A framework for multi-robot control in execution of a Swarm Production System

Avhad, Akshay; Schou, Casper; Madsen, Ole

Published in:
Computers in Industry

DOI (link to publication from Publisher):
10.1016/j.compind.2023.103981

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Computers in Industry 151 (2023) 103981

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

A framework for multi-robot control in execution of a Swarm Production System

Akshay Avhad *, Casper Schou, Ole Madsen

Department of Materials and Production, Aalborg University, Fibigerstræde 16, Aalborg East, 9220, Denmark

ARTICLE INFO

Keywords:
Reconfigurable Manufacturing
Swarm Production
Agile manufacturing
Autonomous Robots
Multi-agent framework

ABSTRACT

Swarm Production Systems adopt an agile, reconfigurable and flexible production philosophy using mobile robot platforms for workstations and material transport. As a result, the factory floor can continuously restructure itself to an optimal spatial topology suited to any given production mix. This new production paradigm has to deal with frequently changing factory layouts and an execution plan for a fleet of autonomous robots in the planning stage. For every reconfiguration in the event of a change of order, the carrier and process robots require an initial task plan prior to runtime production and a reactive mechanism to adapt to uncertainties on the shop floor. An interoperable management system across the production and robotics domain called the Swarm Manager handles the task planning, allocation and scheduling for process and product transport robots. This research provides conceptualization with an abstract framework and an architecture describing methods with required functionalities for a Swarm Manager. A generic framework based on multi-agent systems addresses the explicit functional scope for individual agents inside the Swarm Manager. Based on the functional needs, a system-level architecture is proposed to explain algorithms within task planning, allocation and scheduling agents, and information flow within them.

1. Introduction

Koren et al. (2018) explained the trend in manufacturing with a focus on mass production in the early 20th century, followed by mass customization expectations in the latter half, eventually leading to more personalization in production. Manufacturing has seen the evolution from dedicated manufacturing lines (DML) to Changeable paradigms like Flexible manufacturing systems (FMS) and Reconfigurable manufacturing systems (RMS) (Koren et al., 1999) that adopted agility, flexibility, scalability and reconfigurability in industrial production. Parunak (1996) explained the prospect of Multi-agent Systems(MAS) application in scheduling and controlling agile and reconfigurable manufacturing needs with a network of autonomous agents.

Holonic Manufacturing System (HMS) proposed in Valckenaers et al. (1997) harnessed hierarchical and heterarchical distributed multi-agent control for manufacturing execution with autonomous and collaborative entities called holons. Greschke et al. (2014) proposed a flexible production paradigm known as Matrix-Structured Manufacturing Systems (MMS) that offers a suitable solution for production with high variance needs with very flexible material flow routing and modular matrix-like factory layout. Another changeable paradigm within manufacturing known as Biological Manufacturing System (BMS) (Ueda, 1992; Ueda and Ohkura, 1994) promoted autonomous and adaptive manufacturing inspired by biological evolution. Self-Organizing Manufacturing Network (SOMN) (Qin and Lu, 2021) is a similar paradigm intended for mass personalization that thrives on principles of self-configuration, self-optimization and self-healing through induction of a network of autonomous entities (e.g., hardware & software tools, humans). Fluid Manufacturing System (FLMS) (Fries et al., 2021) on-demand allocation and reconfiguration of resources to production units through self-integration and self-parameterization. Line-less assembly systems (LMAS) (Hüttemann et al., 2019) is based on a production philosophy beyond assembly lines with mobile robots adapting to flexible stations on the shop floor. All the changeable paradigms mentioned above lead to non-linear factory layouts. However, that invokes complexities through shared resources like products, tools, mobile robots etc., and allocation to workstations. Though it enables cycle time independency for the workstations, the scheduling of the production system as a whole remains the function of customer demand or takt-time. Cyber–Physical Production Systems (CPPS) in Hsieh (2022b) incorporate such awareness by contextualizing the current state of runtime production with a perspective of future-generated states to reduce wait times and deadlocks during the run-time.

Most changeable and self-organizing production systems rely on mobile robots for flexibility in material flow and reconfigurability in

* Corresponding author.
E-mail address: akshayra@mp.aau.dk (A. Avhad).

https://doi.org/10.1016/j.compind.2023.103981
Received 20 February 2023; Received in revised form 19 May 2023; Accepted 20 June 2023
Available online 5 July 2023

Symbols and Abbreviations

C_m	Marginal Cost
E_t	Execution time for task
M	Makespan
m	Total Product Variants in an order
MS	Material Source
n	Total Product Instances in order
P	Predecessor node number
PG	Precedence Graph
PI	Product Instance
PV	Product Variant
SM	Swarm Manager
SR	Service Robot
STN	Simple Temporal Network
T_R	Carrier task for Transfer Robot
T_W	Process task for Workstation Robot
TM	Topology Manager
TR	Transfer Robot
TS	Tool Source
WR	Workstation Robot
AGV	Automated Guided Vehicles
AMR	Autonomous Mobile Robot
BMS	Biological Manufacturing Systems
CNET	Contract Net Protocol
CPPS	Cyber–Physical Production Systems
ERP	Enterprise Resource Planning
FLMS	Fluid Manufacturing Systems
FMS	Flexible Manufacturing System
HMS	Holonic Manufacturing Systems
LMAS	Line-less Mobile Assembly System
MARTHA	Mobile Autonomous Robots for Transportation and Handling
MAS	Multi-agent Systems
MES	Manufacturing Execution System
MMS	Matrix-Structured Manufacturing
MRTAS	Multi-robot Task Allocation and Scheduling
RMS	Reconfigurable Manufacturing Systems
SOMN	Self-Organizing Manufacturing Network
SPS	Swarm Production System
TePSSI	Temporal and Precedence-constrained Sequential Single-Item

production layout. Traditionally, fleet management systems for mobile robots in production are deployed post-commissioning of the process workstations or work cells on the production floor. They must adapt to the production scenario, and with limited interoperability between enterprise-level production technologies and the multi-robot control domain, the macro objectives of changeability and autonomous re-organization will be unachievable.

Swarm Production System (SPS) proposed in Schou et al. (2022) hypothesize extensive use of AMRs to enable the autonomous, self-organized, adaptive and scalable production system. SPS envisage a “dense” population of robots on a confined shop floor area exhibiting coherence and coordination throughout the production lifecycle; and not distinctive of demonstrating bio-inspired intelligence. AMRs are type transfer robots (*TRs*) and workstation robots (*WRs*). As a result, the shop floor layout, typically a topology, can be adapted to the current product demand during runtime. This gives the system a high degree of flexibility. However, it also poses several scheduling and control challenges to achieve production efficiency, as discussed in Schou et al. (2022) as future research challenges.

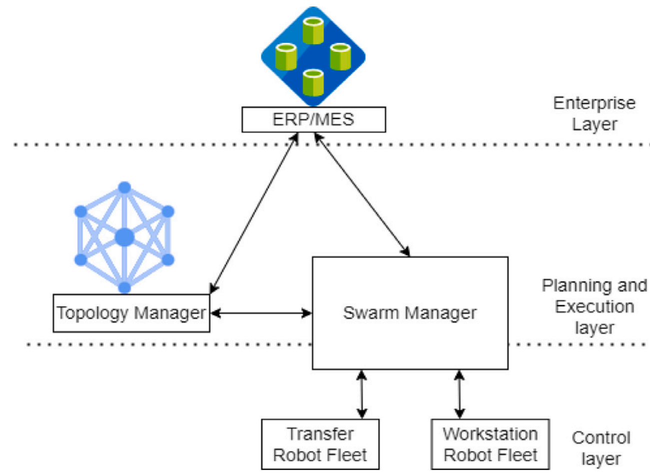


Fig. 1. High-level system architecture for SPS.

Fig. 1 shows the high-level system architecture for SPS, with the key actors involved and their interrelation. As shown in the figure, the architecture covers planning, scheduling and control levels, and also shows the integration with the common business infrastructure of Enterprise Resource Planning (ERP) and Manufacturing Execution System (MES). The topology manager (*TM*) determines a suitable shop floor layout of the workstations based on a subset of the order queue. The design and implementation of a *TM* have already been addressed in Avhad et al. (2023), where a *TM* that provides an optimized topology with spatial information and process sequences for a specific batch order of product mix, called batch topology, was proposed. Given a topology, the Swarm Manager (*SM*) orchestrates and coordinates the production of the orders. This includes scheduling individual product instances (PIs) and controlling the low-level order execution on the *TRs* and *WRs*.

The paper undertakes a study in Section 2 across production paradigms beyond traditional DML and control architectures implying the feasibility and robustness of such concepts in factories. The focus pivots on the conceptual model-based design of a *SM* based on the entities to control on SPS's shop floor. Priorly, SPS's generic scenario in Section 3 gives a fair understanding of process and material flow, paving the way to identify constraints and functionalities for the execution of such a system. The proposed *SM* in Section 4 adopts a multi-agent architecture, allowing the tasks of planning, scheduling and control to be distinct functional domains.

