results in limited number of possible jobs as they are listed in Table 2.

Table 1. Scenario of fabricated products

No	Length	Diameter	Rod end
1	100 mm	25 mm	Male
2	100 mm	25 mm	Female
3	100 mm	40 mm	Male
4	100 mm	40 mm	Female
5	150 mm	25 mm	Male
6	150 mm	25 mm	Female
7	150 mm	40 mm	Male
8	150 mm	40 mm	Female

Table 2 – Scenario – jobs

No	Part	Length	Diameter	Rod end	Cleaning
1	Body	100 mm	25 mm	-	Yes
2	Rod	100 mm	-	Male	Yes
3	Rod	100 mm	-	Female	Yes
4	Body	100 mm	40 mm	-	Yes
5	Rod	100 mm	-	Male	Yes
6	Rod	100 mm	-	Female	Yes
7	Body	150 mm	25 mm	-	Yes
8	Rod	150 mm	-	Male	Yes
9	Rod	150 mm	-	Female	Yes
10	Body	150 mm	40 mm	-	Yes
11	Rod	150 mm	-	Male	Yes
12	Rod	150 mm	-	Female	Yes

Source: Elaborated by the author.

There are three workstations considered in this production process – sawing, turning and cleaning. Firstly, both body and rod are cut to desired length at sawing workstation. Then rod continues to turning workstation where relevant thread is created. Both body and rod must be cleaned after machining process (sawing and turning). Time specifications of these tasks per unit are provided in Table 3.

Table 3 – Scenario – job time specifications

Job no.	Sawing [s]	Turning [s]	Cleaning [s]
1	4	-	3
2	2	2	1.8
3	2	2.5	2.2
4	4	-	3
5	2	2	1.8
6	2	2.5	2.2
7	4.2	-	3.2
8	2	2	1.8
9	2	2.5	2.2
10	4.2	-	3.2
11	2	2	1.8
12	2	2.5	2.2

Each workstation consists of two identical pieces of equipment. R_1 and R_2 are sawing machines, R_3 and R_4 turning machines and R_5 and R_6 cleaning devices. This redundancy is a need in robust manufacturing systems in order to ensure at least partial continuity of production in case of machine breakdown. This issue is shown in Figure 4.

