



HAL
open science

Multi-agent systems negotiation to deal with dynamic scheduling in disturbed industrial context

Tsegay Tesfay Mezgebe, Hind Bril-El Haouzi, Guillaume Demesure, Rémi Pannequin, André Thomas

► **To cite this version:**

Tsegay Tesfay Mezgebe, Hind Bril-El Haouzi, Guillaume Demesure, Rémi Pannequin, André Thomas. Multi-agent systems negotiation to deal with dynamic scheduling in disturbed industrial context. Journal of Intelligent Manufacturing, 2020, 31 (6), pp.1367-1382. 10.1007/s10845-019-01515-7. hal-02384288

HAL Id: hal-02384288

<https://hal.science/hal-02384288>

Submitted on 28 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-agent Systems Negotiation to deal with Dynamic scheduling in Disturbed industrial context

Tsegay Tesfay MEZGEBE^{1,2*}, Hind BRIL EL HAOUZI¹, Guillaume DEMESURE¹, Remi PANNEQUIN¹, Andre THOMAS¹

Abstract It is now accepted that using multi-agent systems (MAS) improve the reactivity to treat perturbation(s) within flexible manufacturing system. Intelligent algorithms shall be used to address these perturbation(s) and all smart decision entities within their environment have to continuously negotiate until their common and final goal is achieved. This paper proposes a negotiation-based control approach to deal with variability on a manufacturing system. It has initially formulated and modeled an environment in which all contributing entities or agents operate, communicate, and interact with each other productively. Then after, simulation and applicability implementation experiments on the basis of full-sized academic experimental platform have been conducted to validate the proposed control approach. Product and resource entities negotiate considering different key performance measures in order to set best priority-based product sequencing. This has been done with expectations that the applicability of the negotiation-based decision-making will be more adaptable to deal with perturbation(s) than another alternative decision-making approach called pure reactive control approach. The result showed that negotiation among the decisional entities has brought significant improvement in reducing makespan and hence conveyed better global performance of a manufacturing system.

Keywords Negotiation, Control Protocol, Multi-agent System, Intelligent Decision, Distributed Reactive, Makespan

1. Introduction

In today's volatile market, manufacturing industries are facing severe pressure to deal with dynamic variability of their systems. This variability mainly includes the control of lead times, due dates, and work-in-progress inventory impacted by queue times. Meanwhile, the Industry 4.0 paradigm, which gears towards individualized customer requirements such as higher product variety and smaller batches, is urging these industries to focus on technological advancement. Such advancement could be achieved if their products, resources, and other components are integrated as multi-agent systems (MAS). Consequently, these industries are continuously searching for new functional control systems that enable them to quickly respond to a growing set of perturbation(s) and hence deliver products and services to customers in accordance with specified requirements. As it is shown in Fig. 1, these control approaches could include proactive control, reactive control, or interactive control approaches (El-Haouzi, 2017). An increasing interconnection and networking of computers made the third control approach to associate with environmental infrastructure including protocols for agents to communicate³ and to interact. The communication and interaction realize the capability of smart decision entities⁴ to create adaptive and flexible manufacturing system (FMS) such that these entities within the system not only interact to each other but also adapt and/or learn from their environment (Valckenaers and Van Brussel 2016). For instance, when manufacturing schedules acquire a reputation for rapid invalidity, using dynamic rescheduling methods to overcome such unexpected event is proved to be crucial. Intelligent algorithms shall be used to dynamically overcome the unexpected events where all decisional entities within the environment have to continuously negotiate for final decision. This could be achieved if the decision entities use different message exchanging types such as proposing, accepting, rejecting, counter proposing etc.

¹Université de Lorraine, CRAN, UMR 7039, Campus Sciences, BP 70239, 54506, Vandœuvre-lès-Nancy cedex, CNRS, CRAN, UMR7039, France

²Ethiopian Institute of Technology-Mekelle, Mekelle Univeristy, Ethiopia

*Corresponding author (Email tsegay-tespay.mezgebe@univ-lorraine.fr, telephone number +33615478914)

³Communication protocols enable agents to exchange and understand messages and interaction protocols enable agents to have either structured or unstructured conversations (Weiss, 1999).

⁴Decision entities (or agents) are autonomous and cooperative components within manufacturing system with capability to show physical and informational communication behaviors and hence make decisions(El Haouzi et al. 2008).

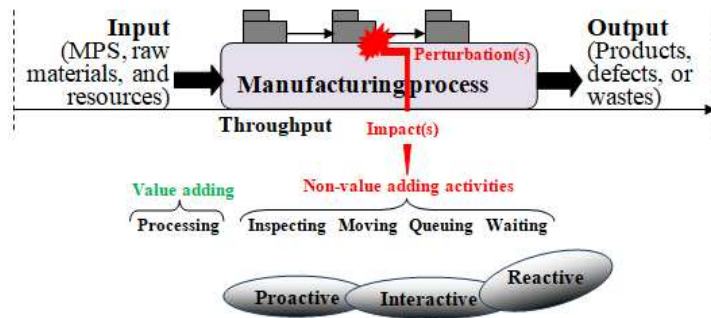


Fig. 1 Functional control approaches within Manufacturing Systems

In addition to manufacturing industries, researchers have also showed the interest of designing and employing different control approaches and heuristics for multi-agent manufacturing systems (Shen et al. 2006). Wang et al. (2018), for example, have proposed metaheuristic algorithm to solve complex optimization problems. Ouelhadj and Petrovic (2009) have also presented the potential of multi-agent based scheduling in comparison to meta-heuristics and other artificial intelligence methods. From their comparative study, these and other researchers have concluded that multi-agent system is a very promising area of current and future researches in dynamic scheduling.

In this paper, negotiation-based reactive control approach (NRC) is proposed to set best priority-based sequential manufacturing process with final objective to minimize makespan. Makespan in manufacturing process is the time difference between production started and production completed for a sequence of jobs or tasks (Skorin-Kapov and Vakharia, 1993). If a shop floor (it could be a job shop, open shop, or flow shop) is not able to reduce this makespan, customer orders received will not be shipped on their planned delivery time. Thus, in order to create an end-to-end alignment between a manufacturing decision-making and the strategic objective of an organization, minimizing makespan by providing efficient planning of tasks to resources within the manufacturing system is significantly required. Product and resource agents in the proposed control approach cooperate considering intention of products to obtain near optimal performance. Meanwhile, product and resource agents, in applicability implementation experiment, are made to broadcast information among each other with complete involvement of Radio Frequency Identification (RFID) and Production Line Controller (PLC) technologies.

The novelty of this work is both in theory and in experimentation. It initially models and describes an environment in which all contributing entities operate, communicate, and interact with each other productively. After this, the model is validated with simulation and applicability implementation experiments on an academic full-sized application platform called TRACILOGIS platform⁵. MATLAB and Python simulation experiments on the basis of the platform have been used to test how the negotiation-based control approach could affect the priority-based sequencing. JADE⁶ is used to test the applicability implementation of the proposed control approach on the TRACILOGIS platform. In order to illustrate the benefit of this contribution, the NRC is benchmarked with pure reactive control approach (PRC) of which agents in the TRACILOGIS platform were initially instantiated for this PRC. Further interaction behavior of agents in PRC is explained in section 6. Product lateness, resource utilization rate, and makespan as performance measures and work-in-progress build-up between a resource and its upper stream as perturbation, caused for example by machine breakdown, are considered during the experiment.

The rest of the paper is organized as follow: section 2 reviews related works. Then, a scaled-down version of an industrial problem is presented and formalized in section 3. A negotiation-based reactive control approach supported with different cooperation protocols and models is proposed in section 4. After this, section 5 presents a TRACILOGIS platform-based case study by transposing the industrial problem formalized in section 3. Prior to end, section 6 simulates and validates the proposed control approach and analyzes its performance in comparison with that of the PRC. Finally, a concluding remark is presented based on results obtained from the experiments.

⁵TRACILOGIS platform is a technological platform located at wood technology Campus, Epinal, France. It represents a manufacturing system and allows studying different types of identification, traceability, and control approaches for products and logistic chains in wood industry. This witnesses the industrial applicability of the proposed approach for controlling disturbed shop floor in wood industries. Meanwhile, it is composed of four intelligent machines /resources to execute different activities.

