



**HAL**  
open science

## CoMM: A consensus algorithm for multi-agent based manufacturing system to deal with perturbation

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

► **To cite this version:**

Tsegay Tesfay Mezgebe, Guillaume Demesure, Hind Bril El Haouzi, Rémi Pannequin, André Thomas. CoMM: A consensus algorithm for multi-agent based manufacturing system to deal with perturbation. *International Journal of Advanced Manufacturing Technology*, 2019, 105 (9), pp.3911-3926. 10.1007/s00170-019-03820-0 . hal-02126766

**HAL Id: hal-02126766**

**<https://hal.science/hal-02126766>**

Submitted on 12 May 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.

## CoMM: A Consensus algorithm for Multi-agent based Manufacturing system to deal with perturbation

Tsegay Tesfay MEZGEBE<sup>1\*</sup>, Guillaume DEMESURE<sup>1</sup>, Hind BRIL EL HAOUZI<sup>1</sup>, Remi PANNEQUIN<sup>1</sup>, Andre THOMAS<sup>1</sup>

### Abstract

The emergence of Cyber Physical System has dramatically impacted the use of traditionally centralized control system in responding to unexpected events. Rush order is quite common unexpected event in the current dynamic market characteristics and has significant perturbing ability to a centrally predictive schedule. This paper is aimed to propose a Consensus algorithm for Multi-agent based Manufacturing system (CoMM) to control the rush order and henceforth minimize a makespan. Consensus is an algorithmic procedure applied in control theory which allows convergence of state between locally autonomous agents collaborating for their common goal. Leader-Follower communication approach was used among the multi-agent to deal with the perturbing event. Each agent decides when to broadcast its state to neighbor agents and the controlling decision depends on the behavior of this state. The consensus algorithm is initially modeled by networking all contributing agents. After this, it is validated with simulation experiment based on academic full-sized application platform called TRACILOGIS platform. The results showed that the consensus algorithm has significantly minimized the impact of rush order on makespan of manufacturing orders launched on a system.

**Key words** Consensus algorithm, State, Multi-agent system, Convergence, Rush order, Makespan

---

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

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

## 1. Introduction

The requirement for short life cycle products is urging manufacturing industries to focus on control systems that network physical and computational components. This demands continuous search for control systems that balance between local and global decisions within their system. However, efficient utilization of resources and time in such dynamically changing environment is not easy due to complex & delayed communication between components. Consequently, these industries are adapting smart transformation process of services supported by autonomous and communicative decision entities<sup>2</sup>. Making these decisional entities to decide autonomously and realize their online monitoring has been used as the best control approach for the previous decades. It is still one of the significant solution searches for resource allocation and sequence management. Nevertheless, as it is presented in [2, 3], drawbacks such as limited capacity to predict process outcomes, own goal orientation, least commitment about other entity's offer etc. are critical issues under this control approach.

These challenges have pushed industries and research institutes to look for alternative control approaches that assimilate locally autonomous decision entities with some other coordinating entities. This seizes collaborative decision support system. In such organizational sociability, decision entities must negotiate with their neighbor entities and maintain some social relationships at their operational and structural levels [4]. As part of this sociability, this paper is aimed to investigate the application of consensus algorithm for multi-agent based manufacturing system (CoMM) to control unexpected event called rush order. Rush order is an implicit customer's priority request for his/her products to be delivered very quickly [5, 6]. Literatures have treated rush order as a special disturbance that disrupts performance of an entire shop floor [7]. Accepting this perturbation could trade-off manufacturing lead time and resource utilization if an immediate action is not steadily taken. Immediate to its happening, it must be inserted to a current schedule by making optimal rearrangement of all resources available within a system. Meanwhile, local and global decision entities must work cooperatively in order to minimize myopic behavior of locally autonomous entities during such decision change.

The consensus algorithm was implemented using leader-follower communication approach with time varying communication topology. The leader agents coordinate their group of followers using a state in relation with their behavior. Convergence state is used to indicate the agreement of agents to work together towards a common goal. The proposed algorithm is simulated considering an industrial problem and a full-sized academic application test-bed platform. The results contribute some findings on the field of negotiation and cooperation among the multi-agents by considering not only their own state but also the state of neighbors after occurrence of the unexpected event. It reduces the impact of the rush order on an overall production performance and hence

---

<sup>2</sup>Decision entity is a building unit (products, machines, order, etc.) within manufacturing system with capability to show physical & informational communication behavior, make decisions, and act for decision [1].

minimizes a makespan. The rest of the paper is organized as follow: Section 2 surveys multi-agent systems (MAS) based cooperative decision and the way agents reach their consensus. Two sub-sections with cooperative decision approach in the first sub-section and consensus theory based control algorithm in the second sub-section are addressed. Section 3 presents and formalizes the industrial problem. Section 4 proposes the consensus algorithm for the problem statement presented in section 3. In addition to this, it adapts the applicability of the algorithm by designing a new informational state. Section 5 simulates & validates the proposed algorithm and discusses its experimental results. It uses the application platform as a case study. Finally, section 6 concludes based on the results obtained and forwards its future work on how the simulation experiment could be implemented in an actual industrial environment.

## 2. Survey of related works

### *2.1 Surveys related to cooperative decisions*

The developments in communication and sensing have evolved significant interest and research activity in the area of cooperative MAS. For example, in intelligent manufacturing system, it is well accepted that MAS based control system is a good way to deal with disturbances and decision changes. Agents in such control system are characterized by their autonomy in which decisions at the time of necessity are made by these agents. This helps to create robust system with respect to time-varying communication. Without being exhaustive, some intelligent control systems based on MAS could be cited: Leitão [8], Isern et al. [9], Xiong and Fu [10] etc. In all these citations, it is addressed that the applicability of MAS is characterized by their large scale in terms of number of agents, dynamic nature, and complex functionality. It is presented in Isern et al. [9] that, even though agents are perceived as autonomous entities, they are also members of a society. They have to exchange information with other group of agents and maintain some relationships at an organizational level. Consequently, the mere presence of multiple agents makes an environment appear dynamic from the point of view of each agent with the control system they follow.

This shows that unless the multi-agents are cooperative, the required performance within a manufacturing system could not be efficiently achieved. Even though individual agents can generate and execute their plans independently, conflicts may arise and hence they need to coordinate their course of action. This cooperation is proved by different authors. For instance, Caridi and Cavalieri [11] have presented that optimal global performance could be achieved if several decision making agents, distributed inside manufacturing cells, cooperate and interact for common offer. Wooldridge [12] has added negotiation among agents is foreseen to host a robust-predictive-reactive scheduling and also to tackle myopia. Myopia, a word from [13], is drawback of autonomous agents during their execution process. These include limited capacity to predict process outcomes, least commitment about neighbor agents etc. Hence, reducing these drawbacks must be the target of the multi-agents and minimizing these drawbacks is highly dependent on structural, operational, and behavioral designs, Table 1.

Tonino et al. [14] illustrated the investigations of different agent negotiation approaches including game-theoretic, heuristic-based, and argumentation-based approaches. Three of them emphasize the importance of exchanging information between agents in order to mutually influence their behaviors. 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 [15, 16]. But it is known for unbounded computational resources as limitation. To overcome this limitation, the heuristic approach has come with the principle of *produce good enough rather than optimal outcomes* [17]. 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 [18]. Subsequently, argumentation-based negotiation approach has evolved to overcome the knowledge limitations of agents in game-theoretic and heuristic negotiation approaches. As it has been surveyed by Rahwan et al. [19], argumentation-based negotiation approach allows agents to exchange additional information or to argue about their beliefs & other’s mental attitudes during their negotiation time. Agents accept, reject, or critique an offer proposed by other agent until they agree on this offer. Meanwhile, argumentation-based negotiation approach has been gained increasing popularity for its potential ability to overcome the limitations of other conventional approaches. As a continuation of developing negotiation protocols for improving manufacturing system’s performance, Xiong and Fu [10] have presented the role of cooperative multi-agents. They designed an immune multi-agent scheduling system to solve a flexible job shop scheduling problem.