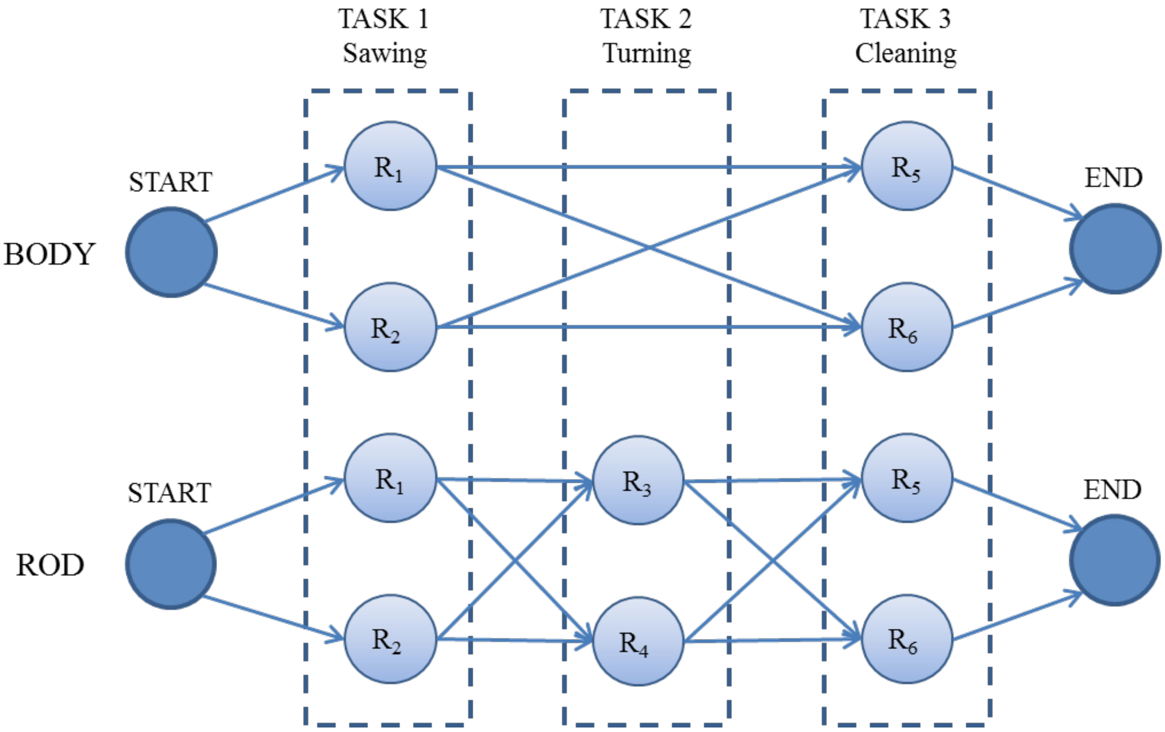

Figure 4 - Process plan for body and rod.

Scenario Analysis

Model designed is implemented in AnyLogic software in order to validate its functions via simulation of different scenarios. Validation seeks to prove correct function of the model. At the same time, correct functioning also implies applicability of agent-based approach to dynamic scheduling of product-driven production.

AnyLogic is a simulation software combining three modelling methods – system dynamics, discrete event and agent-based modelling. It was chosen due to its user-friendliness, limited free license for academic . Simulation of designed model is built upon population of agents representing IPA where those are created and discarded dynamically. Permanent agents (DAPPA and RA) are simulated via discrete event patterns used in process-centric modelling.

Random number generators and statistics distribution were used to represent unpredictability of job arrivals (arrival date, quantity, product type and due date) and uncertainty of each processing time.

Simulations were performed for 200 jobs arriving in each scenario. Every resource was subject to breakdown (simulated with use of MTTF – Mean Time to Failure and MTTR – Mean Time to Repair) and any order could be a rush order. However, schedules of only the first 26 jobs in each scenario are provided. In this manner easier orientation in results is achieved.

Firstly, zero scenario was simulated for verification purposes. This scenario does not count with presence of real-time events. Therefore it is called 'Static' in following figures. Upper part in following figures illustrates job arrival date. The length of stripes is given by the interval before the next job arrives. Due to the structure of pneumatic cylinders, jobs arrive in couple with equal arrival date, quantity and due date. Jobs are numbered in accordance with order of their arrival where the body part always comes first. As a consequence of use of Java-based AnyLogic, numbering is zero-based. Provided schedules are Gantt charts with model minutes as x axis units. Each stripe corresponds to one resource.

There are no rush orders nor any machine breakdown in zero scenario. This comparison enables to recognize easily when the rescheduling occurs and how the newly created schedule differs from the previous schedule. Schedule for zero scenario is shown in Figure 5.

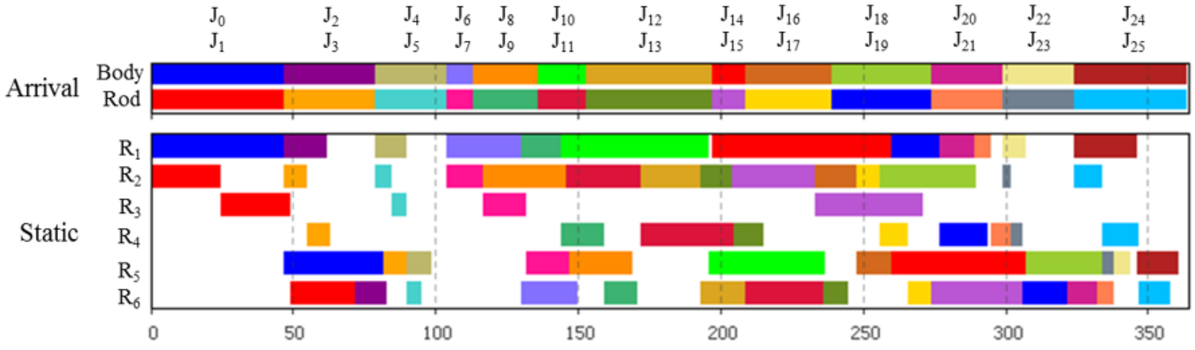

Figure 5. Results - zero scenario (static) schedule.