⁶JADE stands for Java Agent Development Framework

2. Related works

In the intelligent manufacturing systems' survey, it is well accepted that multi-agent manufacturing system is a good way to deal with unexpected events observed within a shop floor. Thanks to built-in capabilities of smart decision-making entities, it provides right answer to exciting needs and requirements by the framework of Industry 4.0. Without being exhaustive, since the start of the Industry 4.0 framework, many works have dealt with MAS based intelligent manufacturing systems such as (Xiong and Fu 2018; Herrera et al. 2016; Leitão 2009; Wong et al. 2006; Jennings et al. 2001) etc. Continuously, many researchers are still working on this control system as it employs different models depending on the nature of real time events. MAS consists of a society of agents that could potentially collaborate with each other and their outer environment in order to perceive, reason, and converge to a suitable solution (Isern et al. 2011; Wooldridge 2009; Botti and Giret 2008). As part of this society, Weiss (1999) has pointed out knowability, reactivity, and sociability as basic characteristics of multi-agent to create an environment that provides an infrastructure specifying communication and interaction protocols. Wooldridge (2009) has also raised two questions that could always come when one needs to implement these cooperative agents:

- a) How one can build agents that are capable of independent in order to successfully carry out tasks delegated to them? Micro level agent design
- b) How one can build agents that are capable of cooperating with other agents in order to carry out tasks that one delegate to them, particularly when the other agents cannot be assumed to share the same interests/goals? Macro level society design; see Fig. 2.

Different authors have been worked to answer and validate these questions. For instance, Isern et al. (2011) have presented that even though agents are perceived as autonomous entities, they are also members of a society. These agents have to exchange information with other agents and maintain some relationships at organizational level. Consequently, the mere presence of multiple agents makes the environment appear dynamic from the point of view of each agent, with the control system they follow, typically distributed reactive control. In Holvoet and Valckenaers (2006), it is stated that the applicability of MAS is characterized by their large scale in terms of number of agents, their dynamic nature, and their complex functional and non-functional requirements. To sustain these characteristics, different application models and control architectures could always take appropriate attention as they bring amplified benefits. Examples include ant colony (Valckenaers and Van Brussel 2016; Blum and Sampels 2004; Liang and Smith 2004), termite colony (Pannequin and Thomas 2012), potential field (Pach et al. 2014), negotiation (Rahwan et al. 2003; Tonino et al. 2002; Kraus 1997) among many others.

From these models and control architectures, negotiation among the multi-agent has taken viable attention as it enables group of agents to achieve their objectives (scheduling a plan in real manufacturing system for example) at mutual agreement. For instance, Dimopoulos and Moraitis (2006) have stated that individual agents can generate and execute their plans independently. However, as they operate in the same environment and at the expense of others, conflicts may arise and hence they need to coordinate their course of action. Wooldridge (2009) and Vlassis (2007) have also indicated that negotiation among multi-agent is foreseen to host a robust-reactive decision and to reduce drawbacks (myopia⁷, for convenience) of individual agent during its autonomous decision. On the other hand, Tonino et al. (2002) have investigated different automated agent negotiation approaches including game-theoretic, heuristic-based, Contract Net Protocol, and argumentation-based approaches. All of them emphasize the importance of exchanging information between agents in order to mutually influence their behaviors.

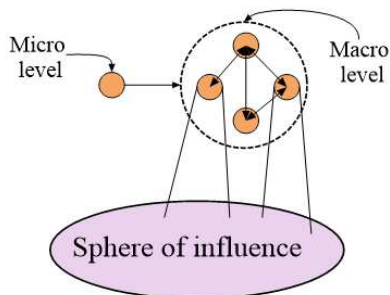


Fig. 2 Structure of cooperative multi-agents (adapted from Wooldridge (2009))

⁷Myopia is drawback of autonomous agents during their decision such as limited capacity to predict events (Rey et al. 2014)

The game-theoretic approach helps to determine an optimal strategy by analyzing the interaction of agents as a game rule between identical and self-interested participants (Nagarajan and Sošić 2008; Rostenschein and Zlotkin 1994). However, unbounded computational resources can be taken as limitation of this approach. To overcome this limitation, a heuristic approach (Aydogan et al. 2013; Kraus 2001) has come with the principle of *produce good enough rather than optimal outcomes*. Irrespective of its advantage, this approach is also known for its sub-optimal outcome as it does not examine the full space of possible outcomes. Consequently, argumentation-based negotiation approach (Monteserin and Amandi 2011; Rahwan et al. 2003) has evolved to overcome the knowledge limitations of agents in the former two approaches. Monteserin and Amandi (2011) have presented that when agents negotiate, the arguments uttered to persuade the opponent are not the result of an isolated analysis, but of an integral view of the problem that they want to agree about. Before the negotiation starts, they have in mind what arguments they can utter, what opponents they can persuade, which negotiation can finish successfully and which cannot. Meanwhile, Rahwan et al. (2003) have surveyed that argumentation-based negotiation approach allows agents to exchange additional information or to argue about their beliefs and other's mental attitudes during the negotiation process. Agents accept, reject, or critique an offer proposed by other agent until they agree on this offer.

MAS negotiation has been also specifically used by different researchers to solve the problem of scheduling in manufacturing systems. For example, it is presented in Madureira et al. (2014) that negotiation in scheduling is generally used to improve quality of final solutions. Madureira et al. (2014) worked on negotiation mechanism to deal with resolution of scheduling in real manufacturing system. They proposed this mechanism considering set of resource agents and a coordination mechanism combining a single solution obtained by the resource agents into a global solution. The works of Zattar et al. (2010) and Adhau et al. (2012) in Madureira et al. (2014) have also proposed negotiation mechanisms for integrating process planning and scheduling and for distributed multi-project scheduling respectively. In Xiong and Fu (2018), a new immune multi-agent scheduling system has been developed to solve a flexible job shop scheduling problem. But its applicability was based on immune system such that a body is protected from foreign antigens by immune response. If the immune system is unexpectedly failed, there will not be chance to recover and solve the scheduling problem.

To summarize the review, an agent in multi-agent system maximizes its benefit on the expense of its neighbor agents. Hence, several researchers have given due attention to the applicability of negotiation for controlling such problems in the field of networking sciences, sociology, anthropology, philosophy, economics etc. However, several challenges within manufacturing systems of a dynamic shop floor such as design of decision model, design of interaction protocols, integration of technological solutions etc. remains yet unsolved. Therefore, exploring the applicability of different MAS negotiation algorithms that help manufacturing components dynamically cooperate for improving performance of the manufacturing system is still a demanding research area. Mezgebe et al. (2018) have proposed negotiation model for flexible manufacturing system considering smart product agents scheduled to be processed on different resource agents. The communication protocol was fully controlled by the product agents. Meanwhile, the role of RFID technology was partly emulated to help detect unexpected events and send back to product agents for its management. Nevertheless, the interaction media⁸ among all agents was not fully formalized and modelled. This is the reason that leads us to further investigate the application of negotiation-based control approach for setting best priority-based product sequencing based on a real manufacturing system extracted from carpentry factory. The problem formalization, interaction protocol, and experimental experiences in this work are significantly improved with a greater analysis of performance indicators on higher instance.

3. Problem statement

As it is emphasized in Fig. 1, if unexpected system⁹ disturbance that significantly impact a master production schedule (MPS) has occurred, it generates non-value adding activities such as queuing, waiting, and moving times. Heuristic decision is expected right after such disturbance to deal with the variabilities and hence to save the master schedule validity. In this case, the big issue is to make decisional entities defining best priority-based routing and/or sequencing in order to behave in a sense that the whole system stays globally near optimal. Empowering decisional entities to decide cooperatively and making all key performance indicators under the control of these entities would be best option. In order to substantiate such decision, a physical system (a scaled version of lacquering and polishing shop floor from carpentry factory) with four chronological activities namely cutting

⁸Interaction media includes cooperation, collaboration, communication etc.

⁹System is an entire working environment within a shop floor

(resizing), drilling, sanding, and coating is considered. For experimental convenience, these activities are modeled based on the TRACILOGIS platform presented in section 5. Meanwhile, it is considered that a planning department of the carpentry factory prepares a weekly predictive schedule for each customer order the shop floor received. If unexpected event/s is not noticed, this predictive schedule realizes the full completion of all orders with best rewards. However, it is taken that the physical system is suffering with work-in-progress (WIP) build-up, caused by machine breakdown for example, between a resource and its upper stream. This event can happen either at the beginning, centre, or at the end of a launched system; in our case it is considered to happen at the beginning and at the center. Accepting this unexpected event could trade-off throughput and resource utilization if an immediate action is not steadily taken. And rescheduling at this scenario is considered as dynamic scheduling such that the real time decision makers dynamically adapt and/or react to this work-in-progress build-up.