**Table 1**Controlling methods of multi-agent

Design	Control approach	Role	Examples	Citations
Structural cooperation	Task determination through command cooperation	Overall system view	Robotic system	[4, 13]
Operational cooperation	Belief, Desire, and Intention alteration through communication protocols	Sharing equal rights by direct information exchange	Negotiation & consensus: member debate to make law	[11, 14, 19]
Bio-inspired (behavioral cooperation)	Modification of route using pheromones	Food searching: pheromone based route exploration	Ant and Termite colonies	[20–22]
Physical-inspired	Machines emit attractive fields to attract jobs	Potential field	Electrical fields	[23]

Mezgebe et al. [24] have also proposed negotiation model considering smart product agents scheduled to be processed on different resource agents. The communication protocol was fully controlled by product agents and role of Radio Frequency Identification (RFID) technology was used to help detect disturbances and send back to product agents for its management. However, the role of giving and accepting consent to provide one's state and follow a common protocol was not well addressed. That is why consensus paradigm is explored to formalize such consent and hence to share one's state for the interest of a whole.

### ***2.2 Surveys related to consensus in cooperative decisions***

Literatures on MAS have revealed that negotiation and cooperation are good elements of networked agents to reach at their final offer called consensus. Consensus is an algorithmic procedure to negotiate among groups of agents and hence asymptotically reach an agreement for the best interest of whole [25]. It pursues certain quantities of interest with or without predefined common orientation of individual agents. Currently, it becomes applicable in mathematics, physics, control theory etc. and its applicability is revealed in synchronization of collaborative decision support systems [26]. These applicability's include rendezvous in the space [27, 28], flocking [29, 30], decentralized scheduling in networking domain [31], sensor networks [32], coupled oscillators [33] among many others. In all these application domains, agents not only update their own state but also the state of neighbors. This helps to design an update law and hence to converge to a common value based on the designed update law.

Consequently, these and other researchers have been worked to design different control laws and protocols. Dimarogonas and Kyriakopoulos [27] for example, have proposed discontinuous and time-invariant non-holonomic control law so as to examine stability of a system. They described convergence of designed multi-agent system relies on connectivity of a communication graph that represents an inter-agent communication topology. Lee and Spong [29] on their flocking study have proposed a provably-stable flocking control law. It was proposed considering a stable flocking of multiple inertial agents (keeping their shape and velocity). It ensured that internal group formation is exponentially stabilized to a desired shape while all agents' velocities converge to a centroid velocity with time-invariant evolution. Moore and Lucarelli [31] have also proposed consensus variables with nearest-neighbor communication topology to solve a decentralized adaptive scheduling of tasks or mission timing problems. On the other hand, Ren and Beard [34] have hypothesized that to reach consensus among group of agents, there must be a shared variable of interest called informational state. Wang and Shao [35] have strengthened the role of informational state by developing consensus protocol where state of neighbor agent is continuously controlled by state of initial agent. All the above reviews indicate that there has been a great extent of concern on consensus algorithm to solve conflicting problems among own goal oriented agents.

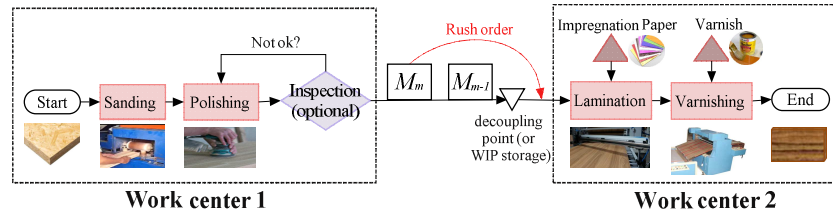
To sum-up the survey, many researchers have given due attention to the applicability of sociability based multi-agent systems for controlling communication problems. However, its applicability for manufacturing control

problems and specifically to scheduling problems was not well investigated and studied. It was highly comprehended with challenges such as difficulty to guarantee minimal level of information & operational performance, difficulty of synchronization, lack of commercial platforms etc. Jimenez et al. [4], for instance, have confirmed the difficulty of synchronization of the level of autonomy of contributing entities. Therefore, further exploration on different MAS based control algorithm that help synchronize these agents and hence create robust system with respect to time-variant communications is becoming another research area [36]. And to the best of author(s) knowledge, the consensus algorithm from consensus theory has rarely been adapted to decision making algorithm or not yet implemented in flexible manufacturing systems (FMS). Thus, this paper presents the application of the algorithm to minimize non-value adding activities created after unexpected disturbance and hence minimize makespan of a manufacturing process.

### 3. Problem statement

A make-to-order impregnation and lamination flow shop, inspired from a particle board manufacturing factory called Maichew Particle Board Manufacturing Company, is considered to set the problem statement, Fig. 1. Sanding of raw board, polishing & inspecting, lamination, and varnishing are its sequential operational activities. Its products are supplied to subsequent job shop or external customers to serve as raw materials for office furnitures, kitchen cabinets, computer desks etc. The flow shop has a capacity to produce different standard and variety of products for different manufacturing orders by changing its setup for each order. Meanwhile, the company's planning department proposes a weekly predictive (centralized) schedule for each customer orders (each with their own product types) the flow shop receives. If perturbation and decision changes are not noticed, this predictive schedule proposed at the beginning of the execution process realizes the full completion of all orders received with best rewards.

However, it often suffers from many external and internal perturbations & changes and the following two are set as the most common one: (a) last minute's rush order due to highest priority and (b) extra buffer (or higher buffer size) at decoupling point due to the rush order. The high buffer size, which in turn results higher work-in-progress inventory, encounters because each product of each manufacturing order chooses a destination with shortest completion time scheduled at the central level.



**key:**  $M_{m-1}$  is a preceding manufacturing order and  $M_m$  is its successor manufacturing order

**Fig. 1** Flow chart of impregnation and lamination shop floor

Accepting such perturbations could trade-off manufacturing lead time and resource utilization if an immediate action is not steadily taken. For example, it may lead to non-value adding activities such as waiting time that could be observed in the work centers presented in Fig. 1

### 3.1 Formalizing the industrial problem

To formalize the industrial problem, it is considered that the flow shop has received  $m$  manufacturing orders  $M_1, M_2, \dots, M_m$  each with its own product types  $P_i$  and due date  $dd_1, dd_2, \dots, dd_m$  respectively. Meanwhile, the below set of indexes, notations, variables, and parameters have been used throughout this paper to design the required control algorithm.

<i>Indexes and notations</i>			
$r$	Resource index	$\widetilde{M}_m$	Rush order
$i, j$	Product indexes	$la_m$	Leader (or last product) agent of manufacturing order $m$
$m$	Manufacturing order index	$\widetilde{la}_m$	Leader (or last product) agent of rush order
$k$	Operation index	$\Phi_s$	Intention based decision (and/or conflict) node
$n$	Number of agents	$t_s$	Time where event based perturbation or decision change happens
<i>Variables</i>			
$c_i$	Completion date of product $i$	$c_m$	Completion date of manufacturing order $m$
$C_{\max}$	Makespan	$W$	Waiting time of product $i$ in a system
<i>Parameters</i>			
$M$	Set of manufacturing orders	$w_{kir}$	Waiting time of product $i$ after its processing in resource $r$ or route completion in a system <sup>3</sup> until its batch members couple it
$R$	Set of resources within a system	$\varepsilon_{kir}$	Exit/release time of product $i$ from resource $r$
$R_{ki}$	Set of resources which perform operations $k$ of product $i$	$t_{wr}$	Actual working time used by resource $r$
$O_{im}$	Set of operations of product $i$ of order $m$	$t_{vr}$	Maximum available time of resource $r$
$o_{ki}$	$k^{\text{th}}$ operation of product $i$	$\lambda$	Arrival rate of product $i$ to resource $r$
$m_r$	$m^{\text{th}}$ manufacturing order to be processed in resource $r$	$\tau$	Exit/release rate of product $i$ from resource $r$

---

<sup>3</sup> System defines an entire working environment of the flow shop

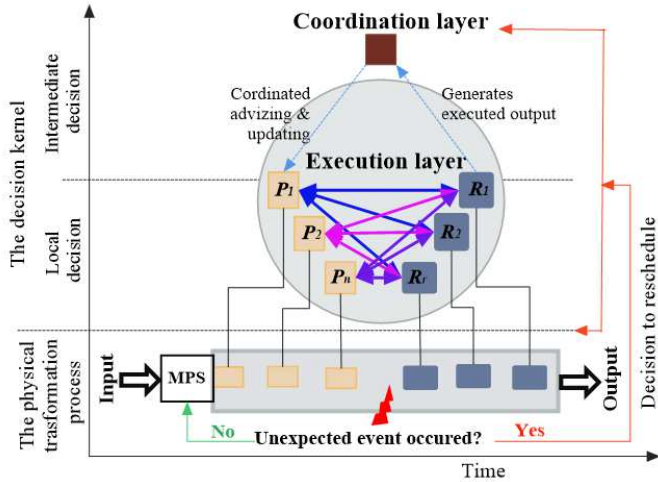


$s_{kir}$	Setup time for operation $k$ of product $i$ in resource $r$	$dd_m$	Due date of manufacturing order $m$
$\mu$	Current time to estimate a completion date	$v_i$	Intention of agent $i$
$t_{kir}$	Moving time of product $i$ to resource $r$	$x_i$	Informational/convergence state of agent $i$ at time $t$
$a_{kir}$	Arrival time of product $i$ to resource $r$	$x_j$	Informational/convergence state of neighbor agent $j \in N_i$
$p_{kir}$	Processing time of product $i$ in resource $r$	$u(t)$	Control action required to be used by each agent
$q_{kir}$	Queuing time of product $i$ before its operation $k$ in resource $r$	$y_{ir}$	Binary variable
$Tard_i$	Tardiness of product $i$		