2. State of the art

We studied planning and control characteristics spanning changeable and self-organizing production paradigms and supporting architectures in the literature reviewed in Buckhorst et al. (2022b), Nielsen et al. (2023) and Kaiser et al. (2023) based on multiple aspects within production and manufacturing domain. A qualitative distinction illustrated through Table 1 encompasses factory planning and supervisory objectives of the entities and resources deployed on the shop floor. HMS and MAS enables production complex planning and control systems through faster design and deployment of multi-optimization and multi-constraints frameworks (Caridi and Cavalieri, 2004; Wu et al., 2023) Different control architectures in Van Brussel et al. (1998), Leitão and Restivo (2006) and Valckenaers (2020) adopt principles of HMS, enabling holarchy with hierarchical and heterarchical organization of holons empowering cooperative and self-organized manufacturing

Table 1
Qualitative comparison of manufacturing/production paradigms.

Paradigms	References for control architectures	Planning and Control features	Shopfloor	
			Autonomy	Maturity
HMS/MAS	PROSA (Van Brussel et al., 1998) ADACOR (Leitão and Restivo, 2006) ARTI (Valckenaers, 2020; Caridi and Cavalieri, 2004)	Holarchy (Hierarchy and Heterarchy) Co-operation of holons Logical self-organization	⊙	●
RMS/FMS	Koren and Shpitalni (2010) Da Silva et al. (2016) Florescu and Barabas (2020)	Modular, scalable control approach Dynamic configuration with flexible material flow Reconfigurable machines on shop floor	⊙	⦿
BMS/SOMN	Ueda et al. (2006) Ueda (2007) Qian et al. (2019)	Bio-inspired self-organization Centralized decision making Mobile robots, reconfigurable machines. Self configuration and self-optimization	◉	○
MMS	Trierweiler et al. (2020) Schmidtke et al. (2021) Nielsen et al. (2023) Filz et al. (2019)	Reconfiguration within workstation cell AGV based flexible material flow Dynamic product assignment. Takt-time independent production	⦿	●
FLMS	Fries et al. (2021) Hinrichsen et al. (2023)	Dynamic layout with discrete locations Flexible material flow and scalable	◉	⦿
LMAS	Buckhorst et al. (2022b) Göppert et al. (2023) Buckhorst et al. (2022a) Mathews et al. (2023)	Flexible layout with movable workstations AMR enabled material flow based on Precedence graph Holarchy in fleet and production management. Capable of matrix and non-linear production topologies.	●	⦿
SPS	Schou et al. (2022) Avhad et al. (2023) Rodriguez et al. (2021)	Heterogeneous AMR Fleet Management Dynamic topologies (line, matrix, swarm) Demands faster deployment and topologies and execution plan Scalable with high density factory	●	⦿

"Autonomy" refers to level of freedom paradigm exhibits with dynamic workstation placement and flexible material flow.
"Maturity" quantifies the level of granularity on control system design for execution of process and material flow.
High = ● , Medium to high = ◉ , Medium = ⦿, Low to Medium = ⊙, Low = ○.

goals. The factory-level MMS concept follows a static grid-based symmetric layout of complex work cells. Still, it can enable the capability with multi-variant production through restructuring within individual work cells in tandem with flexible material flow (Schmidtke et al., 2021; Trierweiler et al., 2020; Schönemann et al., 2015). AGVs primarily drive the material flow in MMS; fleet management for these AGVs would focus on the pick and delivery tasks while navigating through a pre-defined set of routes through the free spaces. On the planning level, LMAS and BMS propose self-organized, adaptable, evolving factory layouts that are a function of the product and production resources comprised of tools, process machines & robots (Hüttemann et al., 2019; Ueda, 1992). LMAS and SPS exhibit coherence towards the free-spaced material flow and flexibility in stationing process machines.

The control reference architectures linked with different paradigms in Table 1 provide high-level abstract viewpoints describing composition within systems, holons, agents etc. The MAS and HMS approaches are the basis for most production planning and control systems. SPS specifically addresses dynamic reconfiguration in Avhad et al. (2023) for the production layout performed in varying frequencies throughout runtime production, making it distinct from most relatable paradigms, i.e. BMS and LMAS. Consequently, an execution system capable of quick adaptive estimation of a plan for a heterogeneous fleet of workstations and transport AMRs is desired. As such, part of such an execution system has a functional overlap with traditional fleet managers used in deployments of multi-robot AMRs, typically for logistics. The control architectures, compared before, assist the design and implementation of production systems as a whole but lacks granularity when it comes to the design of interoperable fleet manager co-existing in production operations and multi-robot AMRs. Thus, we continue this section by reviewing related work on fleet managers in production, warehouses and manufacturing logistics.

2.1. Requirements for a fleet manager

Alami et al. (1998) conducted the earliest study on autonomous mobile robots (AMRs) with multi-robot cooperation to orchestrate a large fleet in the MARTHA (Mobile Autonomous Robots for Transportation and Handling Applications) project. This was demonstrated with a dual-layer control architecture with a higher-level task plan decision layer and a lower-level functional layer that handles localization, obstacle avoidance and motion control on the robot. Hyland and Mahmassani (2017) emphasized that vehicle fleet management is a dynamic multi-robot pickup and delivery problem with implicit or explicit time window constraints. An interpretation can be made for dynamic scaling of robot fleets up and down with promptness is crucial and needs to be investigated. Souto et al. (2021) presented fleet management in a shop-floor environment with a dynamic task scheduling allocation to establish uninterrupted task flow without human intervention. It is ensured by eliminating bottlenecks and traffic congestion on the navigating routes.

2.2. Automated task planning and allocation

Task planning is a high-level sequence of actions that allows the robot to estimate the required task to complete the mission. Galindo et al. (2008) describe task planning as a function of spatial information and domain knowledge. Spatial information is sufficient for navigation and localization actions, whereas domain knowledge enables autonomy through intelligence. Kattepur and Purushotaman (2020) proposed an automated task planning framework to simulate a single robot in a pick-and-deliver application of warehouses. The automated planner adapts to the environment with an explicit knowledge base associated. Task planning requires semantic knowledge about the corresponding environment of station types and navigation routes. Topological maps provide the required semantics when combined with spatial and hybrid information containing high-level reasoning for the robot to navigate in complex environments (Crespo et al., 2020; Wu et al., 2014). Multi-robot task allocation is a constrained NP-hard problem where the computation of an optimal solution is prone to the size of the AMR fleet and the task planned (Gao and Cai, 2006). De Ryck et al. (2020) describes a spectrum of control requirements and methods to enable the strategies in centralized and distributed execution scenarios for Automated Guided Vehicles (AGV). Task allocation is one of the complex challenges in AGV fleet control that can be handled centrally and distributed. The scope of task allocation is global, while path planning is a mix of global and local objectives for the robot fleet. Then there are entirely local-level objectives of battery, maintenance and fault management.

2.3. Distributed task allocation

Optimization-based methods in De Ryck et al. (2020) and Mosteo and Montano (2010) target optimal cost on global data from the total fleet size for task allocation mechanism. This approach restricts the allocation mechanism to a central entity that overlooks global data and can be computationally time-consuming to find an optimal solution. Market-based approaches adopting an auctioneer–bidder protocol to obtain a simple, viable solution could overcome the temporal constraint in the task allocation mechanism. This approach has a better prospect of incorporating a flat hierarchy than other solutions proposed in De Ryck et al. (2020) and Khamis et al. (2015) A basic CNET (Liang and Kang, 2016) protocol implementation allows the robot agent to bid on the requested task from the auctioneer. The bidding data consist of Marginal cost in De Ryck et al. (2020) to execute the auctioned task, a comprehensive scheduled list of all assigned task and preemptive downtime status.

2.4. Scheduling in robot-based production systems

The cycle time in the makespan calculation equates to the average process time of the workstations in the MMS production. Eliminating static cycle time and maintaining consistency in the runtime process is a key to line-balancing in variable process times scenarios. Schönemann et al. (2015) emphasize system utilization as an indicator of the economic performance of a manufacturing system and can be achieved with low waiting and idle times. Lead times and travel costs can be secondary objectives if the system utilization is high. Re-routable principle material flow often requires a combined scheduling strategy with minimal makespan objective for workstations and part-carrying AGV and metaheuristic algorithm provides a feasible solution (Deroussi et al., 2008; Jerald et al., 2006; Zheng et al., 2014). Nishi et al. (2011) discussed the goal of simultaneous scheduling and non-overlapping AGV routing by minimizing the cumulative tardiness of jobs assigned to the corresponding task. A hybrid time and cost model (Fazlollahtabar and Hassanli, 2018) for path planning suits modelling production scenarios with queuing and penalty for tardiness. Lacomme et al. (2013) proposed oriented disjunctive graph modelling, explicitly specifying tasks performed by workstations and robots in job-shop joint scheduling problems. A graph-based aggregated joint spatial–temporal network formed merging similar spatial–temporal networks structured on heterogeneous tasks in Hsieh (2022a) increased search space for an optimal solution in reinstating resilience and robustness in CPPS.