Formalizing the problem To formalize the problem, it is considered that \mathcal{C} customer orders (or batches for convenience) each with their own product types and different processing times on a resource are received by a shop floor. Meanwhile, product lateness, resource utilization, and makespan as key production performance indicators are used to formalize the work-in-progress build-up problem. For convenience of description, indexes, variables, and parameters used are listed as follow.

<i>Indexes</i>		<i>Variables</i>	
i, j	Product index	C_i	Completion time of product i of customer order c
r	Resource index	U	Utilization rate of resource r
c	Customer order or batch index	C_{\max}	Makespan
k	Operation index	L_i	Lateness of product i
n	Number of products	$Tard_i$	Tardiness of product i
Φ_s	Decision node within a system		
CR	Critical ratio		
<i>Parameters</i>			
R	Set of resources which perform operations k of product i	e_{ir}	Release/exit time of product i from resource r
\mathcal{C}	Set of batches or customer orders	ε	Exit/release rate of product i
O_{ic}	Set of operations k of product i of customer order c	CR_i	Priority of product i on neighbor products $j, \forall j \in \mathcal{C}_c$
v_{kr}	Intention of product on resource r	ct_i	Current time of product i
t_{ir}	Travel time of product i to resource r	lt_i	Lead time remained a product i normally takes to completion
q_{ir}	Queuing time of product i before its operation k on resource r	dd_c	Due date of last product of customer order c
q_n	Buffer size: number of products in a queue	s_{ir}	Setup time of product i on resource r for every switch of products n
a_{ir}	Arrival time of product i to resource r	x_r	Actual working time used by resource r
λ	Arrival rate of product i	y_r	Maximum available time of resource r
p_{ir}	Processing time of product i on resource r	z_{ir}	Binary variable

Two different groups of decision-making entities (or agents) namely product entities and resource entities are used to model the proposed control approach. They are engaged in processing a knowledge that contains information and methods on how to execute a certain operation on a certain resource. Explicitly, product entities are central and active elements within the decision loop and they store process and knowledge needed to ensure the correct execution within a system. This cognitive ability allows them to make the required decision easier. Meanwhile, each product is in charge of transferring customer orders (and/or master schedules) to resources and needs to keep its due date by completing its route as early as possible. Hence, they frequently compute their completion time using Eq. (1) with an objective to minimize at least their queuing time. This gives the completion time of each customer order as it is presented in Eq. (2). Subsequently, these two models help them to

evaluate the impact of unexpected event, if occurred, on MPS by computing makespan using the model specified in Eq. (3).

$$C_i = ct_i + \sum_{i=1}^n \sum_{r \in R} z_{ir} \bullet (s_{ir} + p_{ir} + t_{ir} + q_{ir}), \quad \forall i \in \mathcal{C}_c \quad (1)$$

$$L_i = C_i - dd_i, \quad \forall i \in \mathcal{C}_c$$

$$C_c = \max_{i=1}^n (C_i) \quad (2)$$

$$C_{\max} = \max_{c \in \mathcal{C}} (C_c) \quad (3)$$

With an objective to minimize C_{\max}
Subjected to

$$(1) C_{ik} \geq C_{i(k-1)} + p_{ir}, \quad \forall i \in \mathcal{C}_c, r \in R, k \in O_{ic} \quad (3a)$$

$$(2) q_n \leq (\lambda * q_{ir}), \quad \forall i \in \mathcal{C}_c, r \in R \quad (3b)$$

$$(3) Tard_i \leq \max(L_i, 0), \quad \forall i \in \mathcal{C}_c \quad (3c)$$

$$(4) \sum_{c \in \mathcal{C}} c_r = 1, \quad \forall r \in R \quad (3d)$$

$$(5) \{C_i, C_{\max}, L_i, U\} \geq 0 \quad (3e)$$

$$(6) z_{ir} \in \{0, 1\} \quad (3f)$$

Constraint (3a) indicates precedence constraint such that in order a product i to be processed in next resource r , it must be processed in resource $r-1$. Constraint (3b) states the effect of high arrival rate of products on buffer size that each product entity has to reduce based on the supervision of the resource that they are approaching. Constraint (3c) presents no product tardiness if its completion time ends before its due date. Constraint (3d) ensures only one customer order and/or product is executed on one resource at a time. Constraint (3e) presents the positivity of variables specified. Finally, constraint (3f) illustrates a binary variable which is set to 1 if product i is processed on resource $r \in R$, 0 otherwise.

On the other hand, resource entities (mainly machine entities) offer production capacity and functionality in order to execute intentions broadcasted by product entities. They control the timing and type of execution through their informational part. During the execution process, resources must be fully utilized in order to return their investment cost before its depreciation time. This requires understanding the interdependence of the resource's investment cost and its utilizability and hence such dependency results in trade-offs between investment and usability costs (Vander Veen and Jordan, 1989). Thus, part allocation, production cycles, and setup times of the resource must be managed so that the required utilization decision would always be higher. This hypothesis triggers the resource entities to calculate their utilization rate (U) for every event occurred using the model presented in Eq. (4). According to APICS¹⁰, utilization rate measures how intensively a resource is being used and it is expressed as percentage of the amount of resource's available time required for productive and billable execution process.

$$U = \frac{x_r}{y_r} * 100\% \quad (4)$$

Meanwhile, this resource entity serves as supervisor to select set of products from set of candidates, based on a defined parameter, approaching to it. This helps the resource to keep the defined queue size (or to satisfy constraint (3b)) around its workspace.

4. Proposal: a Negotiation-based Reactive Control approach (NRC)

The c batches are received to be executed at full time horizon based on their sequence of arrival: $(\mathcal{C}_c) \prec, \dots, \prec (\mathcal{C}_2) \prec (\mathcal{C}_1)$ such that \mathcal{C}_1 is received first, \mathcal{C}_2 , and \mathcal{C}_c followed. Each batch has its own completion time: $C_2 < C_1 < \dots, < C_c$. At this instant, the work-in-progress build-up event is made to

¹⁰APICS stands for American Production and Inventory Control Society (12th edition)

happen. If the routing sheet (or launched system) continues with the arrival operational sequence, products of each batch understand that they will be late and higher makespan because of a bottleneck resource¹¹ will be expected. Thus, the batches received over a specified period of time must be combined into a production plan and dispatched together by their sequence of completion time: $(C_c) \prec, \dots, \prec (C_1) \prec (C_2)$. Products of C_2 precede products of C_1 and then products of C_c follow. This limits a work-in-progress size in order to create smooth and adaptable system with minimal product's queuing time, Eq. (1). In order to satisfy this new routing sheet, the system is made to pursue negotiation¹²-based reactive control approach using broadcasting interaction protocol. It uses *update status and routing* principle to show that decision entities continuously interact to optimize the routing sheet. Its control architecture is presented in Fig. 3 maintained by explanation of each step revealed on the architecture. Meanwhile, the interaction protocol is supported with UML sequence and UML class diagrams shown in Fig. 4 and Fig. 5 respectively. The directory service in the UML class diagram acts as a repository for the system providing information about active decision entities.

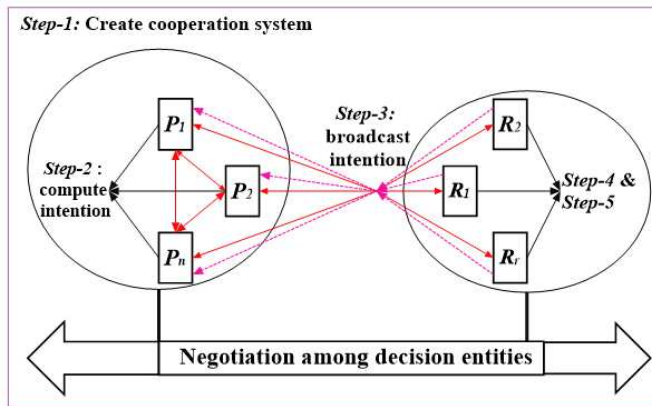
Step-1: Products and resource entities create a cooperation system for their common goal. They negotiate on execution of operations needed to perform dispatched quantities of products of each customer order. Products to broadcast their intention and resource to select and execute a product or set of products by comparing its utilizability and the products' intention

Step-2: The intention of every product of each customer order is to arrive and process in each resource as early as possible. Hence, each product computes its intention according to the model presented in Eq. (5) such that new product intention is a function of its previous intention and the intention of others.

$$v_{kr} = [a_{ir} \ p_{ir} \ e_{ir}], \quad \forall i \in C_c, \ r \in R, \ k \in O_{ic} \quad (5)$$

Step-3: The computed products' intention is broadcasted into the created system that incorporates a resource that the product/s is approaching and neighbour products j , $\forall j \in C_c$.