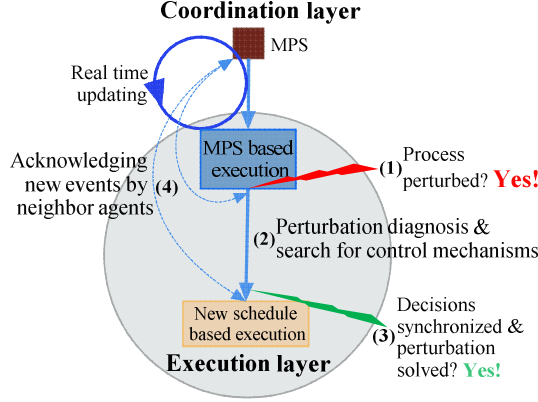
It is a threshold measure ranging from zero to large number that helps to see the effect of unexpected event.

$\beta$  It compares with  $u(t)$  such that  $\begin{cases} \text{if } \beta \text{ is set to large, } u(t) \rightarrow 0 \\ \text{if } \beta \text{ is set to be closer to zero, otherwise} \end{cases}$

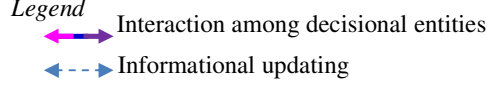
Three groups of decision entities namely product, resource/machine, and coordinating entities are considered during the problem formulation. They cooperate to sustain a Master Production Schedule (MPS) developed centrally. In order to update the cooperation & communication between these entities, a pyramidal inter-agent interface (or decision kernel in Fig. 2(a)) is used by bridging the local entities with coordinating entities.



(a) Framework for hybrid manufacturing process and meta-model decision kernel



(b) Synchronization of two control layers

**Fig. 2** MAS based cooperative decision model

**Product entities** ( $P_i$ ) are the decision entities, within the execution layer in Fig. 2(a), constructed with different product types for different manufacturing orders. They have physical part and informational part. The physical part is the hardware that is in charge of the correct execution of configuration and manufacturing operations. The informational part is in charge of transmission of manufacturing orders to resources, reasoning capabilities, and validation of key performance indicators (KPI) within a manufacturing process, Fig. 2(b). They are the drivers of decision changes at the time of perturbations if the decisional system is needed to stay in a centralized or distributed control mode. Each product entity computes its completion date  $c_i$ , “Eq. 1” by minimizing a non-value adding activities in particular waiting time ( $W$ ). This gives the completion date of each manufacturing order as it is computed in “Eq. 2”. Hence, makespan is obtained from “Eq. 2” as it is modeled in “Eq. 3”.

$$c_i = \mu + \sum_{k \in O_m} \sum_{r \in R_{ki}} y_{ir} \cdot (s_{kir} + p_{kir} + t_{kir}) + W, \quad \forall i \in M_m \quad (1)$$

$$W = \sum_{k \in O_m} \sum_{r \in R_{ki}} y_{ir} \cdot (q_{kir} + w_{kir}), \quad \forall i \in M_m$$

$$c_m = c_{ta} = \max_{i \in M_m} (c_i) \quad (2)$$

$$C_{\max} = \max_{m \in M} (c_m) \quad (3)$$

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

Subjected to

$$(1) c_{ik} \geq c_{i(k-1)} + p_{kir}, \quad \forall i \in M_m, r \in R_{ki}, \forall k \in O_{im} \quad (3a)$$

$$(2) W \leq dd_i - (\mu + \sum_{k \in O_m} (t_{kir} + p_{kir})), \quad \forall i \in M_m, r \in R_{ki} \quad (3b)$$

$$(3) q_{kir} \begin{cases} > 0, \text{ if } \lambda > \tau, \forall i \in M_m, r \in R_{ki}, k \in O_{im} \\ \equiv 0, \text{ otherwise} \end{cases} \quad (3c)$$

$$(4) w_{kir} \begin{cases} > 0, \text{ if } q_{kir} > 0, \forall i \in M_m, r \in R_{ki}, k \in O_{im} \\ \equiv 0, \text{ otherwise} \end{cases} \quad (3d)$$

$$(5) \text{Tard}_i = \max_{i \in M_m} (c_i - \text{dd}_i, 0) \Rightarrow \text{Tard}_m = \max_{m \in M} (C_m - \text{dd}_m, 0) \quad (3e)$$

$$(6) \sum_{m \in M} m_r = 1, \forall r \in R \quad (3f)$$

$$(7) \sum_{r \in R_{ki}} y_{ir} = 1, \forall k \in O_{im} \quad (3g)$$

$$(8) \{c_i, C_{\max}, W\} \geq 0 \quad (3h)$$

$$(9) y_{ir} \in \{0, 1\} \quad (3i)$$

Constraint “3a” indicates precedence constraint such that in order a product  $i$  to be processed in resource  $r$ , it must initially be processed in resource  $r-1$ . Constraints “3b-3d” present a quick decision at decision node  $\Phi_s$  by product entities to minimize their waiting time after occurrence of unexpected event. Constraint “3e” also presents no product tardiness if its completion date ends before a due date. Constraint “3f” ensures only one manufacturing order and/or product is executed on one resource at a time. Constraint “3g” presents the requirement of  $r$  resources to perform  $k$  operations of product  $i$ . Constraint “3h” ensures that all variables are non-negative. Constraint “3i” presents a binary variable which is set to 1 if product  $i$  is processed on  $r \in R_{ki}$ , 0 otherwise.

**Resource/machine entities ( $R_r$ )** are the other physical elements of the execution layer in Fig. 2(a) configured with hardware and control parts. The hardware part executes manufacturing operations sent by product entities and the control part controls the hardware operations. In case of unpredictable bottleneckness of either resource, others have the capability to perform all operations left unprocessed. Meanwhile, each resource is responsible to perform additional operation  $k+1$  for a single product of either manufacturing orders. Changeovers of production from one product type to another associates setup times and resource entities are expected to fully utilize their capacity. Taking this hypothesis, resources calculate their utilization ( $\rho$ ) based on the model presented in “Eq. 4”. Product’s prioritization has to be validated by recalculating this utilization rate for their every acceptance.

$$\rho = \frac{t_{wr}}{t_{vr}} * 100\% \quad (4)$$

This computation is necessary in order to process products if their arrival rate does not affect the computed utilization rate. If not, the control part of the resource queries all sequentially approaching products to smartly wait in buffer zone until its setup is completed. This makes product entities to smartly wait by estimating their queuing time and route forward for processing after the resource’s setup is adjusted.

**Coordinating entities** are set of centralized coordinators (or informational control system, CS) within the coordinating layer in Fig. 2(a). They prepare and dispatch an MPS to be executed by the executing agents in the execution layer. Product and resource entities stipulate these coordinating entities that statically advice and update adaptive cooperation to have a shared schedule among all the decision entities. Meanwhile, these coordinating entities manages a myopia [13] by cooperating with the local decision entities & dynamically creating bilateral output generating communications, Fig. 2(b).

#### 4. Proposal: Consensus algorithm to deal with rush order

The performance of CoMM has been shown by applying to flexible route of product entities. It is aimed at maintaining the same performance of planned schedule after a disruptive rush order event has occurred. To provide such adaptability and ensure coherency to a global objective, agents have used a Leader-Follower based communication approach. Leader-Follower consists of a group of leader agents with some amount of power to facilitate cooperation among a large group of their followers [37].