2.5. Simultaneous planning, allocation and scheduling

Faruq et al. (2018) proposed simultaneous task planning and allocation for a stochastic environment. They use models based on the Markov decision process for individual robots and linear temporal logic to generate a sequence of tasks and adaptive reallocations considering robot failures. Messing et al. (2022) demonstrated a unified interleaved approach towards simultaneous task planning and allocation to reduce frequent backtracking for optimal target agents. This agnostic planning approach is more task-focused rather than agent-specific and interleaves tasks for robots based on the individual's potential to execute them. Distributed computation in simultaneous task allocation and motion coordination in Kulatunga et al. (2007) provides a sub-linear speedup compared to the centralized approach. A framework for simultaneous task allocation and planning in Schillinger et al. (2018) automatically generates high-level actions based on semantics and allocates without a priori cost estimation for all tasks.

2.6. Summary

The market-based task allocation method enables scaling fleet in a centralized distributed control approach and setting up a prospect to switch to a decentralized control network in future. Execution of heterogeneous multi-robot fleets in a production environment additionally brings challenges with the on-time delivery commitments associated with products in order inventory. A joint process and material flow scheduling are central to achieving the desired lead times and takt-times. Ultimately, an integrated approach to planning, allocation and scheduling is desired. Complexity in automated task planning is proportional to the complexity of the production environment; a pre-defined scenario of SPS can accelerate the initial prototyping of an execution system like *SM*. Before an *SM* framework is proposed, we exemplify SPS with a generic scenario that describes the structural aspect.

3. Generic swarm production scenario

This section outlines an abstract scenario that exemplifies an SPS's composition. The purpose is to highlight the different planning, scheduling and control objectives that arise in SPS, which later in Section 4 are used to derive the objectives for the SM architecture. Consequently, the description of the generic scenario is kept as a high-level representation of the logic layout and product flow. Fig. 2 visualizes the layout and logic flow of the generic scenario. The production logic flow starts at Material Source (MS), following *WRs* precedence pre-determined from ERP/MES and ending at a *Sink* station to be dispatched. Fig. 2 exemplifies material and production flow for 3 *PVs*, all conveyed through TRs based in a service warehouse (idle and maintenance purposes). A *Tool Source* station, floating or docked, provides equipment for process tasks (gripper heads) and repair.

3.1. Production type

SPS is economically suitable for medium to high customization production processes adhering to Batch production, Job shop or Lot size one production. The production task of the generic scenario is to produce a product family with *n*-number of product variants (PVs). Orders for each *PV* vary in quantity from *1..m* and following Fig. 1, customer orders arrive in the ERP and are then released as production orders to a pre-planning database, from where the *TM* and *SM* draw the orders. However, a finite list of *PVs* is essential for the topology optimization and reconfiguration process inside the *TM* that is either forecasted or queried from a pre-planning database. The size of the production order data is a $m * n$ and is usually referred to as the SPS batch order.

3.2. Actors

The generic swarm production scenario includes several actors, which are described below.

Workstation Robot (WR): The *WRs* constitute the processing capability of the SPS. As such, the *WR* is a workstation that implements a given process that it can apply to a product or a material. In SPS, the *WRs* are mobile and can be relocated on the shop floor according to

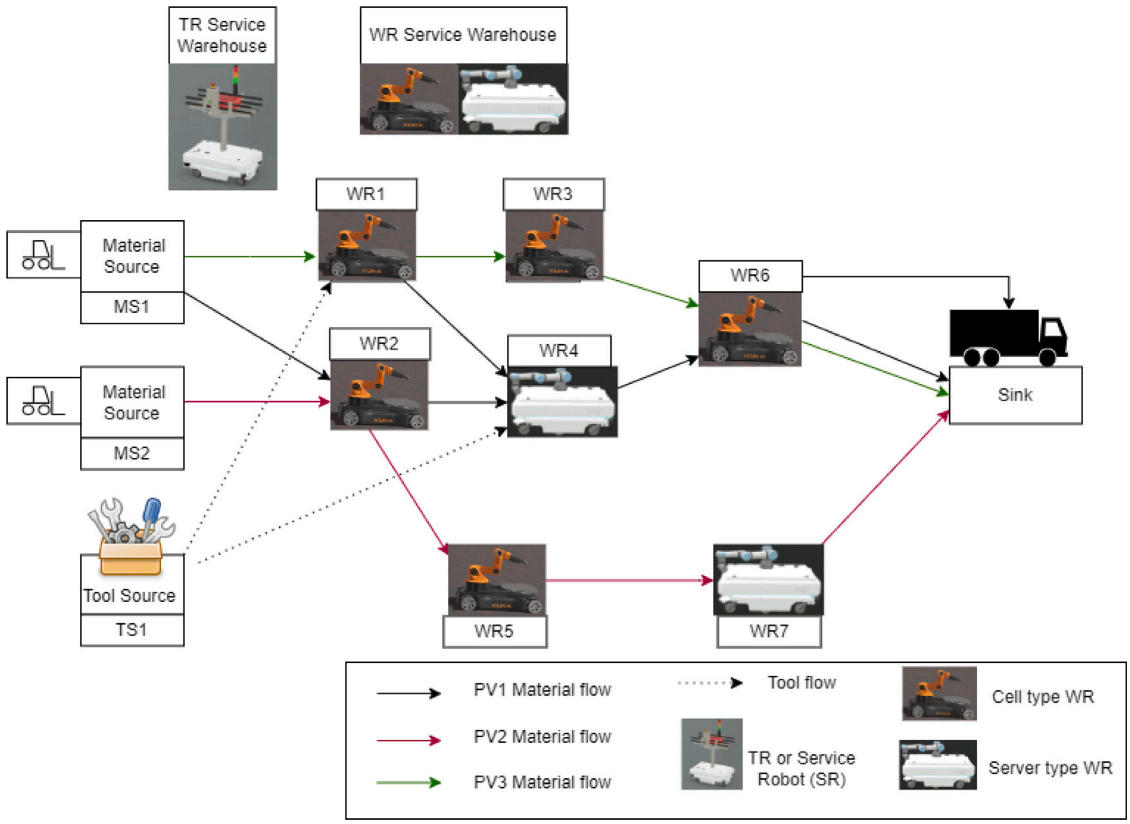


Fig. 2. Generic swarm production scenario used to highlight the different control objectives. The images in the figure are intended towards visual understanding and do not represent SPS's equipment selection.

the topology determined by the *TM*, either manually or autonomously. *WRs* can be either a *Server* or *Cell* type. Server *WRs* process the product or material directly on the *TR* while Cell type requires the product or material to be unloaded from the *TR* and placed in the *WR*. The choice between cell and server depends on the given process, the processing time and the overall process flow.

Source and sink: The Source entity acts as an entry point for materials and tools in the SPS and therefore has type Material Source (*MS*) and Tool Source (*TS*). *MS* represents the input of materials and products entry into the SPS where a product, a sub-part of the product or a material is loaded onto a *TR*. *TS* is a tools station that provides the necessary equipment for *WR* to process the unfinished products, also conveyed by *TR*. Sinks represent the output of the SPS, where a finished product (or sub-product) is unloaded from a TR to exit the scope of the current SPS. Sources and sinks are considered *WRs*, albeit often with the constraint of being spatially fixed in the environment.

Transfer Robot (TR): The *TRs* are autonomous units that convey products and materials between *WRs*. According to the type of the *WR* (cell or server), the *TR* may carry a product or material through several consecutive processing steps (*WRs*), or it may perform simple pick-up and deliver between *WRs*.

Service Robot (SR): As in most other production systems, an SPS can include tasks that extend beyond the core production task, e.g. the transportation of tools between workstations or error-resolving actions. These tasks are termed *service tasks* and are performed *SRs*, which are also mobile robotic units that travel autonomously through the production floor.

Service Area: An SPS should include a service area (*SA*), which serves as a "pitlane" for the *TRs* and *WRs*. The role of the *SA* depends on the specific production case and *WR* design, but it may include recharging *TRs* and *WRs*, refilling *WRs* with materials, reconfiguring TRs and WRs and resolving errors.

3.3. Production flow

The production flow in an SPS is not constrained by the equipment, but the specific *PV* will often dictate a specific process sequence. Although process redundancy amongst the *WRs* is possible, the specific *PV* will impose constraints in the physical product flow. The production flow for an order with multiple *PVs* can be represented as a precedence graph as illustrated in Fig. 2. Combining the graph of all *PVs* yields a complete representation of all necessary *TR* routes. As shown in Fig. 2, a *WR* can take several inputs, e.g. to accept several components for an assembly, and it can produce several outputs, e.g. a product and waste. The product flow indicated by directed edges initiates at MS nodes and follows precedence depending on the *PV* type it belongs to. Every edge represents a product transfer task, and the sequence of actions depends on the *WR* type the carrier *TR* caters to, i.e. *Cell* or *Server*.