Step-4: After each product entity broadcast its intention, resource entities are expected to being fully utilized in order to return their investment cost within short period of time. Taking this precondition, product prioritization has to be validated by resource entities through calculating their utilization rate (U), Eq. (4), for every broadcasted intention of the product. As soon as resource entity computes its utilizability, it compares with the broadcasted product's intention in order to accept if it does not affect its utilization rate. Otherwise, the resource entity queries products to revisit their intention so as to contest with its utilizability. This strategy helps product entities to balance an input buffer in front of the resource that they are approaching. For example, when last product of C_1 and first product of C_c meet in decision node Φ , the intention of last product of C_1 is to precede the first product of C_c as the due date of C_1 is less than that of C_c . If this intention is not satisfied, it delays the entire completion time of all orders and hence the makespan.



Legend

- \longleftrightarrow Bi-direction communication
- \dashrightarrow Unidirection communication (Supervision)
- \rightleftarrows Association for updating and informational sharing

Fig. 3 The proposed negotiation architecture

¹¹A resource with highest product's operation processing time

¹²Negotiation is a process by which a joint decision is reached by two or more agents, each one trying to reach an individual objective (Madureira et al., 2014)

Step-5: In order to balance an input buffer for product's remaining operations, maximize the resource utilization, and hence minimize the makespan, decision entities choose a critical ratio (CR). Critical ratio is mostly used dispatching rule to prioritize products within a shop floor (Wong et al. 2006). Resource entity requests products to update their state based on the critical ratio and product entities calculate their critical ratio (CR_i) according to the model given in Eq. (6). lt_i in the equation includes setup, processing, travel, and queuing times.

$$CR_i = \frac{dd_i - ct_i}{lt_i} \quad (6)$$

Such that $\begin{cases} CR_i < 1; \text{ product is behind schedule (tardy)} \\ CR_i > 1; \text{ product is ahead of schedule (early)} \end{cases}$

After this, with an intention to select a product (or set of products) with least critical ratio first, resource entity request products to arrive based on their critical ratio sequence. This balances the intention of products to complete their route within their due date and the utilization rate of resource. Meanwhile, resource entity could be able to understand whether the products are approaching tardy or not. If products are approaching early (or constraint (3b) is not satisfied), *step-6* follows

Step-6: While the routing sheet execute following *step-1* to *step-5*, a resource might be incapacitated to perform according to the predefined negotiation strategy and protocol for different reasons (for example, machine breakdown). At this time, the resource entity (where the unexpected event originated) creates another state (called buffer size, constraint (3b)) to begin another negotiation with the tasks (or semi-finished products) that previously assigned to that resource. Hence, the resource informs the upcoming products to wait somewhere in between their current position and the resource. This leads product entities to wait or buffer in their current location until the buffer size is reduced to a defined limit. The buffering ceases when the *restart forward routing query (or ready to work message)* is sent by the constrained resource entity.

This negotiation-based control approach continues until the completion of all customer orders launched or a cancellation request is sent by a customer.