##### 4.1 Preliminaries on consensus

A communication topology is represented using graph;  $G = (V_n, E_n)$  for  $n$  nodes, where  $V_n$  is the set of vertex and/or decision entities ranging from 1 to  $n$ ,  $E_n \subseteq V_n \times V_n$  is the set of communication edge ranging from 1 to  $n-1$ . All edges are said connected if agents  $\{i, j\} \in E_n \Leftrightarrow \{j, i\} \in E_n$ . The graph theory also defines a Laplacian matrix,  $L_n = D_n - A_n$ ; where  $A_n = \{a_{ij}\} \in \mathbb{R}^{n \times n}$  is the adjacency matrix,  $D_n = \{d_i\} \in \mathbb{R}^{n \times n}$  is the degree matrix with  $d_i = \sum a_{ij}, \forall i \neq j$ , and  $a_{ij}$  is the  $(i,j)$  entry of the adjacency matrix at time  $t$ ,

$$\text{Such that } \begin{cases} a_{ij} = 1 \Leftrightarrow a_{ji} = 1, \text{ connected, } \forall i \neq j, (i, j) \& (j, i) \in E_n \\ a_{ij} = 0 \Rightarrow a_{ji} = 0, \text{ otherwise; } \forall i = j, (i, j) \& (j, i) \in E_n \end{cases}$$

The connectivity among agents during their communication is also proved by respecting interdependence of different performance measures that help them to improve their adaptability, Algorithm 1

##### Algorithm 1: Dependence of agents' state

*$n$  communicating agents, with  $n \geq 2$  and  $i \neq j$ , must converge to a common goal by designing state  $x_i$  and considering interdependent performance measuring parameters, "Eq. 5a". Likewise, an agent 'i' has to share state of its neighboring agent 'j' and use this state as an estimator of its future behavior for execution on a time varying communication topology, "Eq. 5b".*

$$u(t) = f(x_i, x_j), \quad \forall i \& j \in M_n, i \neq j \quad (5a)$$

$$x_i(t+1) = g(x_i(t), u_i(t), u_j(t)), \quad \forall j \in N_i \quad (5b)$$

This dependence would be attained if all decision entities are able to cooperate with each other and evaluate the impact of unexpected event on neighboring agents, Algorithm 2.

**Algorithm 2: Consensus priority protocol**

**Input:**  $\{(t_{kir} + p_{kir}), (s_{kir} + t_{kir} + p_{kir} + W), dd_i\}, \forall i \in M_m, r \in R_{ki}, \forall k \in O_{im}$

Set  $n$  communicating agents with  $n \geq 2$  and  $i \neq j$

**For**  $i = 1, 2, \dots, n$  and  $j = 1, 2, \dots, n$ , **do**

**If** intention based conflict between  $i$  &  $j$  in decision node  $\Phi_s$ ,

    Compute state  $x_i$  and  $x_j$  & exchange the computed state

    Compute  $u(t)$ , “Eq. 6”;

**If**  $u(t) > \beta(\text{threshold})$ ;

        Execute a priority offer requested by agent  $i$ ,

**Else** flout the priority

**End if**

**End for**

**Output:**  $c_m = c_{ia} = \arg \max_{i \in M_m} \{c_i\}, \forall m \in M$

Each agent during the protocol considers three operational characteristics,

- ❖ The information that agent  $i$  has about the probabilistic occurrence of an event (the rush order in our case)
- ❖ State  $x_i$  that agent  $i$  would wish to achieve at minimal waiting time
- ❖ Intention that agent  $i$  has agreed to cooperate with others in order to minimize the impact of an event on its makespan  $C_{max}$

Hence, a consensus model [34] that guarantees these characteristics and a convergence for collective decision by all agents is given as in “Eq. 6”.

$$u(t) = - \left( \sum_{j \in N_i} a_{ij} [x_i(t) - x_j(t)] \right), \forall i, j \in n \quad (6)$$

Alternatively, “Eq. 6” can be rewritten in matrix form as in “Eq. 7”

$$u(t) = - \{L_n(t)x(t)\} \quad (7)$$

Where  $x = [x_1, x_2, \dots, x_n]^T$  is the information state of each agent  $i$ ,

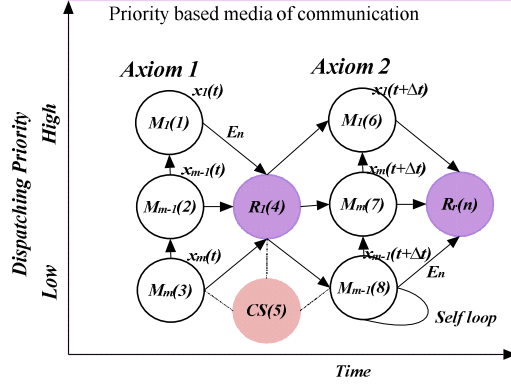
$L_n(t) = [l_{ij}(t)] \in R^{n \times n}$  is non-symmetrical Laplacian matrix.

In “Eq. 6”, each agent  $i$  must create its own informational state  $x_i$  and then after reach consensus with neighbor agents’  $j$  to approve a requested priority offer. To create the information state, each agent sets an event time as  $x_i(0)$  where it initially meets with its neighbor agents  $j$  [34]. This promises that the common value is converged combination of all the initial informational states. After all the  $n$  agents have shared their interest, consensus is reached when the variability of interest among them approach towards zero, “Eq. 8”.

$$|x_i(t) - x_j(t)| \rightarrow 0, \forall i, j \in M_m \quad (8)$$

#### 4.2 Design and adaptation of informational state

The  $m$  manufacturing orders with their communication and execution topology shown in Fig. 3 are considered to design and adapt our state.



**Fig. 3** Communication topology among different entities

Initially, manufacturing orders are made to enter a system based on shortest processing time (SPT) first dispatch list presented in Axiom 1 below.

Axiom 1: Initial dispatch list

$(M_m) < (M_{m-1}) < (M_{m-2}) \dots < (M_1)$  such that the last operation  $k$  of product  $i$  of  $M_{m-1}$  has to be completed in next resource  $r$  before the last operation  $k$  of product  $i$  of  $M_m$  is started and same for others.

While manufacturing orders route according to this dispatch list, it is taken that  $M_m$  becomes a rush order after  $t + \Delta t$ , Axiom 2.

Axiom 2: Modified dispatch list

$(M_{m-1}) < (M_m) < (M_{m-2}) \dots < (M_1)$  such that the last operation  $k$  of product  $i$  of  $M_m$  has to be completed in next resource  $r$  before the last operation  $k$  of product  $i$  of  $M_{m-1}$  is started and same for others

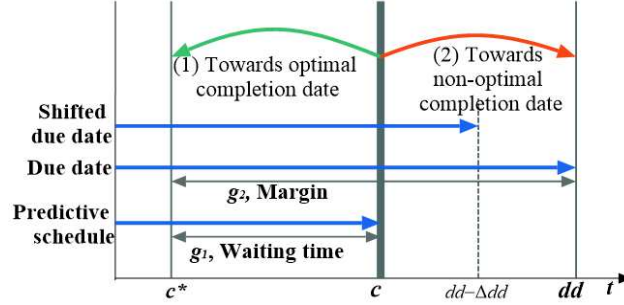
In order to satisfy Axiom 2, the future impact of the rush order on the cost of waiting time of preceding manufacturing order has to be evaluated. Deciding when a leader agent of  $\widetilde{M}_m$  would broadcast its intention to leader agent of  $M_{m-1}$  sharing same decision node is the required consensus. This is necessary because if the leader agent of  $M_{m-1}$  passes decision node  $\Phi_s$  before that of the rush order, it increases its completion date without value addition. This happens because the coordinating agents would not allow this agent to precede the next decision node  $\Phi_{s+1}$  before that of the rush order. Consequently, each leader agent of each manufacturing order competes to minimize its waiting time and hence minimizes its completion date presented in “Eq. 9”. From Algorithm 1,

$$c_i = f(W) \text{ and } c_i = f(c_j), \forall (i \neq j) \in M_m, k \in O_{imp} \ r \in R_{ki}$$

$$\text{This gives, } c_{la} = \arg \max_{i \in M_m} \{c_i\}, \forall m \in M \quad (9)$$

Let  $c_i^*$  be lower bound (with  $c_i^* = \mu + \sum_{i \in M_m} (t_{kir} + p_{kir})$  &  $c_i^* < dd_i$ ) of the actual

completion date  $c_i$ ;  $g_1$  and  $g_2$  be a gap of this lower bound from the actual completion date and a margin respectively, Fig. 4.