3.4. Generic objectives of SPS

The SPS is a new production paradigm with different planning challenges that eventually lead to unique scheduling and execution problems. Therefore, the macro-level objectives of planning can be broken down into specific goals associated with various time horizons. A summary of objectives based on planning and control for AMRs in Fragapane et al. (2021) is illustrated in Fig. 3.

Strategic: Agility in production demands a quick response to market needs that require the production system to adapt to new product families with efficient throughput. SPS inherits similar principles to achieve the capability to produce a product family with *m PVs*. The strategic objectives enlisted in Fig. 3 are directed towards producing multiple product families with variants; Efficient temporal cost or makespan;

Table 2
Requirement specifications within the SPS production and mobile robotics domain.

Actors	Domain-specific relevant information	
	Production	Mobile Robots
ERP/MES integration	Order Input	Resource list
TM integration	Location data of process machines	Warehouse and Service location
Task planning	Product specific process task list	Global task list loading and unloading points
Task allocation	Task distribution for process machines	Dynamic task allocation on TRs & WRs
Scheduling	Start/Stop timings for process machines	Joint temporal graphs of TRs & WRs
Global positioning	Realtime positions of WRs & TRs, for navigation and obstacle avoidance over production and safety areas	
Traffic supervisor	Resolve conflict at the crossroads between humans and AMRs	
Wireless channel	Low-latency wireless for deterministic control of AMRs and machines	

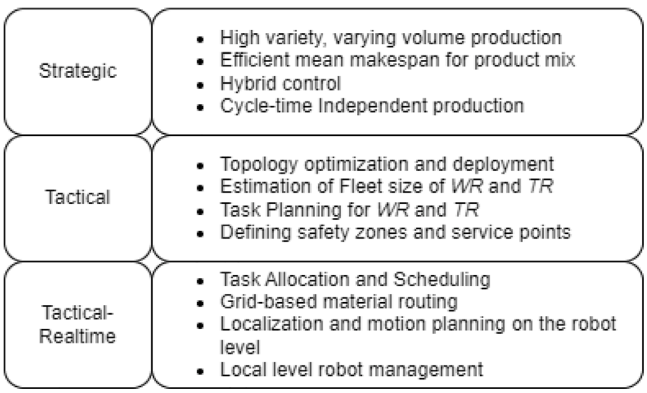


Fig. 3. Objectives in the generic SPS scenario categorized into different horizons based on Fragapane et al. (2021).

Varying Cycle time production, and Hybrid distributed and centrally orchestrated control of *WRs* and *TRs*.

Tactical: Tactical objectives initiate whenever an order is released from the ERP and MES, and the Topology Manager (TM) in Avhad et al. (2023) estimates an optimal layout suited to process requirements and material flow sequence. The reconfiguration starts with moving *WRs* to optimal positions from the estimated topology and stays until a new topology is dispatched from TM. Therefore, tactical timeline refers to the lifecycle of the completion of the current order makespan. Estimating the fleet size of *TRs* and a comprehensive task plan for the individual TR are prerequisites to production execution. Defining operator safety zones and service points for the *TR* fleet are also tactical goals executed in every topology reconfiguration.

Tactical-Realtime: An initial plan might be made prior to runtime, but an adaptable system like SPS should be able to handle online continuous planning. Therefore, task allocation and scheduling are adaptive processes frequently needed during production. Execution of low-level AMR control in both *TR* and *WR* fleets requires localization and motion planning on a real-time basis. Before, global route planning and scheduling for individual robots, a grid-based coordinate system is a prerequisite to collision-free material flow.

3.5. Requirement specification for an SM

Interoperability between the production and mobile robotics domains is the key to the comprehensive execution of SPS. Information relating to both domains is an essential draft required specifications in *SM*. A summary of required functional aspects within SPS is framed in Table 2 to scaffold a viable and robust *SM* development. The execution of *WRs* and *TRs* for process and carrier tasks is possible only when a list of *PI* for production and resources (list of *WRs*) associated with the high-level production planning systems ERP, MES. Dynamic factory layout reconfiguration in SPS is handled by *TM* with optimized *WRs* deployment locations on the shop floor. to enable the seamless product-specific material flow. The *TR* fleet requires offline locations for charging and service operations. *SM's* distinct core functionalities include task planning (estimation of process and carrier task based on the order from ERP/MES), allocation(distribution of tasks to WRs and TRs during runtime), and scheduling (reactive execution of *WRs* process task, and collective joint task graph of all *TRs* similar to the approach in Hsieh, 2022a). An integrated multi-agent framework exhibiting planning, allocation, and scheduling mechanism is required, supported by the summary in the Section 2.6. Additionally, a global positioning for AMR fleets for motion planning, collision avoidance for fleet orchestration, and a traffic supervision actor resolving potential conflict on dynamic crossroads within confined production areas. The wireless communication medium between centrally located *SM* and distributed AMRs must be deterministic with low latency for real-time execution.

4. Swarm Manager framework

A Swarm Manager (*SM*) is the execution engine in an *SPS*, which orchestrates a set of *WRs* and a set of *TRs*. Similar to an AMR fleet, *SM* deals with the problem of Multi-robot Task Allocation and Scheduling (MRTAS). A multi-agent concept to handle MRTAS with explicit task planning based on a dynamic topological factory layout and production scheduler is the backbone of developing a framework for SM. This section starts with a conceptual architecture explaining the high-level functions of *SM* and provides a scaffold for developing an exemplified architecture.

4.1. Objectives within SM Scope

The first objective in the *SM* is to estimate a plan for an order dispatched from the ERP/MES for which the topology in *TM* has been optimized. The quest for "*What*" is needed to be done to enable the product flow for all *PVs* in the associated production order is a goal in the task planning stage. The task planning is subsequently followed by assigning them to *TRs* and *WRs* that can perform the allotted task with minimal temporal cost, i.e. time to execute the task to minimize the production makespan. Task allocation addresses the question of "*Who*" does "*Which*" tasks in the associated production. The task assignment is an NP-hard problem as the optimization as an efficient makespan is a trade-off between the number of *TRs* for carrier fleet and stochastic delays due to congestion in a dynamic environment. The final stage in the *SM* addresses the question of "*When*" the allotted task to individual *TRs* and *WRs* should be executed to enable a production on the shop floor. A proactive mechanism shall assist in validating the feasibility of auctioned tasks with individual task schedules in every *TR*. The proactive approach shall be rewarding in the task allocation process to build a forecast of an approximate makespan. Contrary, execution of the tasks needs situation awareness at the state of execution schedule.

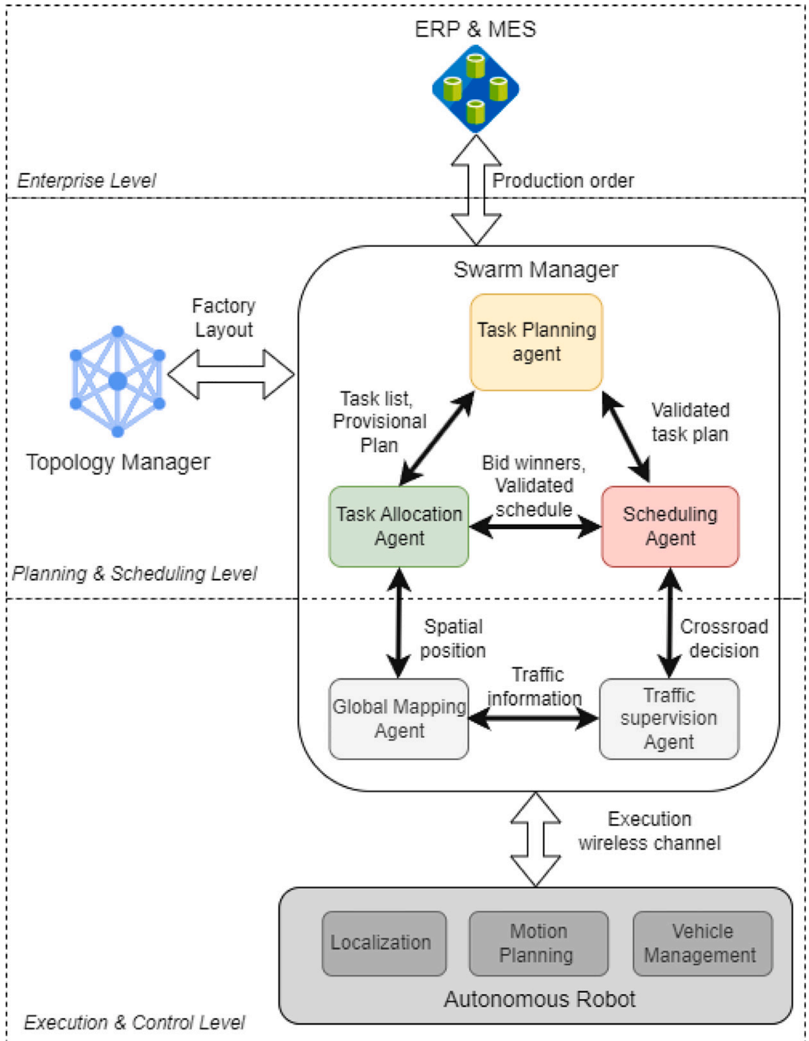


Fig. 4. Swarm Manager conceptual multi-agent framework describing functions for planning and execution phase in SPS.