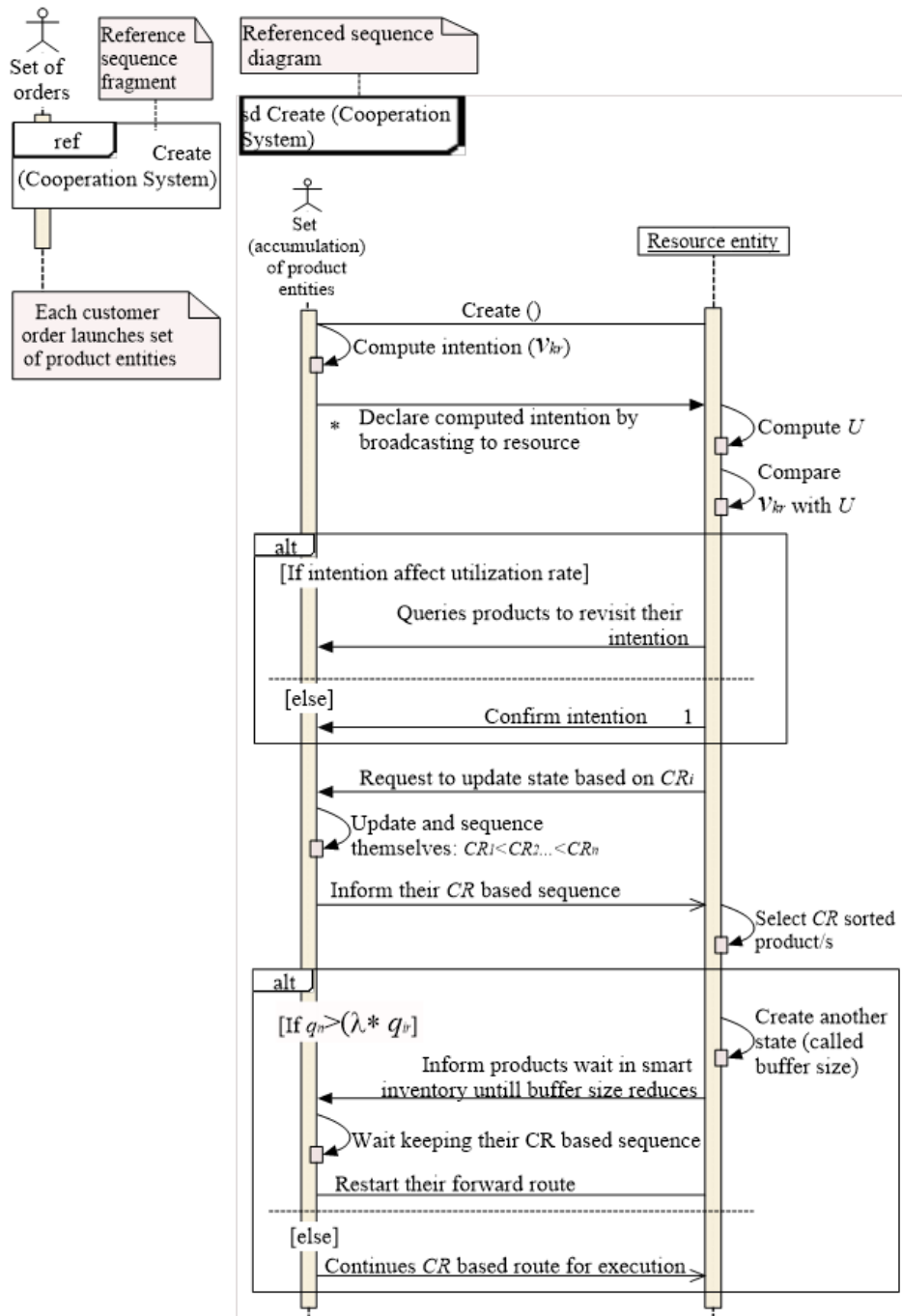


Fig. 4 UML sequence diagram for the proposed control approach

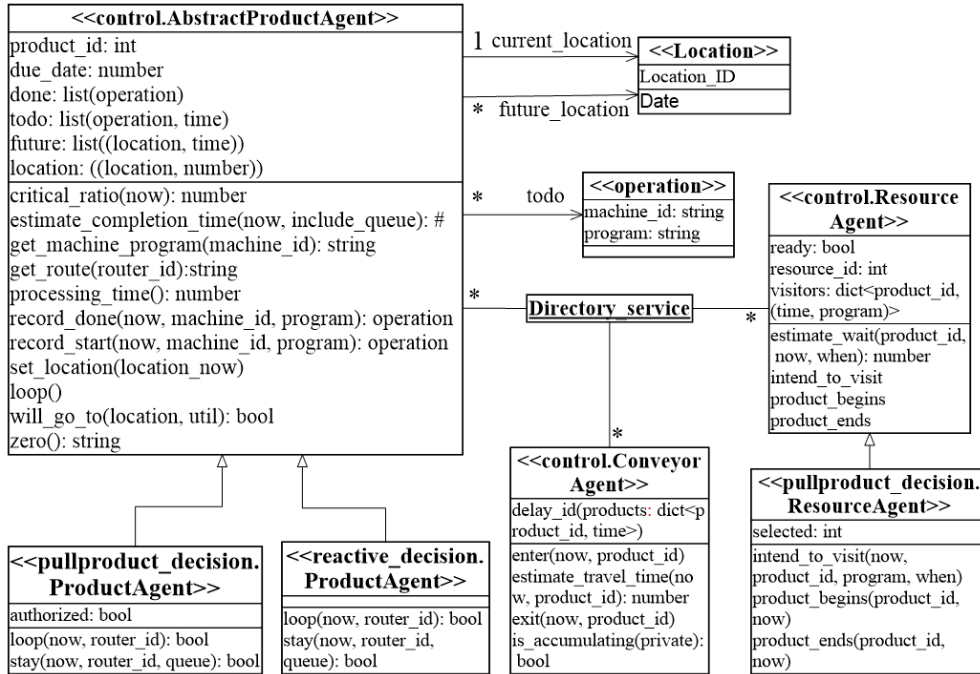


Fig. 5 UML class diagram for the developed multi-agent system negotiation

5. Case study: a TRACILOGIS platform

In order to verify and validate the performance of the proposed control approach, TRACILOGIS platform shown in Fig. 6, is used as case study. TRACILOGIS platform is composed of extensive system of networks linking different actors for optimal achievability. Sensors, actuators to automata or automata to computers, RFID sensors, Production Line Controller (PLC) that manage all automaton actions are some components of the networked system. Meanwhile, this platform allows assessing the impact of new and smart technologies, testing and demonstrating new production decision modes, and confirming running modes for production control. It has four automation zones: Zone-A for transformation, Zone-B for smart buffering, Zone-C for colored plate (components to be assembled) sorting, and Zone-D for plate and tablet assembly/disassembly. The R_i notations in Fig. 6 denote the four intelligent resource entities $R_1, R_2, R_3,$ and R_4 for line marking, point marking, plate assembling, and tablet assembling respectively. In case of unpredictable failure of either resource, the other resources have capability to perform all operations left unprocessed. The Cutting (resizing), drilling, sanding, and coating activities presented in section 3 are transposed to these customer resources respectively. Each resource has different processing times for each product type of each customer order.

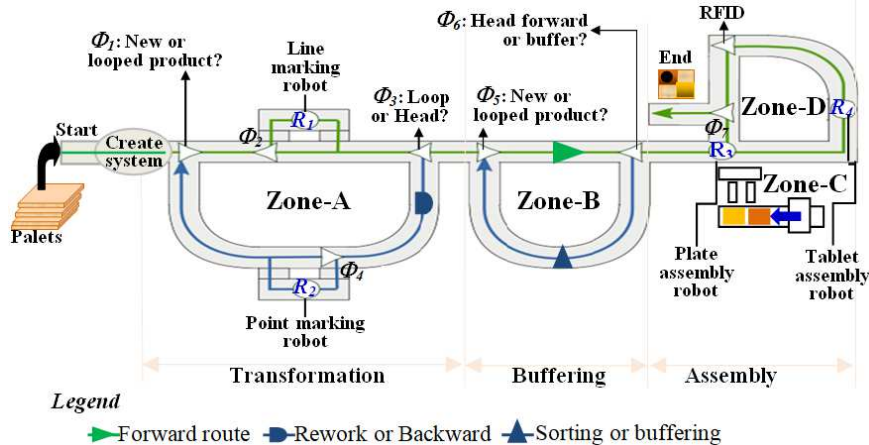


Fig. 6 Layout of the TRACILOGIS test-bed platform

Simulation configuration of the platform A central schedule for three customer orders C_1 , C_2 , and C_3 each with their own product types (presented in Fig. 7) and due date is developed by a central scheduler. For each customer order, ten intelligent products (where products $\{P_1, P_2, \dots, P_{10}\}$ stands to C_1 , $\{P_{11}, P_{12}, \dots, P_{20}\}$ to C_2 , and $\{P_{21}, P_{22}, \dots, P_{30}\}$ to C_3) are launched on the platform and the processing time of each operation of each product type is presented in Table 1. Note in the table that the processing time in R_2 for order C_2 is set to zero to indicate that products of this order have no operation to be performed in resource R_2 . Meanwhile, the completion time C_i and due date dd_c of each customer order is estimated from the lead time of the launched system (or the conveyor based platform). It is developed considering the travel time and the processing time of each operation of a product on each resource r . Each product agent follows a standard routing sheet based on the platforms' configuration shown in Fig. 6. For example, every product of C_1 are expected to pass through R_1 for their first operation, move to R_2 for their next operation, loop back to R_1 for inspection (optional), travel to R_3 for their third operation, and finally route to R_4 for the last operation. As per the algorithm presented in section 4, these orders are received based on their arrival sequence: $(C_3) < (C_2) < (C_1)$. However, if the routing sheet follows such a work-in-progress disturbed system, it increases the completion time of products of C_2 as the products will be obligated to loop in Zone-B of Fig. 6 without added value until all semi-processed products of C_1 complete their route in Zone-A and proceed to resource R_3 for their next operation. This happens because, as per the initially launched execution plan, products of C_1 purely need to serve first in R_3 .

Hence, products of each customer order *create a system*, shown at the beginning of Fig. 6, to be combined and dispatched by their sequence of completion time: $(C_3) < (C_1) < (C_2)$. While entities decide to make such change, critical ratio is used with an objective to process product/s with shortest critical ratio first. This indicates that the intention of each product is to complete the routing sheet based on the sequence of earliest due date first. Consequently, from Eq. (5), the intention, at current time ct_i , of a single product from each customer order is computed in Table 1. In the computed intention values, the exit time of customer order c from resource r (e_{ir}) is the cumulative sum of travel time of the order to resource r (t_{ir}) and processing time of the order on the resource (p_{ir}). From the configuration in Fig. 6, the travel time of the product from 'start to R_1 ', ' R_1 to R_2 ', ' R_1 to R_3 ', ' R_2 to R_3 ', ' R_3 to R_4 ', and R_4 to End' are given as 5, 29.5, 18.5, 60, 7.5 and 30 seconds respectively.

Meanwhile, in the simulated experiment, Zone-B in Fig. 6 is used to resequence and buffer products (at the time of necessity) and the travel time of a complete loop in this zone is taken to be 43.5 seconds. After computation, the product's intention is evaluated by a resource that the product is approaching and neighbor products j for execution.

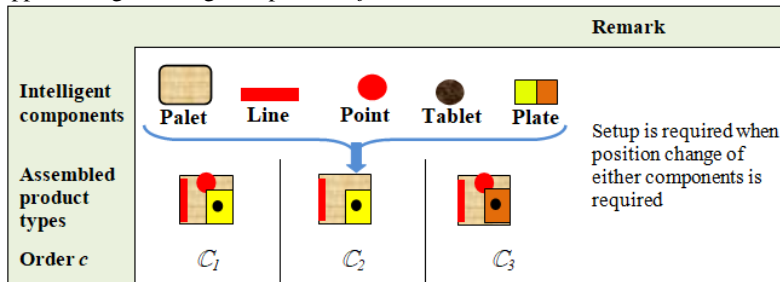


Fig. 7 Intelligent components and assembled product types of each customer order c

Table 1 Processing time (in seconds) and computed intentions of each order

Order/ batch	Processing time of each job (in seconds)				Computed intention of each product type					C_i (Seconds)	dd_c (Seconds)
	R_1	R_2	R_3	R_4	$v_{1(1)}$	$v_{2(2)}$	$v_{3(1)}$	$v_{4(3)}$	$v_{5(4)}$		
C_1	7	1	9.5	7.7	[5 3 8]	[37.5 1 38.5]	[85.5 4 89.5]	[108 9.5 117.5]	[125 7.7 132.2]	162.2	400
C_2	8	0	9.5	7.7	[5 8 13]	[0 0 0]	[0 0 0]	[31.5 9.5 41]	[48.5 7.7 56.2]	86.2	300
C_3	11	3	9.5	7.7	[5 11 16]	[45.5 3 48.5]	[0 0 0]	[108.5 9.5 118]	[125.5 7.7 133.2]	163.2	600

6. Results and discussion of simulation and applicability implementation experiments

The performance of the proposed control approach is mainly showed on bottleneck resource and its upper stream. It is benchmarked with pure reactive control approach, which uses *change the product intention* routing principle, in order to pinpoint best priority-based product sequencing. Agents in the platform were initially instantiated for this PRC such that product agents have little information about neighbor's state, resource status, and routing sheet. Product agents can cancel the CR based selection by the resource entity. Moreover, product agents are active and well known of their own state (where they simply send their intention to be executed as per their arrival sequence) but resource agents are mostly dormant that simply waits for instructions from product agents. Such communication may create mixing up of product intentions within the routing sheet and hence high WIP build-up, unutilized resource capacity, and linearly increased makespan follows. Taking three performance indicators namely product lateness (L_i), resource utilization rate (U), and makespan (C_{max}), MATLAB and Python coded simulation experiments with two hundred simulation replications have been executed to articulate the significance of NRC over PRC; shown in sub-section 6.1 and sub-section 6.2. Meanwhile, in order to validate the applicability of the proposed control approach, an implementation experiment has been conducted and its results are presented in sub-section 6.3.