**Fig. 4** System resequencing with respect to rush order

As shown in Fig. 4, the rushed order shifts down its initial due date to a new due date  $dd - \Delta dd$ . This implies that its completion date reduces at least by the processing time of the preceding manufacturing order. On the contrary, the completion date of the preceding manufacturing order increases by the same amount. However, the target of products of each manufacturing order is to minimize its waiting time ( $W$ ) in order to achieve its best individual performance. Thus, consensus allows ensuring a minimal tradeoff between agents' state which they want to improve their own performance. Consequently, the informational state that makes each agent to converge towards a common value (invariant centroid state)  $d \in \mathbb{R}_+ \setminus \{0\}$  so that "Eq. 8" would be satisfied is computed as,

$$x_i(t) = \frac{g_1}{g_2} = \frac{(c_i - c_i^*)}{(dd_i - c_i^*)}, \forall i \in M_m \Leftrightarrow x_{la}(t) = \frac{(c_{la} - c_{la}^*)}{(dd_{la} - c_{la}^*)} \quad (10)$$

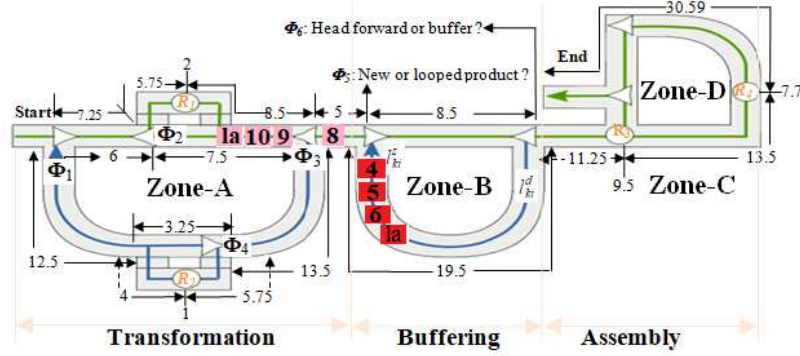
$$\text{Where } c_{la}^* = c_i^* + \alpha p_{kir}, \forall \alpha \in \mathbb{N} \setminus \{0\}$$

This indicates that the completion date of each product agent of the rush order must close to the lower bound value so as to obtain  $x_i \approx 0, \forall i$ . To compare the impact of the calculated state, percentage of closeness towards zero could be used with an objective to reward lowest percentage values, Sect. 5.3.1. For instance, as shown in the first option of Fig. 4, if the completion date could be able to approach towards  $c^*$ , the percentage of closeness declines to zero.

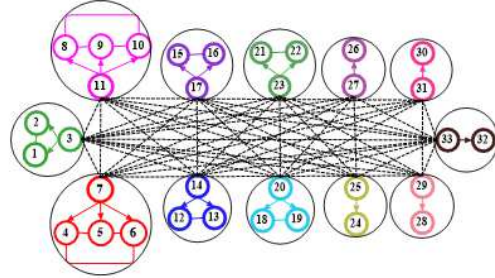
## 5. Simulation Experiments

### 5.1 Physical system description

To validate the applicability of the proposed algorithm, an academic application platform called TRACILOGIS test-bed platform, Fig. 5(a), is used as case study. The sanding, polishing & inspection, lamination, and varnishing sequential activities of a shop floor presented in Fig. 1 are transposed to this platform for simulation. This platform inscribes to centrally predictive, distributed reactive, and hybrid control systems through its smart and cooperative agents. It is composed of extensive system of networks; sensors, actuators to automata, Radio Frequency Identification (RFID) sensors, and Production Line Controllers (PLC). It has four zones: Zone-A for transformation, Zone-B for buffering, Zone-C for colored pallet (materials to be assembled) sorting, and Zone-D for assembly/disassembly.



(a) Operation times (in seconds) and positions of products within the platform



(b) The leader-follower communication topology among product agents

Legend

- Forward route, → Backward or buffer route
- Informational messaging among leader & their followers
- Cooperative messaging among leader agents

**Fig. 5** The physical system and its communication topology

If unexpected disruptive events (e.g., the rush order) occur, Zone-B helps to minimize cost penalties for trade-off between actual and planned completion time of products. It sets aside certain products to make other products take priority over a manipulator for example. Meanwhile, it allows for buffering of semi-processed products as in a real production system. As it is shown in Fig. 5(a), Dynamic looping time ( $l_{ki}^d$ ), static looping time ( $l_{ki}^s$ ), and number of looping by product  $i$  ( $\eta_{ki}$ ) in Zone-B have been used to model this buffering such that,

$$\eta_{ki} \begin{cases} \geq 2 & \text{if rush order is not fully satisfied} \\ = 1 & \text{otherwise} \end{cases} \quad \forall i \in M_{m-1}, l_{ki}^d = f(\eta_{ki})$$

$$l_{ki}^s \begin{cases} > 0 & \text{if rush order is not fully satisfied} \\ = 0 & \text{otherwise} \end{cases}$$













Product, resource, roller conveyor, PLC, and RFID as coordinating agents are used to launch the platform. The  $R_r$  notations in Fig. 5(a) denote the four resource entities  $R_1$ ,  $R_2$ ,  $R_3$ , and  $R_4$  for line marking, point marking, pallet assembling, and pastille assembling respectively. Each resource has a processing time of 2, 1, 9.5, and 7.7 seconds for single operation of each product  $P_i$ . In case product  $P_i$  of manufacturing order  $M_m$  requires, for instance, two line marking operations, resource  $R_1$  is responsible to perform all the additional operations. Moreover, the static looping time at the end of Zone-B is managed by the RFID as coordinating agent.

### 5.2 Input data

A central schedule for 12 manufacturing orders  $M_1, M_2, \dots, M_{12}$  each with their own product types is prepared based on real processing time of the platform, Table 2 and Table 3. They are made to enter the platform based on *Axiom 1*:  $\{(M_{12}) < (M_{11}) < (M_{10}) < (M_9) < (M_8) < (M_7) < (M_6) < (M_5) < (M_4) < (M_3) < (M_2) < (M_1)\}$ . While they route according to this dispatch list,  $M_3$  becomes rush order at time  $t = 75$  seconds (*Axiom 2*):  $\{(M_{12}) < (M_{11}) < (M_{10}) < (M_9) < (M_8) < (M_7) < (M_6) < (M_5) < (M_4) < (M_2) < (M_3) < (M_1)\}$ . To fulfil this new dispatch list, product agents have used the Leader-Follower based MAS approach. The leader agent of  $M_2$  ( $P_7$ ) is made to be precedence communication neighbor of the leader agent of  $\widetilde{M}_3$  ( $P_{11}$ ) and successor neighbor of that of  $M_1$  ( $P_3$ ), Fig. 5(b). The communication between product agents of these two manufacturing orders is taken to emulate and validate our algorithm. During the emulation, it is considered that all the products of successor manufacturing orders behave same as that of the currently interacting agents. Meanwhile, from the six decision nodes  $\Phi_3$  shown in Fig. 5(a), decision nodes  $\Phi_5$  and  $\Phi_6$  are chosen to test the consensus. The communication on the other decision nodes behaves same evolution as in the state based decisions made in  $\Phi_5$  and  $\Phi_6$ .

**Table 2** Intelligent components and product types of each customer order  $C_o$

	Product components					Remark
						
	Platter	Line	Point	Pastilles	Pallets	
<b>Product type (<math>P_i</math>) of each customer order <math>C_o</math></b>						Setup is required if there is position change of either components
	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	

**Table 3** Centrally predictive schedule for each manufacturing orders

$C_o$	$M_m$	$P_i$	Processing time (in sec.) of each product $P_i$ on resource $R_r$				Planed $c_i$ (in sec.)	$dd_m$ (in sec.)
			$R_1$	$R_2$	$R_3$	$R_4$		
$C_1$	$M_1$	$P_1$	2	-	19	15.4	132.75	250
		$P_2$	2	-	19	15.4	153.75	
		$P_3$	2	-	19	15.4	174.75	
$C_2$	$M_2$	$P_4$	2	1	28.5	23.1	289.25	600
		$P_5$	2	1	28.5	23.1	329.25	
		$P_6$	2	1	28.5	23.1	369.25	
		$P_7$	2	1	28.5	23.1	409.25	
$C_3$	$M_3$	$P_8$	2	1	38	30.8	462.82	800
		$P_9$	2	1	38	30.8	493.32	
		$P_{10}$	2	1	38	30.8	523.82	
		$P_{11}$	2	1	38	30.8	554.32	
$C_4$	$M_4$	$P_{12}$	4	1	38	30.8	599.32	950
		$P_{13}$	4	1	38	30.8	630.32	
		$P_{14}$	4	1	38	30.8	661.32	
	$M_5$	$P_{15}$	4	1	38	30.8	759.57	1200
		$P_{16}$	4	1	38	30.8	791.32	
		$P_{17}$	4	1	38	30.8	823.07	
	$M_6$	$P_{18}$	4	1	38	30.8	854.82	1350
		$P_{19}$	4	1	38	30.8	886.57	
		$P_{20}$	4	1	38	30.8	918.32	
	$M_7$	$P_{21}$	4	1	38	30.8	950.07	1400
$P_{22}$		4	1	38	30.8	981.82		
$P_{23}$		4	1	38	30.8	1013.57		
$C_5$	$M_8$	$P_{24}$	4	2	38	30.8	1111.82	1600
		$P_{25}$	4	2	38	30.8	1142.82	
	$M_9$	$P_{26}$	4	2	38	30.8	1173.82	1700
		$P_{27}$	4	2	38	30.8	1204.82	
	$M_{10}$	$P_{28}$	4	2	38	30.8	1235.82	1800
		$P_{29}$	4	2	38	30.8	1266.82	
	$M_{11}$	$P_{30}$	4	2	38	30.8	1297.82	1900
		$P_{31}$	4	2	38	30.8	1328.82	
	$M_{12}$	$P_{32}$	4	2	38	30.8	1359.82	2050
		$P_{33}$	4	2	38	30.8	1390.82	