4.2. Conceptual architecture

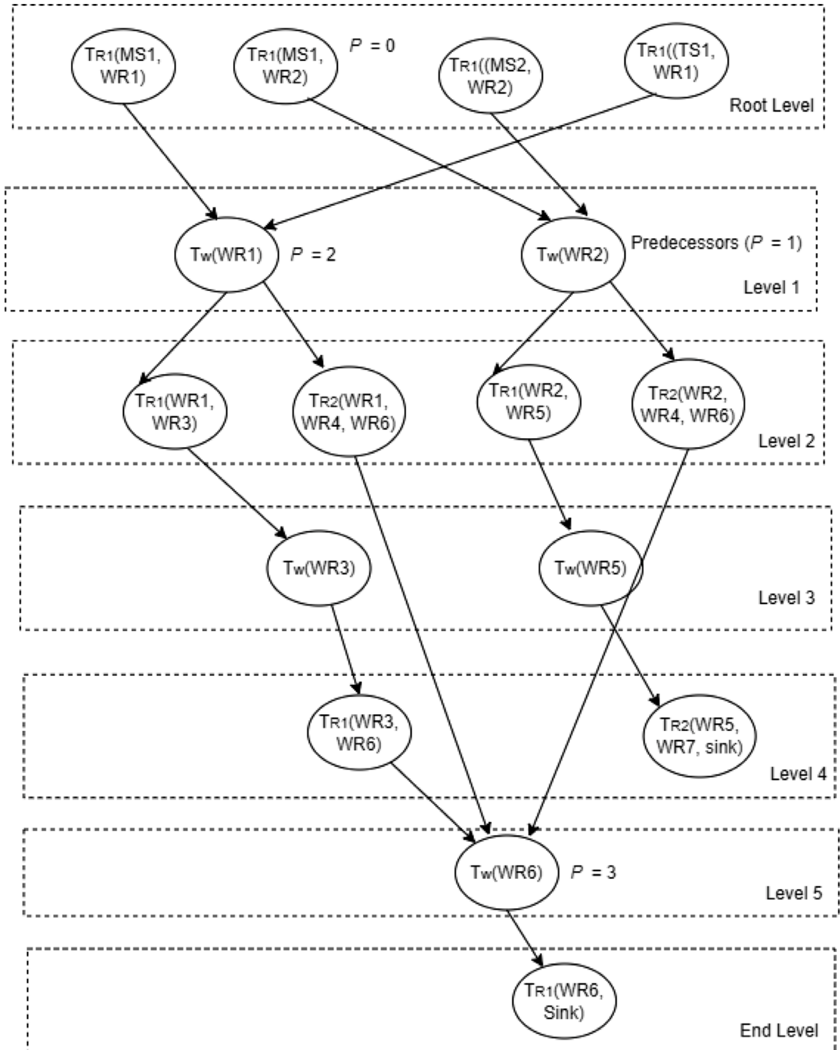
The section intends to provide a conceptual scaffolding based on the generalized objectives of an *SM*. A multi-agent framework with individual agents is explained inside the *SM*.

Task Planning agent: Task planning is an agnostic approach which does not consider target *TR* that eventually will execute the planned tasks. This process focuses mainly on deriving tasks based on the topology from *TM* and the Precedence Graph (*PG*) information associated with aggregated material for all *PVs* in order. Task planning is a high-level, event-driven process after introducing the new order and estimating an optimal topology in the *TM*. The topology provides information about the shop floor environment with *WRs*, their 2D positions and the inter-workstation shortest route.

Task Allocation agent: A market-based task allocation was deemed a desirable approach in the literature study in Section 2.3 based on the potential of a distributed control approach in the future. The agnostic task planning agent shares the elaborated task plan of loading and unloading tasks for material flow through *TRs*. The auctioneer broadcasts the carrier task with a travel cost to every *TRs* and expects multiple bids for every task auctioned. The bid with the lowest marginal cost is a prospective winner, but a final validation is subjected to approval from the scheduling agent.

Scheduling agent: As described in the literature review in Section 2.4, flexible material routing-based production systems enforce uncertain delays due to the stochastic nature of the production environment. The throughput is dependent on the average process times on the workstations. Therefore, tandem scheduling of *WRs* and *TRs* becomes essential to efficient makespan. The Joint Process and Carrier Scheduling agent aims to have a continuous production flow with a reactive scheduling policy focusing on uninterrupted and high utilization of process *WRs*. The validation of prospective winners from the Task Allocation agent needs proactive scheduling to foresee an impact on the makespan of complete orders.

Other agents: A Global Mapping agent actively communicates 2D floor maps and real-time spatial position data with *TR* and *WR* fleets. The Task Allocation agent depends on this spatial information to assign a cost value to an individual task. The SPS shop floor is a dynamic topological graph optimized in the *TM* with the *WRs* as nodes and edges forming a potential "highway" for *TR* navigation. Overlappings of these edges are persistent throughout the topology, causing potential conflict in the case of traffic management. To resolve the conflicts on crossroads, i.e. inter-*TRs* or between *TR* and human-driven transporters within the production area, the Traffic Supervision agent manages



No.	Task Type	Task Sequence
1	T_{R1}(Pickup node, Delivery node)	Dispatch to Pickup node --> Pickup *PI* -->Dispatch to Delivery node-->Drop *PI*
2	T_{R2}(Pickup node, Transit node, Delivery node)	Dispatch to Pickup node --> Pickup *PI* -->Dispatch to Transit node-->Wait at Transit node-->Dispatch to Delivery node and Drop PI
3	T_W(*WR* node)	Process task for *PI* on *WR* node

Fig. 5. Task-based precedence graph based on Fig. 2 while each node representing a task type for *TR* or *WR* and the edges directing the flow.

bottlenecks depending on the task priorities. The Traffic Supervision agent essentially performs Multi-robot Path Finding (MRPF), a well-established area of research. This research work does not intend to address the functions and objectives; it only refers to the name Traffic Supervisor. Global Mapping agent communicates continuously with Traffic Supervisor agent about the severity of crowding by incoming traffic and their real-time positions on crossroads.

Integrated Framework: The conceptual *SM* framework illustrated in Fig. 4 is an integrated task planning, allocation and scheduling agent concept to handle multi-robot task allocation based on the literature review in Section 2. The Global Mapping and Traffic Supervision agent interacts with these agents to collectively provide *TRs* and *WRs* localization data and motion planning information. The agnostic task planning agent estimates a set of tasks based on topology and order information. The task allocation receives the task plan and auctions to *TRs* and prepares a list of prospective winners based on the accepted bid. The joint Process and Carrier Scheduler agent validates the winning bids received from the Task Allocation agent. The final task plan with assigned *TRs* and *WRs* and a provisional schedule for each task is validated and compiled within an *SM*.

4.3. Exemplified Architecture for SM

We aim to build an exemplification based on conceptual *SM* architecture incorporating specific methods and algorithms to solve the tasks of MRTA within an SPS environment. The autonomous *TRs* can handle low-level motion planning and control, which generates information mainly on the local physical agent. These robot agents lack high-level