Secondly, three different scenarios are simulated where rescheduling occurs. Scenario no. 1 counts with machine breakdowns (resource-related real-time event) but no rush orders. In Figure 6, schedule for the first scenario is provided. Vertical black line marks the moment when resource R_2 broke down and thus triggered rescheduling. At the moment of breakdown, R_2 was performing sawing on job J16. From the demanded quantity of 230 bodies, 83 were successfully cut. During the work on 84-th piece of body, R_2 broke down and produced a scrap. Therefore, a new sawing task with quantity of 147 bodies must be scheduled. Due to EDF policy, J_{16} is scheduled to the nearest time slot on sawing machine

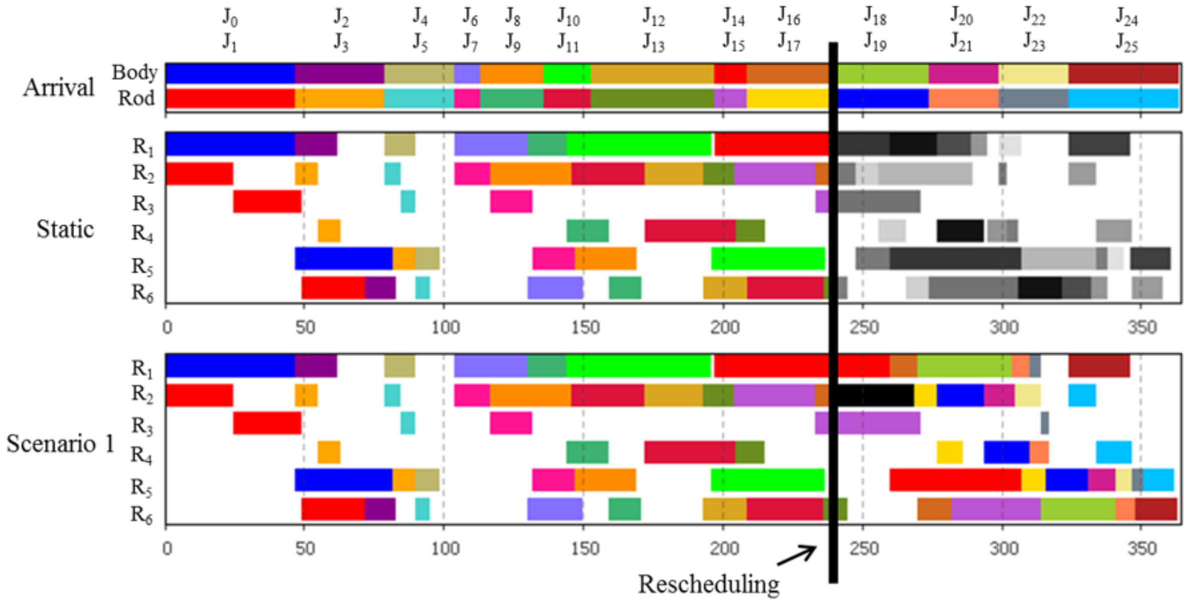

Figure 6. Results - scenario no. 1 with machine breakdown.

Scenario no. 2 in Figure 7 counts with rush orders (job-related real-time event) but no machine breakdowns. Vertical black line marks the moment when rush order consisting of J_{18} and J_{19} arrives. This arrival triggers rescheduling so that J_{18} and J_{19} get ahead the other jobs.