As it is shown in Fig. 5(b), all the leader agents negotiate with each other and share the agreed offer to their followers for implementation. As each product agent  $P_i$  has its own programmed intention, the decision process starts by sensing the cumulative sum of informational state of neighboring agents. Each product agent of the rush order computes its intention  $v_i$  and broadcasts to defined service providing agents  $j \in N_i$ .

$$v_i = [a_{kir} \ p_{kir} \ \varepsilon_{kir}], \forall k \in O_{im}, i \in M_m, r \in R_{ki} \quad (11)$$

The service providing (or responding) agents acknowledge the broadcasted intention and respond considering their current state. Subsequently, the product agents of the second manufacturing order ( $P_7, P_6, P_5, P_4$ ) are obligated to loop in Zone-B until the products of the rush order ( $P_{11}, P_{10}, P_9, P_8$ ) pass  $\Phi_5$ . This decision changing hypothesis is experimented using MatLab and/or Python simulation considering all the 11 cooperating product agents from both manufacturing orders, Sect. 5.3.

### 5.3 Results and discussion

To articulate the performance of the consensus based decision, myopic decision is taken as comparative algorithm. Agents in the platform were initially instantiated for this algorithm such that (a) resource agents are passive agents which only wait for instructions from product agents, (b) product agents calculate their production range at every move the shortest path to their final destination, and (c) product agents have no awareness on state of their neighbors. Three performance indicators namely informational state ( $x_i$ ), makespan ( $C_{max}$ ), and lateness ( $L$ ) have been considered in expressing the performance and are discussed through convergence analysis (Sect. 5.3.1), production performance indicators (Sect. 5.3.2), and sensitivity analysis (Sect. 5.3.3).

#### 5.3.1 Convergence analysis

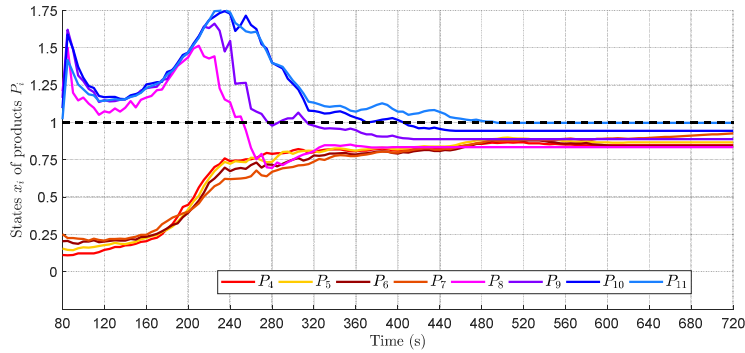
As it is stated earlier, the required consensus is mainly showed in zone-B (considering  $\Phi_5$  &  $\Phi_6$ ) of Fig. 5(a). Product agents of the rush order and that of the preceding manufacturing order have agreed to support a decision for the best interest of all members. They follow the *update intention for the best of the whole* sequencing and routing. As it is modeled in “Eq. 8”, consensus would be achieved if the *variability* of interest among all product agents approaches towards zero. Where in this case,  $x_i$  represents  $\{x_8, x_9, x_{10}, x_{11}\}$  and  $x_j$  represents  $\{x_4, x_5, x_6, x_7\}$ ,

$$\text{Such that } \begin{cases} \text{if } |x_i - x_j| \rightarrow 0, \text{ agreed} \\ \text{if not, otherwise} \end{cases}$$

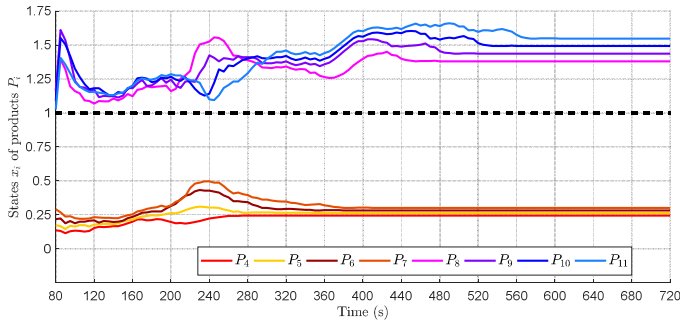
While supporting this decision, the location of product agents, Fig. 5(a), is determined by the priority request. If the sum of the difference of state value of neighbor agents is greater than a threshold  $\beta=1$ , product agents of the proceeding manufacturing order will loop in zone-B otherwise not. As shown in Fig. 6, the simulation results have shown convergence (closeness to the invariant centroid state  $x_i=1$  or the black dotted line in the figure) in consensus based decision rather than its non-convergence in myopic decision algorithm, Fig. 7. The intention of product agents of the rush order was to lower their completion date at-least by 290 seconds, Fig. 6. As a response to this intention, the preceding product agents have shared the state to show their agreement by increasing their completion date at least by 290 seconds. This continues until the rushed product agents completely pass  $\Phi_5$  in Fig. 5(a) and this is confirmed when the difference of state value of conflicting product agents start to be below the threshold  $\beta=1$ , Algorithm 2. To examine this agreement level, it is taken for instance, the state of the two leader

agents  $x_{11}$  and  $x_7$  with their last time-evolved simulation results of 0.998035 (or the minimum value) and 0.938344 (or the maximum value) respectively. “Eq. 8”, gives  $|0.998035-0.938344| = 0.05969$  which implies that the variability of interest among these product agents is close to zero; thus, convergence to the invariant centroid state  $x_i = 1$  is obtained.

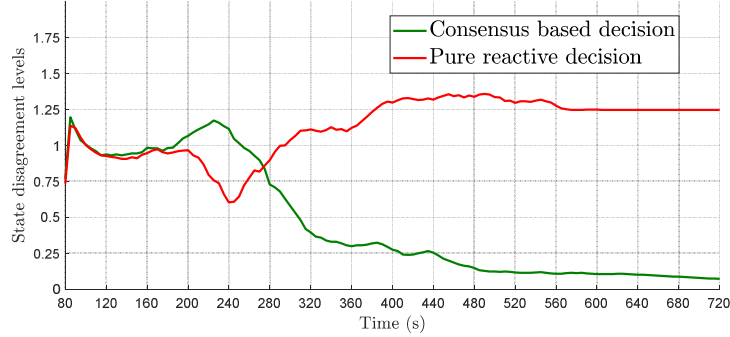
On the contrary, as it is presented in Fig. 7, each product agent decides to choose a destination with the shortest completion time scheduled centrally. Product agents of the preceding manufacturing order become myopic as they use *change the production intention for only own goal routing*. This indicates that these product agents were not interested to consider the designed state so as to satisfy the rush order and hence a conflict results. As a result, these agents have higher probability to loop in zone-B as many as  $\eta_{ki}$  without value addition. To examine this myopic decision, consider again the simulated state value of the leader agent of the rush order and that of the preceding manufacturing order which are recorded as 0.301589 and 1.547314 respectively. “Eq. 8” once more gives a value of  $|0.301589-1.547314| = 1.24573$  which implies that the variability of interest among these two agents is large enough or non-convergent. This has happened because the former agent has tried to shift-down its completion date but that of the later agent to keep its original completion date and hence initial state.