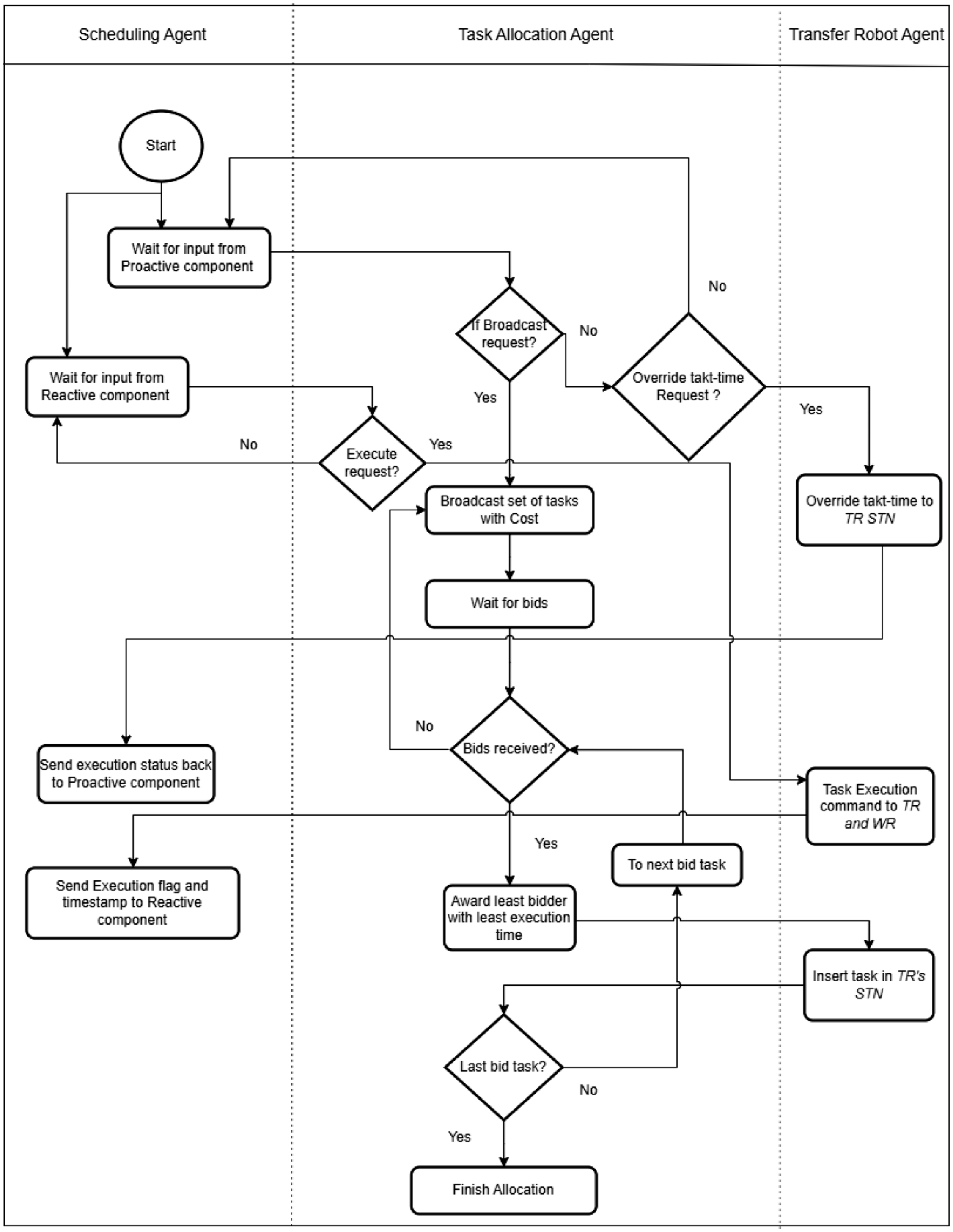


Fig. 6. Task allocation algorithm.

spatial knowledge of entities, i.e. *WRs* and other *TRs*. To plan a set of tasks for a complete production, the *TRs* must acquire global position and precedence information for the *WRs*, product *Source* and *Sink* nodes from the topology. The *TM* provides a topology representing the factory layout with a network of SPS *actors* for the task planning stage, a generalized scenario illustrated in Fig. 2. The *SM* adopts parallel single-item auctions to reduce complexity in computation as compared to combinatorial auctions based on a review in De Ryck et al. (2020) and Lagoudakis et al. (2004). The task allocation problem belongs to single-robot tasks, multi-task robots, and extended-time assignments. The basic prerequisite to this process demands each *TR* has a start position, a maximum velocity explicitly parameterized, and topological information of the *WRs* on the shop floor.

Precedence constraints: The tasks are subjected to constraints in market-based approaches. The NP-complete task allocation problem in the early development of SPS is precedence-constrained. The task for *TRs* follows precedence based on the material flow information embedded in the topology from the *TM*. A *PI* can only be processed on a *WR* if the TR has executed product transfer to the required workstation. Therefore, a *WR's* task for processing a *PI* for both *Server* and *Cell* type is always after a *TR's* task. In precedence where the material flow

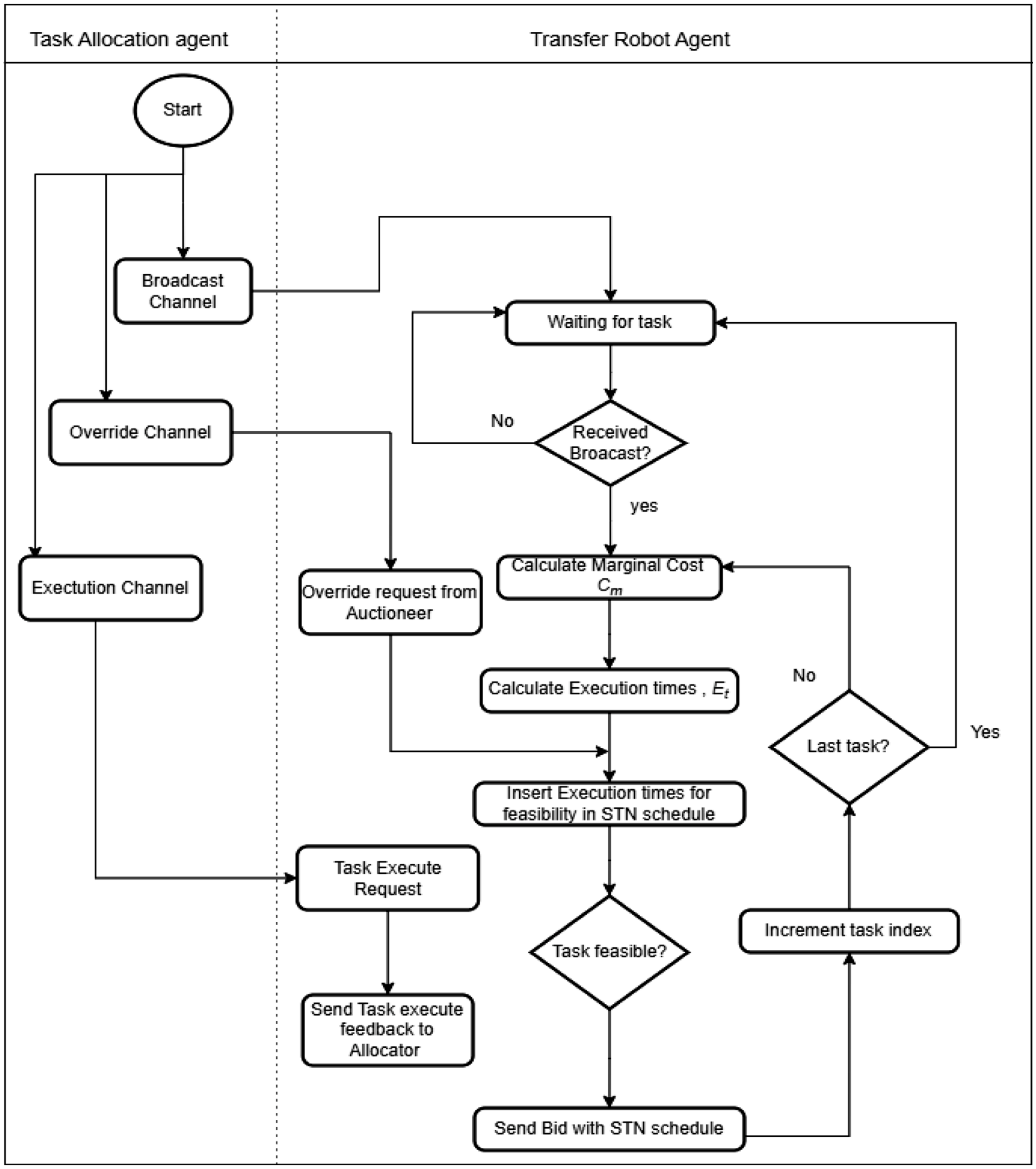


Fig. 7. Bidding process flowchart.

for a *PI* follows a linear sequence originating at a *Source* node and terminating at a *Sink* node, the scheduling of *TRs* and WRs is time-independent. It can be handled with a First-In-First-Out (*FIFO*) policy in an online scheduler.

A precedence graph for the task based on the topology from Fig. 2 is illustrated in Fig. 5. The nodes represent three distinct task types explained earlier, and the edges indicate the precedence for a *PI*. A task node can have multiple incoming edges indicating dependency on multiple predecessor tasks. The higher the number of predecessors (P) to any task nodes, the more critical the task node becomes (for $P > 1$) regarding uncertain queue and idle times. The criticality of this node is proportional to the length of the precedence chain and the degree of branches on the incoming side. A heuristic approach to contain the uncertain queues on the nodes requires a time window associated with this task to enable synchronicity.