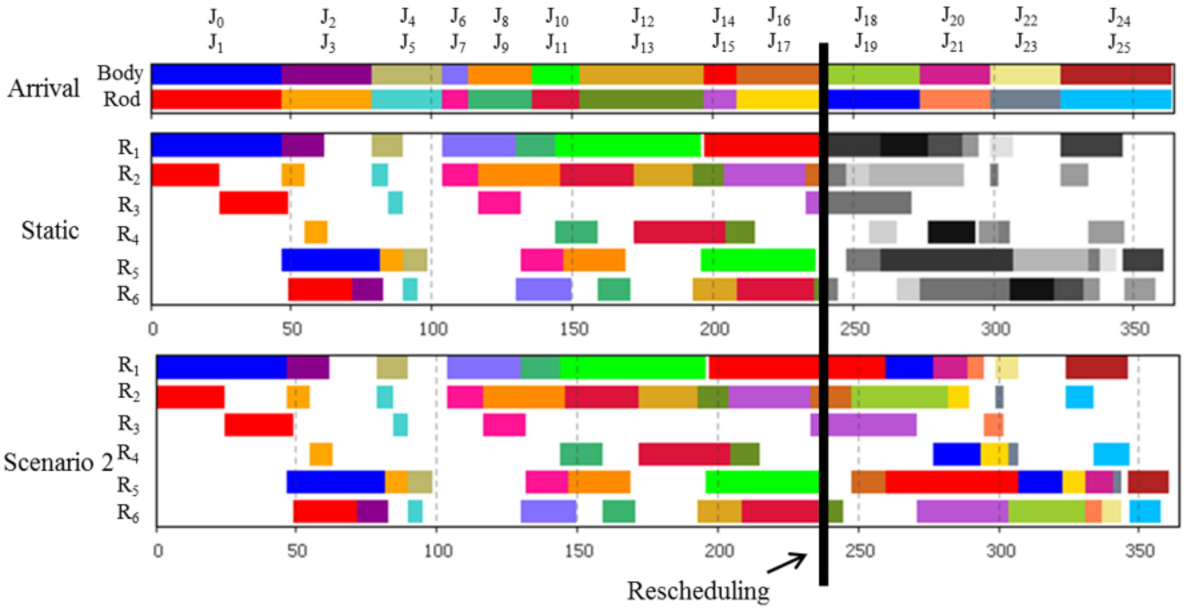

Figure 7. Results - scenario no. 2 with rush order.

Scenario no. 3 counts with both rush orders and machine breakdowns. In Figure 8, schedule for the third scenario is provided. Vertical black line marks the moment when both the resource R_2 breaks down and rush order consisting of J_{18} and J_{19} arrives. These events turn previously created schedule ('Static') infeasible and trigger rescheduling.

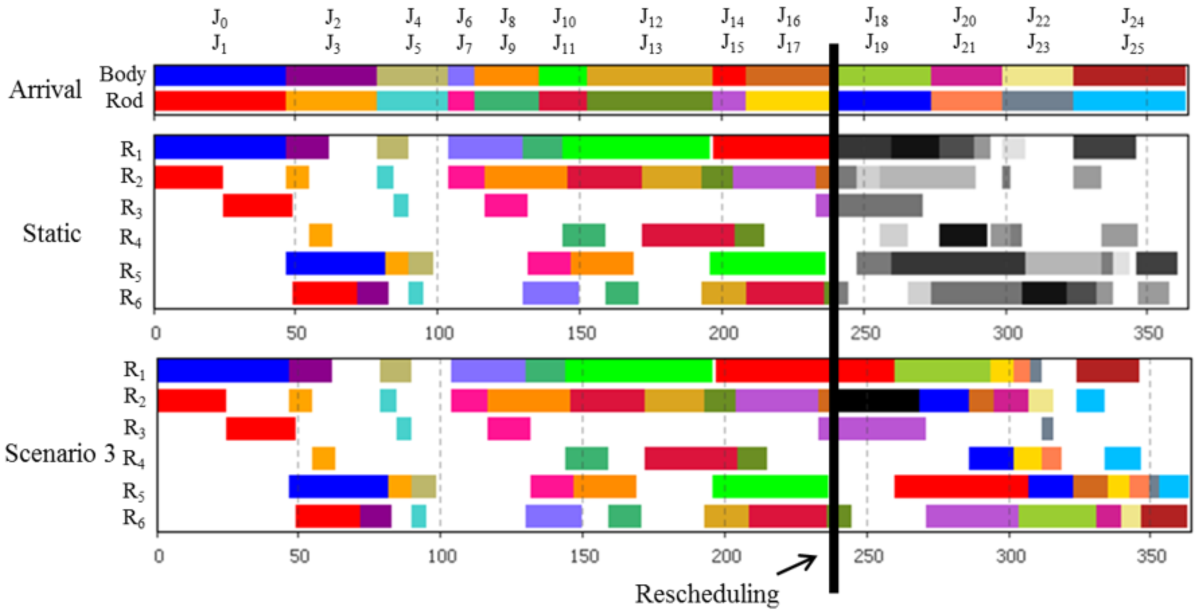

Figure 8. Results - scenario no. 3 with machine breakdown and rush order.