**Fig. 6** Time based evolution of state for consensus based decision



**Fig. 7** Time based evolution of state for myopic decision algorithm

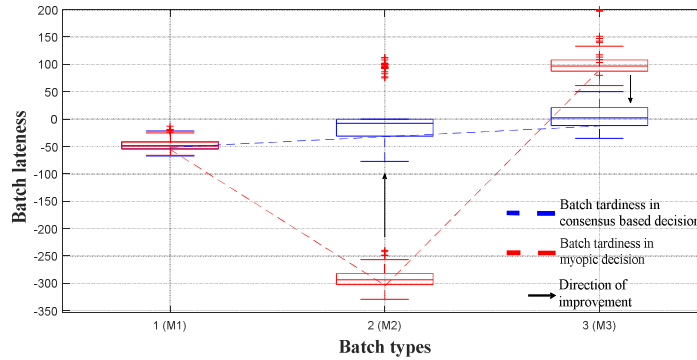


**Fig. 8** State disagreement ( $[x_i, x_j]$ ) level for the two decision algorithms

From these two calculated state values, it can be concluded that the disagreement level among product agents for the rush order offer in myopic decision is 95.21% higher than that of the consensus based decision, Fig. 8. Even though there is strong rush order priority by product agents of  $M_3$ , product agents of  $M_2$  did not show to satisfy this offer.

### 5.3.2 Production performance indicators

Lateness and makespan are taken as key production performance indicators. The simulation results considering the product lateness is presented in Fig. 9 where the negative values indicate product earliness and positive values for tardiness. In myopic decision algorithm, there is no clear behavioral pattern to show either the continuous tardiness or earliness of products in the execution process; the instability shown by the dotted red line in Fig. 9. As soon as the rush priority is sent by the leader agent of  $M_3$ , all the product agents of the proceeding manufacturing order have shown strong interest to be processed as early as possible ignoring the state of the rush order. For example,  $P_7$  of  $M_2$  has showed, on average, 290.65 seconds earliness in this decision not to satisfy the rush priority by  $P_{11}$  of  $M_3$ . And  $P_{11}$  in-turn becomes, on average, 99.89 seconds ( $\approx 5\eta_{ki}$ ) tardy from its priority intention.



**Fig. 9** Batch lateness for both decision algorithms

On the contrary, in consensus based decision, the pattern shows gradual and stable increment to satisfy the rush order; the dotted blue line in Fig. 9. Taking same example as the myopic decision, the leader agent of  $M_2$  has reduced its earliness to 69.42 seconds and the tardiness of that of  $M_3$  to 18.44 seconds ( $\cong 1\eta_{ki}$ ). This indicates that, as a result of implementing consensus based decision, tardiness of manufacturing orders has lowered by 81.53% than using the myopic decision.

The simulation result has also showed the effect of product's disagreement level on makespan. Even though the makespan in both decision algorithms has scored a value higher than the centrally planned schedule, the consensus based decision has comparatively improved the deliverability. As shown in Appendix 1, the mean completion date in consensus based decision is recorded as 1437.09 seconds or 12.55% lower than that of the myopic decision. This deviation has come from the myopic behavior of preceding product agents to response and set near optimal resequencing. Meanwhile, product agents in myopic decision algorithm might temporarily think that this behavior would lead them to quickly arrive to the next resource. But as the rush order is proposed by centrally agents, these product agents will be obligated to queue or buffer in front of the resource; higher buffer size. It is all this waiting time that led the deliverability of orders more delayed in myopic decision algorithm than in the consensus based decision algorithm.

### 5.3.3 Sensitivity analysis

Sensitivity analysis was also used as a performance indicator so as to evaluate the designed state. From statistical concept, sensitivity analysis is a technique used to determine how independent variable values can impact a particular dependent variable under a given set of assumptions. Fig. 10 shows this analysis considering makespan & state disagreement level as dependent variables and due date as explanatory variable. As it is shown in the figure, the values of makespan & state disagreement level are revealed in the z-axis. Three main colors namely dark blue (lower part of z-axis), light blue, and yellow (upper part of z-axis) are used to indicate the values. In the consensus based decision, the deliverability of the rushed manufacturing order has reduced (shown by the white downward arrow in Fig. 10(a)) by  $c_3$ . However, the deliverability of the preceding manufacturing order has increased (shown by the white horizontal arrow in Fig. 10(a)) by  $c_2$ . Even though the completion date of the leader product agent of  $M_2$  has increased, the makespan of the system has significantly reduced, Appendix 1. Such fulfilment to margin changes at minimized cost has come from the interest of all product agents to reduce their variability until their common goal is achieved. Consequently, the completion date of  $M_3$  has been scored to be lower than 650 seconds; the lower part of z-axis in Fig. 10(a) or option 2 in the figure. If the product agents of both manufacturing orders were deciding myopic, the completion date of  $M_3$  is expected to be higher than 650 seconds. No change from position 1 to position 2, in Fig. 10(a), as products of  $M_2$  did not show interest to increase their completion date from the initial plan. However, products of  $M_3$  are interested to complete their completion date as per the new plan which in turn leads to conflict among the products.

The state disagreement level at steady state is also shown in the z-axis of Fig. 10(b). When the margin of  $M_2$  and  $M_3$  are large enough (see the right upper part of Fig. 10(b) or the darkest blue one), there is no variability of interest between product agents and hence “Eq. 8” is satisfied. However, when the margin of either  $M_2$  or  $M_3$  becomes lower (see the lowest left part of Fig. 10(b)), product agents strive only to choose a destination with shortest completion date scheduled initially. At this time, they have to continuously compute their state and compare with their neighbor agents to minimize the disagreement level and hence to satisfy the rush order priority.

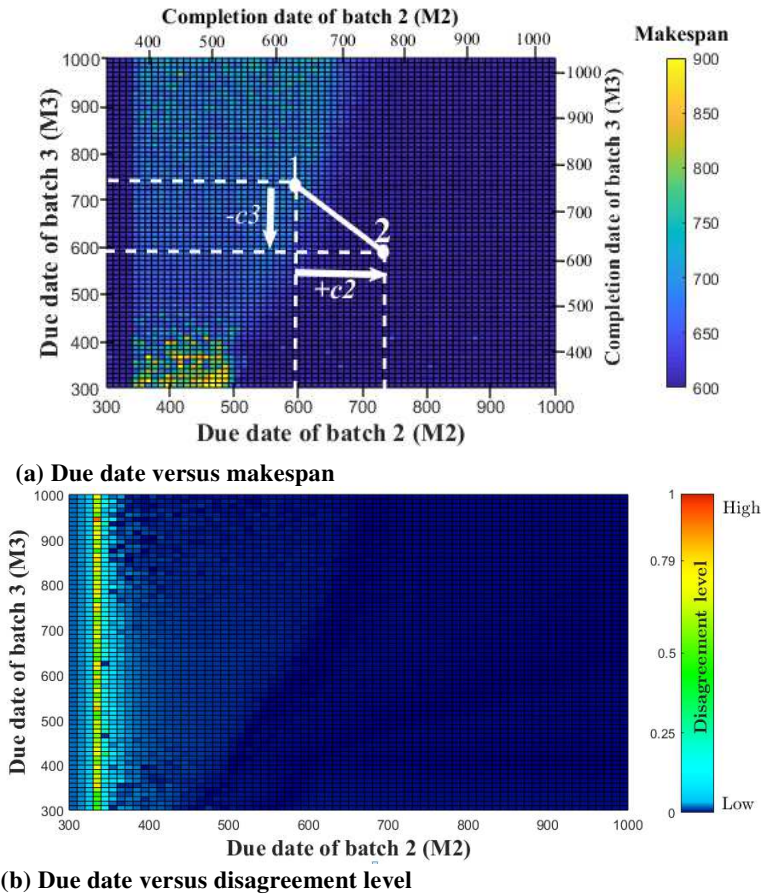


Fig. 10 Sensitivity analysis between performance indicators

## 6. Conclusion and Future works

This paper was aimed to investigate the application of Consensus algorithm for Multi-agent based Manufacturing system (CoMM). An industrial control problem was taken from wood industry and transposed to TRACILOGIS platform. This platform based simulation experiment has been conducted in order to validate the applicability. Convergence state, product lateness, and makespan are used as key performance indicators to compare results of this algorithm with results of a myopic decision algorithm. Accordingly, the consensus based decision approach has shown clear behavioral pattern (or stability) to satisfy an unexpected event called rush order. This has been achieved by making product agents to continuously cooperate using Leader-Follower communication approach to minimize non-value adding activities created after the rush order. For instance, the results obtained from convergence state presents that product agent's disagreement level for the rush order offer in myopic decision algorithm becomes extensively higher than that of the consensus based decision. Consequently, this disagreement level has made product agents of all manufacturing orders to exit from the routing sheet tardy.