Forward Iterative Task Allocation: The objective of a task allocation is minimizing the makespan M. The bid data model for *TRs* consists of the marginal cost to execute the task from its initial position on the topological map. The second is a Simple Temporal Network *STN* (Vidal and Bidot, 2001) data model, which represents a planned provisional schedule of all the tasks assigned to the *TRs*. A bidder *TR* is assumed to have sufficient physical resources to carry every job and battery capacity to execute any auctioned task. A solution to time-extended multi-robot task allocation is proposed in Bischoff et al. (2020) to assign and schedule a set of tasks in a precedent-constrained task environment. The solution improves on a constructive greedy heuristic with a local optimal search approach. The improved heuristics has substantial potential for problems with larger topological graphs. Nunes et al. (2017) developed TePSSI (Temporal- and Precedence-constrained Sequential Single-Item auction), a priority-based iterated auction scheme to handle precedence and temporal-constrained tasks. The core of the implementation is to identify critical tasks based on the shape of the precedence graph. The performance of the prioritized iterated auction over the simpler auction is substantial regarding overall makespan and

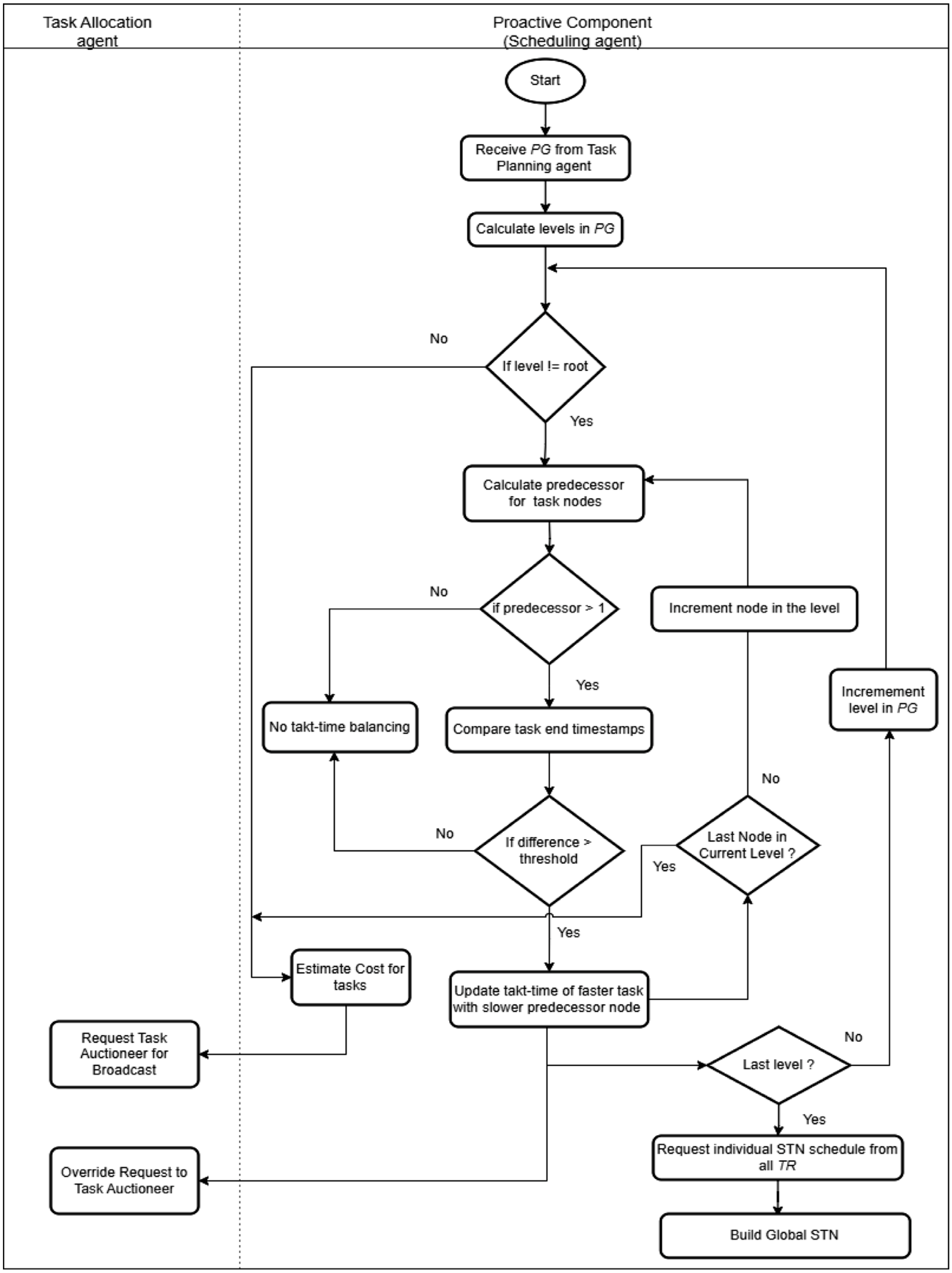


Fig. 8. Proactive scheduling component flowchart.

large problem sizes. The bidder robot maintains an STN to validate the feasibility of auctioned task in comparison to its own scheduled task. We augment this approach to suit task allocation in SM with an explicit scheduler that handles *WR* and *TR* tasks based on priority.

Fig. 5 represents a Directed Acyclic Graph (DAG) without a closed loop. Therefore, clustering nodes based on levels is a logical approach towards organizing a set of nodes based on the depth of precedences. The task allocation would follow forward chaining, starting at the Source and auctioning the set of clustered tasks based on the levels from the Source. At each task node on every level, the allocation would follow an iterative process to cross-validate with already assigned predecessor task feasibility with the current critical node ($P > 1$). The successive forward chaining and iteration must follow until the end level or task related to the sink is assigned successfully.

Proactive scheduling component: A proactive scheduling component is a requirement to optimize task allocation contributing to a pre-allocation phase based on the continuity of *TR* tasks and the feasibility

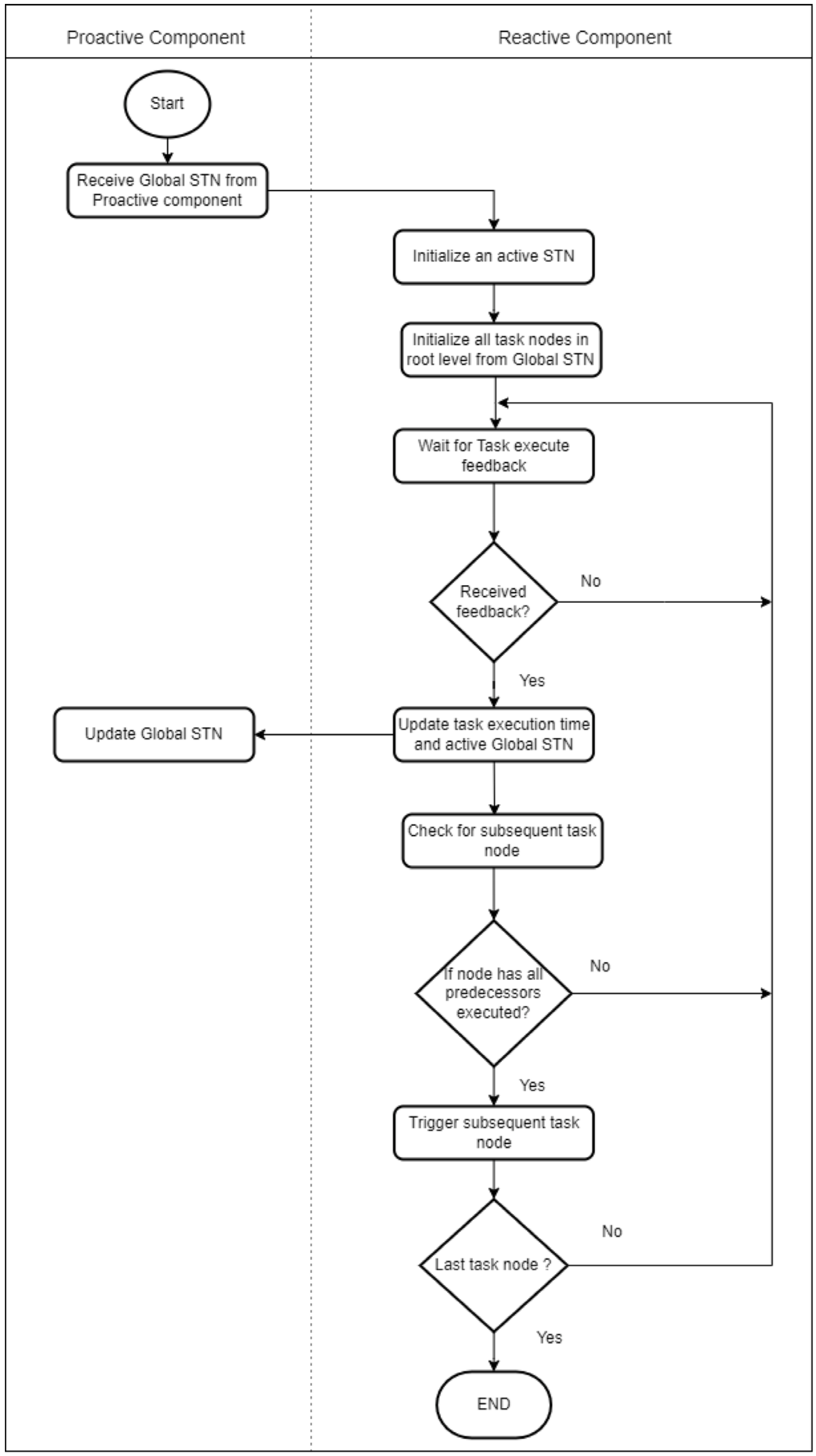


Fig. 9. Reactive scheduling flowchart for task execution.