6.1 Production performance indicators

The simulation result considering the product lateness (L_i) is illustrated in the lowest part of the vertical primary axis in Fig. 8. Lateness measures how products are tardy because of constraints encountered in upper stream or how products are early due to constraints occurred in downstream. In the NRC, product agents have able to reduce lateness to a minimum of 24.59 seconds in one of their simulation run. However, in PRC, the minimum lateness has recorded to be 30.71 seconds. This shows that the product lateness in the routing sheet has reduced by 19.92% as a result of the environment created for NRC. Moreover, the graphical pattern in the NRC is more stable than that of the PRC because the decision entities in the PRC did not show interest to cooperate for minimizing the makespan instead they route for minimizing their individual completion time only. Following the reduction of product lateness, the resource utilization rate has also improved; see the vertical secondary axis in Fig. 8. For instance, on average, resource R_i is utilized in NRC 18% better than in PRC.

Makespan is also used to validate the performance of the proposed control approach. As it can be observed in the upper part of the vertical primary axis in Fig. 8, once product agents in the PRC set their sequential route, they showed interest to follow this route instead of setting another optimal route that help minimize makespan. But in the case of NRC, product and resource agents update their current route based on the unexpected event occurred so as to minimize the makespan. Accordingly, the simulation experiment has resulted, on average, 920.26 seconds of makespan in PRC and 777.88 seconds following the pursuance of NRC. This indicates that the makespan is reduced by 15.47% after employing the negotiation-based control approach. Meanwhile, a statistical probability characteristic is used to find patterns and predict certain outcomes from a set of population (number of simulation replications in our case). As part of this, a statistical parameter called performance density function (*pdf*) is used to show the effect of the proposed control approaches in minimizing the makespan. *pdf* is a function of a continuous random variable, whose integral across an interval gives the probability that the value of the variable lies within the same interval. Consequently, as it is illustrated in Fig. 9, the average makespan in NRC has higher *pdf* value than the makespan in PRC to indicate that all replicated makespan values lies close to their mean value (or the 777.88 seconds). However, in the PRC, the *pdf* is lower to indicate unstable closeness of all replicated makespan to each other across their interval. Likewise, the *pdf* value increases with the decrease of makespan values in the NRC and vice-versa in the PRC. This has happened because in the PRC, product entities always decide to choose their shortest path without the knowledge of the resource entity that they are approaching until they exit from the launched system. This may be successful for some product entities (perhaps for product entities that entered the launched system first) but not for the others. Because of the purely reactive decision taken by the product entities, the resource entity that they are approaching would not select any of these product entities and hence the product entities indefinitely loop in Zone-A or Zone-B in Fig. 6. And this event increases the completion time of each product entity. However, in the NRC, as all the product entities together with the resource entities re-sequence themselves considering an agreed parameter, the simulated makespan values are very close to each other and hence reduced to indicate that a negotiated and defined routing sheet is followed until their exit from the launched system.

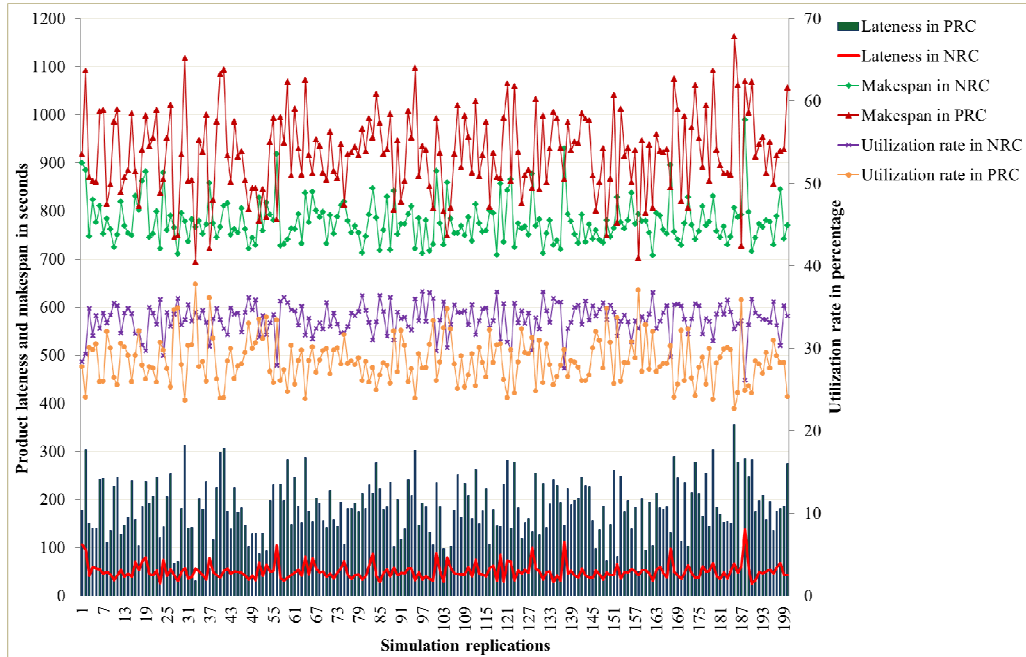


Fig. 8 Product lateness, utilization rate, and makespan in both control approaches

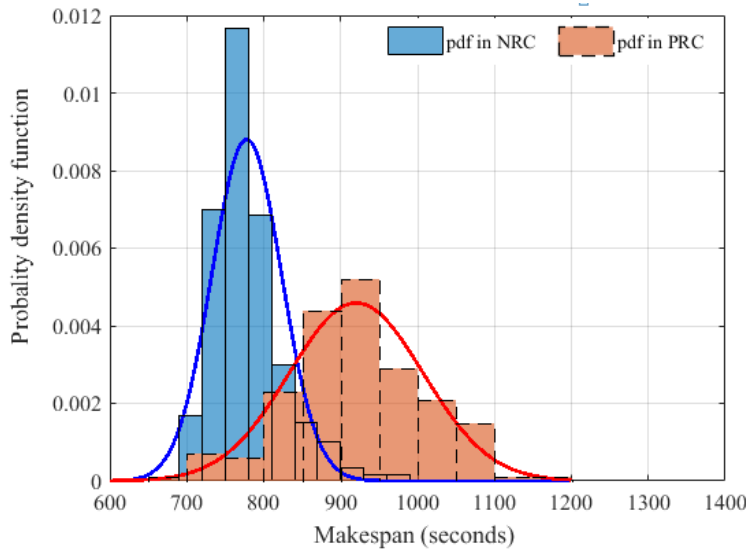


Fig. 9 Probability density function for both control approaches

6.2 Correlation analysis

The simulation experiment has also correlated two performance indicators to illustrate the performance of the proposed control approach. For instance, as shown in Fig. 10, while the resource utilization rate of R_j increases, the makespan decreases. However, the difference is shown at the percentage of variance in the makespan that the resource utilization rate explains collectively. In the dotted red regression line in Fig. 10, the cumulative effect of the resource utilization rate is shown at the left side of the regression line; less utilized and higher makespan. This happens because successor products reach the resource entity without understanding its current status and hence these products are obligated to wait close to the resource until they arrive to the resource for their next operation. The solid black regression line in Fig. 10 has reversed this tendency; the collective effect of the resource utilization rate is shown at the right side of the regression line. The makespan value is linearly shifting downward with the increase of the resource utilization rate. This happens because products are arriving to the resource keeping their initially computed critical ratio and a buffer size defined in constraint (3b).