A disputable issue might be whether it is necessary to separate execution and scheduling parts of agents at all. In real deployment these agents could share memory and processing unit as well as agent communication module. However this decision whether to separate or not can be made on a case-by-case basis. In this manner breakdown of scheduling part does not directly compromise execution part and vice versa.

CONCLUSION

Implementation of designed model and simulation results proved correct functioning of model. A part of realistic production scenario where pneumatic cylinders are produced was used for verification and validation. Scenario corresponds to make-to-order production of some SME. As the customers are unique, orders arrive dynamically and unpredictably. There are two types of simulated real-time events: machine breakdown and rush order.

This research provides specific solution for dynamic scheduling of product-driven production with unique level of detail and original architecture. It is a scalable and flexible. As such it contributes to area of dynamic scheduling of product-driven production suffering from lack of research. Also it is an example of concurrent scheduling and execution. However, there are also some limitations which may be treated in future research. Designed model counts with simple agents behaving on condition-action rules. These agents could be replaced by more sophisticated types of agents such as utility-based or learning agents. Also the implemented coordination mechanism ensuring global view on the scheduling problem is rather simple. Its optimization criterion is makespan. A future research is suggested rather using of due-date based optimality criteria such as minimization of number of tardy jobs. Due-date based criteria helps to avoid delay of order delivery which leads to the need of customer compensations and loss of goodwill.

REFERENCES

Borangiu, T., Raileanu, S., Trentesaux, D. et al. (2014), "Distributed manufacturing control with extended CNP interaction of intelligent products", *Journal of Intelligent Manufacturing*, Vol. 25, No. 5, pp. 1065-75. http://dx.doi.org/10.1007/s10845-013-0740-3.

Fanjul-Peyro, L., Perea, F. and Ruiz, R. (2017), "Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources", *European Journal of Operational Research*, Vol. 260, No. 2, pp. 482-93. http://dx.doi.org/10.1016/j.ejor.2017.01.002.

Ge, X., Yang, F. and Han, Q.-L. (2017), "Distributed networked control systems: A brief overview", *Information Sciences*, Vol. 380, pp. 117-31. http://dx.doi.org/10.1016/j.ins.2015.07.047.

Helo, P., Phuong, D. and Hao, Y. (2019), "Cloud manufacturing – Scheduling as a service for sheet metal manufacturing", *Computers & Operations Research*, Vol. 110, pp. 208-19. http://dx.doi.org/10.1016/j.cor.2018.06.002.

Jürgen Müller, H. (1997), "Towards agent systems engineering", *Data & Knowledge Engineering*, Vol. 23, No. 3, pp. 217-45. http://dx.doi.org/10.1016/S0169-023X(97)00013-X.

Laili, Y., Lin, S. and Tang, D. (2020), “Multi-phase integrated scheduling of hybrid tasks in cloud manufacturing environment”, *Robotics and Computer-Integrated Manufacturing*, Vol. 61.

Moyaux, T., Chaib-Draa, B. and D’Amours, S. (2006), "Supply chain management and multiagent systems: An Overview. Multiagent based Supply Chain Management", *Springer Berlin Heidelberg*, Vol. 4, pp. 1-27.

Derigent, W., Cardin, O. and Trentesaux, D. (2020), "Industry 4.0: Contributions of holonic manufacturing control architectures and future challenges", *Journal of Intelligent Manufacturing*. http://dx.doi.org/10.1007/s10845-020-01532-x.

Xiong, W. and Fu, D. (2018), "A new immune multi-agent system for the flexible job shop scheduling problem", *Journal of Intelligent Manufacturing*, Vol. 29, No. 4, pp. 857-73. http://dx.doi.org/10.1007/s10845-015-1137-2.

Zhang, J., Ding, G., Zou, Y. et al. (2019), "Review of job shop scheduling research and its new perspectives under Industry 4.0", *Journal of Intelligent Manufacturing*, Vol. 30, No. 4, pp. 1809-30. http://dx.doi.org/10.1007/s10845-017-1350-2.

Authors contribution: João Thiago de Guimarães Anchieta e Araujo Campos: Paper writing, review and structure analysis to submission. Jana Blumelova: Paper writing, data collection and analysis. Herman Augusto Lepikson: Project manager, research contributions and technical evaluation. Francisco Gaudêncio Mendonça Freires: Research manager, technical revision, review and structure analysis to submission.