We are aware that this paper has been concentrated in solving problems based on an academic experimental platform. The applicability of the experimental tests must be implemented in real manufacturing flow shop in order to see its effectiveness and to help detect some other problems (in particular, network time constraints). Meanwhile, additional experimental tests must be conducted considering (a) saturated routing sheet or complex communication constraints in order to see the flexibility of the algorithm and (b) large scale of product and resource entities in order to further investigate the adaptability of these entities in more complex environment. However, based on this encouraging result, many works are already engaged in our laboratory to define an optimal number of communicating agents, to define other states in regard of the expected partner company's objectives etc.

## 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

- [1] El Haouzi H, Pétin JF, Thomas A (2009) Design and validation of a product-driven control system based on a six sigma methodology and discrete event simulation. *Prod. Plan. Control.* 20(6):510–524
- [2] Rey GZ, Bonte T, Prabhu V, Trentesaux D (2014) Reducing myopic behavior in FMS control: A semi-heterarchical simulation–optimization approach. *Simul. Model. Pract. Theory.* 46:53–75
- [3] Trentesaux D (2009) Distributed control of production systems. *Eng. Appl. Artif. Intell.* 22(7):971–978



- [4] Jimenez JF, Bekrar A, Zambrano-Rey G, Trentesaux D, Leitão P (2017) Pollux: a dynamic hybrid control architecture for flexible job shop systems. *Int. J. Prod. Res.* 55(15):4229–4247
- [5] Ehteshami B, Petrakian RG, Shabe PM (1992) Trade-offs in cycle time management: hot lots. *IEEE Trans. Semicond. Manuf.* 5(2):101–106
- [6] Wang WP, Chen Z (2008) A neuro-fuzzy based forecasting approach for rush order control applications. *Expert Syst. Appl.* 35(1–2):223–234
- [7] Trzyna D, Kuyumcu A, Lödding H (2012) Throughput time characteristics of rush Orders and their impact on Standard Orders. *Procedia CIRP.* <https://doi.org/10.1016/j.procir.2012.07.054>
- [8] Leitão P (2009) Agent-based distributed manufacturing control: A state-of-the-art survey. *Eng. Appl. Artif. Intell.* 22(7):979–991
- [9] Isern D, Sánchez D, Moreno A (2011) Organizational structures supported by agent-oriented methodologies. *J. Syst. Softw.* 84(2):169–184
- [10] Xiong W, Fu D (2018) A new immune multi-agent system for the flexible job shop scheduling problem. *J. Intell. Manuf.* 29(4):857–873
- [11] Caridi M, Cavalieri S (2004) Multi-agent systems in production planning and control: an overview. *Prod. Plan. Control.* 15(2):106–118
- [12] Wooldridge M (2009) *An Introduction to MultiAgent Systems*. John Wiley & Sons, Liverpool
- [13] Rey GZ, Pach C, Aissani N, Bekrar A, Berger T, Trentesaux D (2013) The control of myopic behavior in semi-heterarchical production systems: A holonic framework. *Eng. Appl. Artif. Intell.* 26(2):800–817
- [14] Tonino H, Bos A, de Weerd M, Witteveen C (2002) Plan coordination by revision in collective agent based systems. *Artif. Intell.* 142(2):121–145
- [15] Nagarajan M, Sošić G (2008) Game-theoretic analysis of cooperation among supply chain agents: Review and extensions. *Eur. J. Oper. Res.* 187(3):719–745
- [16] Rosenschein JS, Zlotkin G (1994) *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. Massachusetts, USA
- [17] Kraus S (1997) Negotiation and cooperation in multi-agent environments. *Artificial Intelligence.* 94(1-2):79-97
- [18] Jennings NR, Faratin P, Lomuscio AR, Parsons S, Wooldridge MJ, Sierra C (2001) Automated negotiation: prospects, methods and challenges. *Group Decis. Negot.* 10(2):199–215
- [19] Rahwan I, Ramchurn SD, Jennings NR, Mccburney P, Parsons S, Sonenberg L (2003) Argumentation-based negotiation. *Knowl. Eng. Rev.* 18(4):343–375
- [20] Pannequin R, Thomas A (2012) Another interpretation of stigmergy for product-driven systems architecture. *J. Intell. Manuf.* 23(6):2587–2599
- [21] Valckenaers P, Van Brussel H (2016) *Design for the Unexpected: From Holonic Manufacturing Systems towards a Humane Mechatronics Society*. Oxford, UK
- [22] Leitão P, Barbosa J, Trentesaux D (2012) Bio-inspired multi-agent systems for reconfigurable manufacturing systems. *Eng. Appl. Artif. Intell.* 25(5):934–944

- [23] Pach C, Berger T, Bonte T, Trentesaux D (2014) ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. *Comput. Ind.* 65(4):706–720
- [24] Mezgebe TT, El Haouzi HB, Demesure D, Thomas A (2018) A Negotiation-based control approach for disturbed industrial context. *IFAC-Pap.* 51(11):1255–1260
- [25] Olfati-Saber R, Murray RM (2004) Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control.* 49(9):1520–1533
- [26] Cao Y, Yu W, Ren W, Chen G (2013) An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Trans. Ind. Inform.* 9(1):427–438
- [27] Dimarogonas DV, Kyriakopoulos KJ (2007) On the rendezvous problem for multiple nonholonomic agents. *IEEE Trans. Autom. Control.* 52(5):916–922
- [28] Sinha A, Ghose D (2006) Generalization of linear cyclic pursuit with application to rendezvous of multiple autonomous agents. *IEEE Trans. Autom. Control.* 51(11):1819–1824
- [29] Lee D, Spong MW (2007) Stable flocking of multiple inertial agents on balanced graphs. *IEEE Trans. Autom. Control.* 52(8):1469–1475
- [30] Olfati-Saber R (2006) Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control.* 51(3):401–420
- [31] Moore KL, Lucarelli D (2007) Decentralized adaptive scheduling using consensus variables. *Int. J. Robust Nonlinear Control:IFAC-Affiliated J.* 17(10-11):921–940
- [32] Ogren P, Fiorelli E, Leonard NE (2004) Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Trans. Autom. Control.* 49(8):1292–1302
- [33] Su H, Wang X, Lin Z (2009) Synchronization of coupled harmonic oscillators in a dynamic proximity network. *Automatica.* 45(10):2286–2291
- [34] Ren W, Beard RW (2008) *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications.* Verlag, London
- [35] Wang X, Shao J (2015) Consensus for Discrete-Time Multi-agent Systems. *Discrete Dyn. Nat. Soc.* <http://dx.doi.org/10.1155/2015/380184>
- [36] Cardin O, Trentesaux D, Thomas A, Castagna P, Berger T, El-Haouzi HB (2017) Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. *J. Intell. Manuf.* 28(7):1503–1517
- [37] Espejo R, Reyes A (2011) *Organizational Systems: Managing Complexity with the Viable System Model.* Heidelberg Dordrecht, London New York

**Appendix 1** Mean completion date of each manufacturing order (in seconds) for each decision algorithm

Centrally predictive schedule		Consensus based decision		Myopic decision	
Batch sequence	Completion date	Batch sequence	Completion date	Batch sequence	Completion date
$(M_1)$	174.75	$(M_1)$	174.73	$(M_1)$	174.79
$(M_2)$	409.25	$(M_3)$	409.25	$(M_2)$	409.44
$(M_3)$	554.32	$(M_2)$	572.76	$(M_3)$	654.95
$(M_4)$	661.32	$(M_4)$	683.32	$(M_4)$	781.37
$(M_5)$	823.07	$(M_5)$	850.45	$(M_5)$	972.49
$(M_6)$	918.32	$(M_6)$	948.87	$(M_6)$	1085.03
$(M_7)$	1013.57	$(M_7)$	1047.29	$(M_7)$	1197.57
$(M_8)$	1142.82	$(M_8)$	1180.84	$(M_8)$	1350.28
$(M_9)$	1204.82	$(M_9)$	1244.90	$(M_9)$	1423.54
$(M_{10})$	1266.82	$(M_{10})$	1308.96	$(M_{10})$	1496.80
$(M_{11})$	1328.82	$(M_{11})$	1373.02	$(M_{11})$	1570.05
$(M_{12})$	1390.82	$(M_{12})$	1437.09	$(M_{12})$	1643.31
<b>Makespan</b>	<b>1390.82</b>		<b>1437.09</b>		<b>1643.31</b>