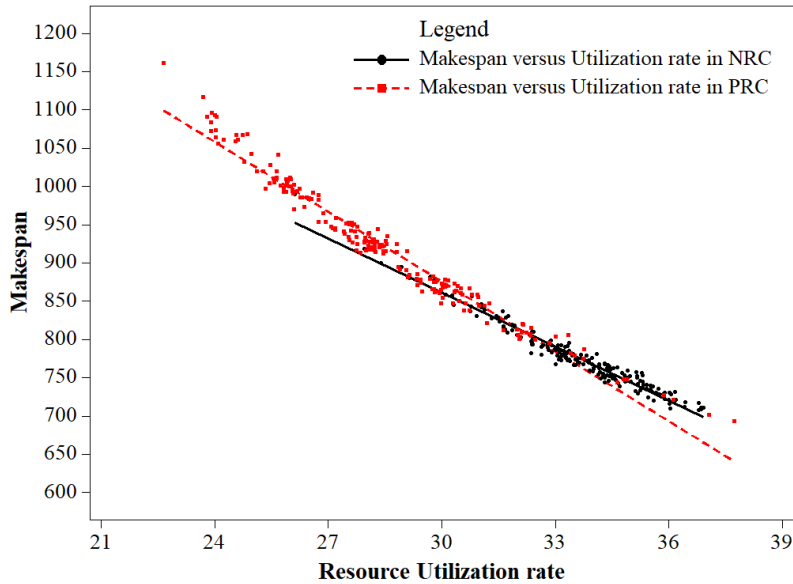


Fig. 10 Scatterplot of makespan versus utilization rate in both control approaches

Meanwhile, even though the R-squared¹³ value is similar in both control approaches, the place of variability closeness to the regression line is different to indicate a linear effect of the resource utilization rate to minimize makespan. Most of the replicated makespan values are closely positioned at the most optimal utilization rate of the resource after employing the NRC; see the most right side of the solid black line in Fig. 10.

6.3 Applicability implementation results

In order to show the applicability implementation of the NRC, production is made to begin at Φ_1 in Fig. 6 where the first RFID reader detects the first palet and production ends immediate to product's operation completion in R_3 , Fig. 12. There is no need for products to go to R_1 for line marking and R_2 for point marking. Meanwhile, the implementation experiment is conducted using three, five, and nine products to show the scalability of the proposed control approach. In this paper, the result obtained using nine products is presented because the implementation platform has a maximum of nine palets (where this palet indicates the initial raw material to be assembled with all necessary parts such as plates in Fig. 6). The applicability test is made to show how a defined queue size can be implemented using TracilOgis Platform Agent SystEm (topase) on the workspace of R_3 in Fig. 6. According to constraint (3b), the defined queue size is two and the decision agents considered in the topase are NegoProductAgent and ResourceAgent. As it is shown in Fig. 11(a), the products *update their position* through the NegoProductAgent. Once the product position is updated, the resource R_3 in turn *informs its ready to work*, Fig. 11(b), based on the defined queue size and the completion of plate assembly operation for the product type on board. If two semi-processed products are queued on the workspace of R_3 , the successor products loops in Zone-B in Fig. 6 keeping their critical ratio until one of the queued product starts to load on R_3 for its next operation. This continues until the completion of all operations of the products launched, see Fig. 12.

Results of the applicability implementation test is shown in Table 2. The real completion time of each product is 99.99% same as their estimated completion time. Two products are only late by only two seconds from the total amount of the estimated completion time. Compared to the real completion time in Fig. 12, the results in Table 2 include the product's travel time from R_3 to *end* and the processing time of the last operation of products in R_4 .

¹³R-squared is a statistical measure of how close data are to the fitted regression line

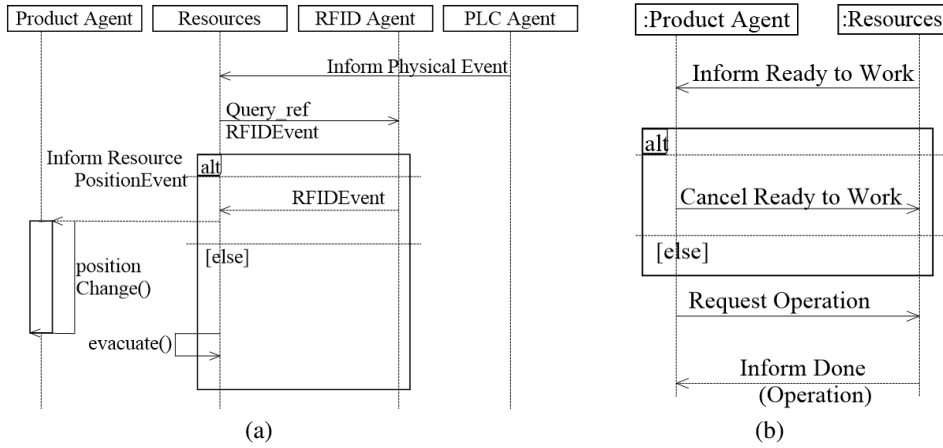


Fig. 11 UML topose sequence diagram (a) for product's *update position*. (b) for resource's *inform ready to work*.

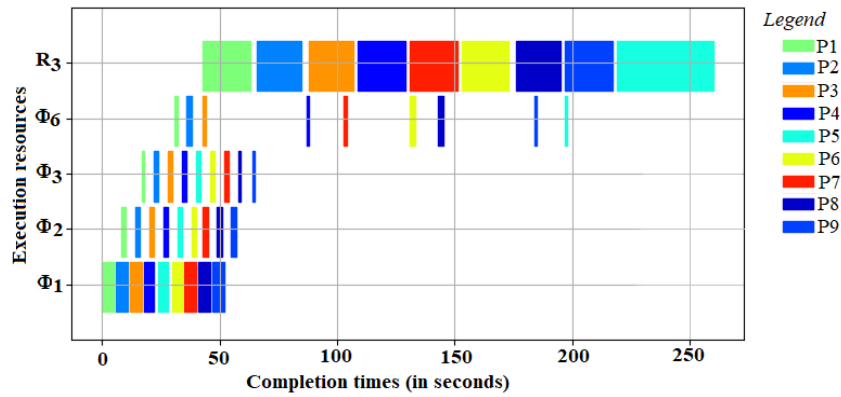


Fig. 12 Gantt chart for real completion time of each product type

Table 2 Estimated completion time versus real completion time for each product

Order	Product_id	Production start (in seconds)	Estimated C_i (in seconds)	Real C_i (in seconds)	$Tard_i$ (in seconds)
C_1	P_1	0	115	116	1
	P_2	6	137	137	0
	P_3	12	159	159	0
C_2	P_4	17	183	183	0
	P_5	23	311	311	0
	P_6	29	225	225	0
C_3	P_7	35	203	203	0
	P_8	41	245	245	0
	P_9	47	268	269	1

6.4 Discussion

Despite the PRC has advantage to be the most generic in a rapidly changing context, regardless of communication protocols and most disturbances, the performance remains poor. In fact, three of the simulated performance indicators have shown significant advantage of NRC in increasing adaptability and flexibility of a system to unexpected event within manufacturing systems. This in turn has made the launched routing sheet to minimize its makespan by considering all necessary constraints. Meanwhile, when decision change (e.g., re-sequencing) is queried by either of decision entity, all entities in the NRC have able to dynamically satisfy this change through the cooperative negotiation. In a real manufacturing system, production satisfaction is perceived if customers receive their finished products within an estimated completion time. The applicability implementation test has satisfied this requirement as the NegoProductAgent and the ResourceAgent during the NRC have able to almost equalize the estimated completion time with their real completion time.

A full study of NRC design parameters tuning need to be done in further works, because it depends on the type of perturbation(s) a system faces and need more experiments. However, some concluding remarks and discussions can be drawn as follow.

Due date Schedule changes should be observed according to product due date values, since critical ratio is used by the resource to decide. Note that when critical ratio of a product is low, it will not be supervised by the resource and switch to pure reactive mode. Thus, in theory, it is possible to observe the emergence of misbehavior where all products switch to non-collective behaviors (pure reactive). It might occur in a extremely strained situation of all products.

Resource field view All resources select a candidate of products, from set products, which will be processed in their workspace. The size of this space has an important impact on the algorithms. For example, if the size is low, lack of proactivity is observed because the resource only selects the first or the closest product. If the size is huge, a meaningless waiting time will occur because the selectin of distant product will let other closer products to wait during the traveling.

Size of waiting queue It corresponds to the number of products simultaneously selected by the resource that they are approaching. If the value of this size is low, the resource shortage risk is high. If the value is high, it might lead to pure reactive mode by most products since each product arriving to the resource is directly selected. Specifically, the minimal queuing time and maximal arrival time of a newly selected product must be compared in order to prevent the resource shortage (i.e. the resource remains supplied).

Conclusions and perspectives

This paper has presented the role of negotiation among multi-agents to support decision-making capability of decision entities in a manufacturing process impacted by variability. Product lateness, resource utilization rate, and makespan were chosen as key performance indicators to validate the proposed control approach. Three of the performance indicators have shown significant advantage of NRC to adapt unexpected event occurred within the manufacturing process. For instance, the simulated result of NRC has minimized makespan, on average, by 15.47% over the pure reactive control approach. The applicability implementation test conducted on the TRACILOGIS platform has also showed that the real completion time of each product is 99.99% same as their estimated completion time. This confirms the suitability of the proposed control approach for industrial applicability.

However, it has left to fully enrich the negotiation-based control approach as decision entities (or agents) may decide alone while they are on their route. Thus, to enhance the negotiation, a consensus-based control approach must be explored with an objective to make agents converge towards highly predefined and common intention. They have to continuously negotiate before they start execution and convergence to a common offer. Design and development of such control system will be the continuous work of this paper. Meanwhile, further applicability implementation test is required, by considering more than nine product agents, to show the scalability and complexity of the proposed control approach. Finally, the *pdf* value presented in Fig. 9 hints that the predictability of NRC may be useful beyond reducing the makespan. Hence, considering this statistics, further research work must be conducted to compare expected results from scheduling with the introduction of different disturbances.

Acknowledgments

The authors gratefully acknowledge the financial support of the CPER 2015-2020 Projet Cyber-Entreprises du programme Sciences du numérique, through regional (Région Lorraine, Grand EST), national (DRRT, CNRS, INRIA) and European (FEDER) funds used to extend The TRACILOGIS Platform.

References

- Adhau, S., Mittal, M.L., Mittal, A. (2012). A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach. **Engineering Applications of Artificial Intelligence**, 25(8), 1738–1751
- Aydogan, R., Baarslag, T., Hindriks, K.V., Jonker, C.M., Yolum, P. (2013). Heuristic-Based Approaches for CP-Nets in Negotiation. In **Complex automated negotiations: Theories, models and software competitions** (pp. 113-123). Berlin, Heidelberg: Springer.
- Blum, C., & Sampels, M. (2004). An ant colony optimization algorithm for shop scheduling problems. **Journal of Mathematical Modeling and Algorithms**, 3(3), 285–308
- Botti, V., & Giret, A. (2008). **ANEMONA: A Multi-agent methodology for Holonic Manufacturing**

- Systems**. London: Springer Science & Business Media
- Dimopoulos, Y., & Moraitis, P. (2006). Multi-agent coordination and cooperation through classical planning. **In Intelligent Agent Technology, IEEE/WIC/ACM International Conference** (pp. 398–402). IEEE Computer Society
- El Haouzi, H., Thomas, A., Pétrin, J.-F. (2008). Contribution to reusability and modularity of manufacturing systems simulation models: Application to distributed control simulation within DFT context. **International Journal of Production Economics**, 112(1), 48–61
- El-Haouzi, H.B. (2017). Contribution à la conception et à l'évaluation des architectures de pilotage des systèmes de production adaptables: vers une approche anthropocentrée pour la simulation et le pilotage (**Habilitation a diriger des recherches, Université de Lorraine**)
- Herrera, C., Belmokhtar-Berraf, S., Thomas, A., Parada, V. (2016). A reactive decision-making approach to reduce instability in a master production schedule. **International Journal of Production Research**, 54(8), 2394–2404
- Holvoet, T., & Valckenaers, P. (2006). Exploiting the environment for coordinating agent intentions. **In International Workshop on Environments for Multi-Agent Systems** (pp. 51–66). Berlin, Heidelberg: Springer
- Isern, D., Sánchez, D., Moreno, A. (2011). Organizational structures supported by agent-oriented methodologies. **Journal of Systems and Software**, 84(2), 169–184
- Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Wooldridge, M.J., Sierra, C. (2001). Automated negotiation: prospects, methods and challenges. **Group Decision and Negotiation**, 10(2), 199–215
- Kraus, S. (2001). **Strategic Negotiation in Multi-Agent environments**. London: MIT press
- Kraus, S. (1997). Negotiation and cooperation in multi-agent environments. **Artificial Intelligence**, 94(1-2), 79–97
- Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. **Engineering Applications of Artificial Intelligence**, 22(7), 979–991
- Liang, Y.-C., & Smith, A.E. (2004). An ant colony optimization algorithm for the redundancy allocation problem (RAP). **IEEE Transaction on Reliability**, 53(3), 417–423
- Madureira, A., Pereira, I., Pereira, P., Abraham, A. (2014). Negotiation mechanism for self-organized scheduling system with collective intelligence. **Neurocomputing**, 132, 97–110
- Mezgebe, T.T., El Haouzi, H.B., Demesure, G., Pannequin, R., Thomas, A. (2018). A Negotiation Scenario Using an Agent-Based Modelling Approach to Deal with Dynamic Scheduling. **In Service Orientation in Holonic and Multi-Agent Manufacturing** (pp. 381-391). Cham: Springer
- Monteserin, A., Amandi, A. (2011). Argumentation-based negotiation planning for autonomous agents. **Decision Support Systems**, 51(3), 532–548
- Nagarajan, M., & Sošić, G. (2008). Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. **European Journal of Operational Research**, 187(3), 719–745
- Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. **Journal of Scheduling**, 12(4), 417
- Pach, C., Berger, T., Sallez, Y., Bonte, T., Adam, E., Trentesaux, D. (2014). Reactive and energy-aware scheduling of flexible manufacturing systems using potential fields. **Computers in Industry**, 65(3), 434–448
- Pannequin, R., & Thomas, A. (2012). Another interpretation of stigmergy for product-driven systems architecture. **Journal of Intelligent Manufacturing**, 23(6), 2587–2599
- Rahwan, I., Ramchurn, S.D., Jennings, N.R., Mcburney, P., Parsons, S., Sonenberg, L. (2003). Argumentation-based negotiation. **The Knowledge Engineering Review**, 18(4), 343–375
- Rey, G.Z., Bonte, T., Prabhu, V., Trentesaux, D. (2014). Reducing myopic behavior in FMS control: A semi-heterarchical simulation-optimization approach. **Simulation Modeling Practice and Theory**, 46, 53–75
- Rosenschein, J.S., & Zlotkin, G. (1994). **Rules of Encounter: Designing Conventions for Automated Negotiation among Computers**. Massachusetts: MIT press
- Shen, W., Hao, Q., Yoon, H.J., Norrie, D.H. (2006). Applications of agent-based systems in intelligent manufacturing: An updated review. **Advanced Engineering INFORMATICS**, 20(4), 415–431
- Skorin-Kapov, J., & Vakharia, A.J. (1993). Scheduling a flow-line manufacturing cell: a tabu search approach. **The International Journal of Production Research**, 31(7), 1721–1734
- Tonino, H., Bos, A., de Weerd, M., Witteveen, C. (2002). Plan coordination by revision in collective agent based systems. **Artificial Intelligence**, 142(2), 121–145
- Valckenaers, P., & Van Brussel, H. (2016). **Design for the Unexpected: From Holonic**

- Manufacturing Systems towards a Humane Mechatronics Society.** Oxford: Butterworth-Heinemann
- Vander Veen, D.J., & Jordan, W.C. (1989). Analyzing trade-offs between machine investment and utilization. **Management Science**, 35(10), 1215–1226
- Vlassis, N. (2007). A concise introduction to multiagent systems and distributed AI. **Synthesis lectures on Artificial Intelligence and Machine Learning**, 1(1), 1-71
- Wang, H., Huang, M., Wang, J. (2018). An effective metaheuristic algorithm for flowshop scheduling with deteriorating jobs. **Journal of Intelligent Manufacturing**, <https://doi.org/10.1007/s10845-018-1425-8>
- Weiss, G. (1999). **Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence.** Massachusetts: MIT press
- Wong, T.N., Leung, C.W., Mak, K.L., Fung, R.Y.K. (2006). Dynamic shop floor scheduling in multi-agent manufacturing systems. **Expert Systems with Applications**, 31(3), 486–494
- Wooldridge, M. (2009). **An Introduction to MultiAgent Systems.** Liverpool: John Wiley & Sons
- Xiong, W., & Fu, D. (2018). A new immune multi-agent system for the flexible job shop scheduling problem. **Journal of Intelligent Manufacturing**, 29(4), 857–873
- Zattar, I.C., Ferreira, J.C.E., Rodrigues, J.G.G., De Sousa, C.H.B. (2010). A multi-agent system for the integration of process planning and scheduling using operation-based time-extended negotiation protocols. **International Journal of Computer Integrated Manufacturing**, 23(5), 441–452