of the predictive schedule on the prospective *TR* bidding agent. The pre-allocation process starts with a release of tasks on each level on a *PG*. The flowchart in Fig. 8 initiates with generating a *PG* based on a batch topology. The task allocation master algorithm in Fig. 6 auctions the first set of tasks from the root level without iterating over the predecessor, as this is the initial level. Every auction happens over a broadcast channel exclusive to un-auctioned and unscheduled tasks in any of the *TR* bidding agents. The task auction request triggers a task's implicit travel cost function depending on the spatial positions of *WR*, source and sink entities. The bidder agent algorithm in Fig. 7 illustrates the responsive process after a broadcast is performed. A bidder can be assigned multiple tasks and therefore carries its *STN* to validate the feasibility of the bid over previously assigned task schedules. The bid agent process computes a marginal cost of C_m based on its position from the auctioned task. It estimates an execution time E_t based on marginal cost and maximum possible stochastic delay due to the potential crowding in the planned motion path. The auctioned task is assigned to the bidder with the least execution time, and the

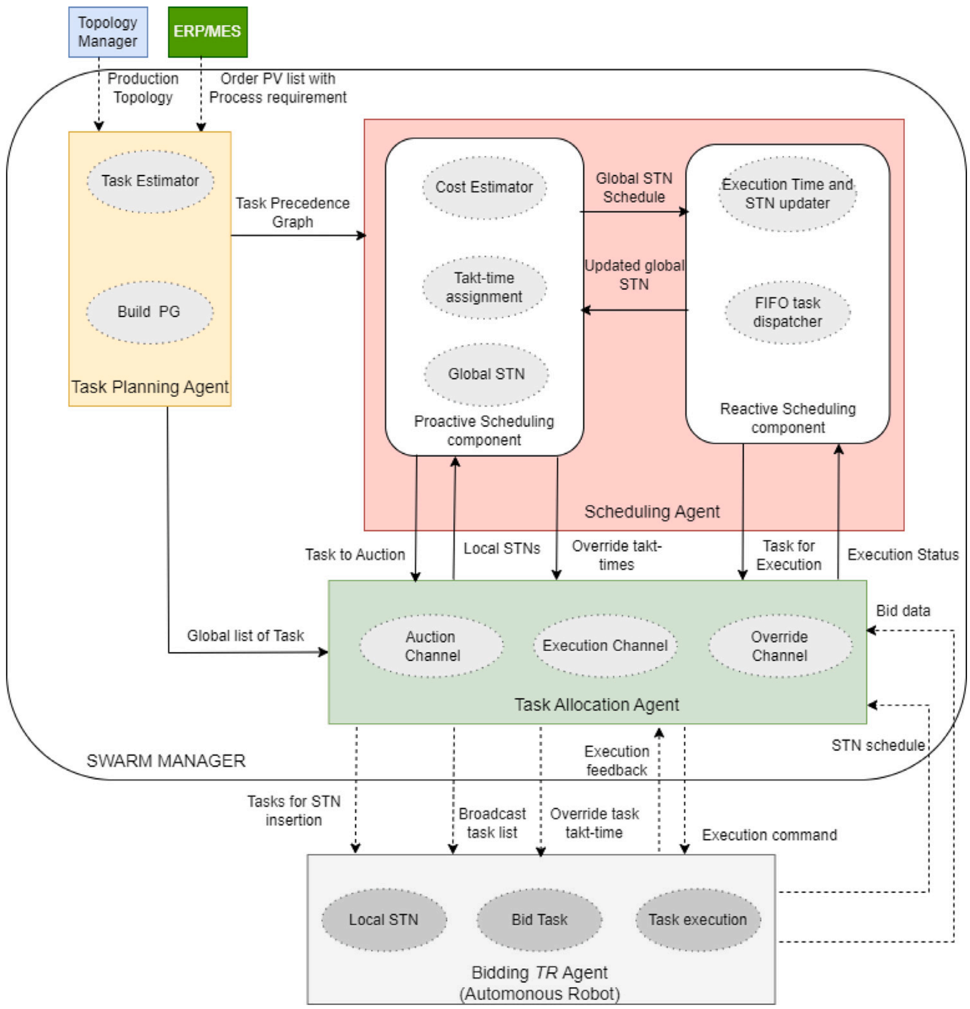


Fig. 10. SM component-level exemplified architecture based on market-based solution featuring interaction with TR.

rewarded task is inserted in the *STN* schedule of the successful bidder. The iterative process happens over successive levels at critical nodes requiring a flow-balancing strategy. The preemptive agent requests the task allocator to revisit the predecessor task nodes. A request is made to the task allocation master to override the assigned task on the bidder agent for a change in execution time. The process times for the slower predecessor are used as a uniform cycle time for all other predecessors. The bidder accepts the request on an override channel, where it simply overwrites the new execution times for the task. After an iteration over every possible level and task node in *PG*, a global *STN* is estimated based on the individual *STN* from the assigned *TRs* in the production.

Reactive scheduling component: Using a purely proactive approach in the execution of tasks would eventually lead to a breakdown in the production flow, considering the uncertainty of possible congestion and other system-level uncertainties in SPS. The reactive component in the scheduler requests the global *STN* from the proactive component, thus informing about all the tasks and their target *TRs* and a prospective schedule. The execution of a task in the flowchart illustrated in Fig. 9 initiates in the reactive component of a scheduler. Multiple FIFO queues are instantiated depending on the parallel execution requirement at the root level of the task *PG* as shown in Fig. 5. Every FIFO queue is enqueued with a next-in-line task from the PG, releasing the output of every FIFO queue for task execution. Whenever a task node undergoes a divergence onto the next level forming a new branch of execution, a new FIFO queue is created to enable parallel execution. Similarly, convergence at the output of a task node demands the FIFO queue termination of the branch.

Summary of exemplified architecture: In summary, an architecture for the *SM* is proposed based on augmented TePSSI and forward iterative chaining in Fig. 10. The multi-agent architecture includes an agnostic Task Planner, Iterative Task Allocator, and a Combined Scheduler (Proactive and Reactive) component. The information exchange within the SM agents is also shown in Fig. 10. Task Planning agent interfaced with a *TM* and ERP/MES builds a global list of tasks based on deployed production floor topology and current order. A *PG* generation is a sub-objective and task estimation to be further transferred to the Proactive component within the Scheduling agent. The Proactive component estimates the travel cost for the execution of individual tasks in the *PG*. The tasks are sent for auctioning through the Task Allocation agent, and the successfully assigned tasks spanning *TR* fleet form a global STN. A Takt-time assignment for *TR* tasks is central for the Proactive component based on the augmented TePSSI discussed earlier. The FIFO in the Reactive component dispatches the task for execution for *TRs* based on the scheduled global *STN*. The execution time feedback from *TRs* is updated in the global STN; a subsequent feasibility check is essential to identify conflicts in the updated global *STN* schedule. Conflict resolution may require re-allocating the task to *TRs* that happens over an Override channel. Similarly, takt-times must be updated to align with the updated global *STN*. The Task Allocation agent assigns these takt-times through the Override channel on *TRs*. The Execution channel releases an execution flag based on the Reactive component's FIFO for the task to be performed on the required *TR*.

5. Discussion

The proposed *SM* architecture is the first attempt towards designing a comprehensive execution system for an SPS known as the Swarm Manager. The conceptual framework targets a generic production scenario independent of a specific industry type and provides a minimalist design for a functional *SM*. Several internal agents to *SM* are defined based on explicit functionalities of planning, allocating and executing tasks. Planning depends on semantics like physical objects, spatial position, pathways and precedence graphs. In contrast, allocating the planned tasks requires a proactive approach to optimize the process based on a minimal makespan. Execution of tasks in a noisy environment with high uncertainty with temporal loss in SPS demands an adaptive scheduling technique. The agent-based approach reduces computational complexity by distributing optimization on multiple levels with a market-based approach. The proposed SM incorporates a market-based approach by distributing optimization on individual *TRs*; the allocation mechanism remains centralized due to dependency on global data. The global data on the batch scope primarily resides within *TM* and *SM* subsystems; therefore, a decentralized architecture would challenge the current *SM* framework.

Task planning based on orders from ERP/MES can be cumbersome for complex environments with structural constraints, confined dedicated spaces, and a limitation for the current *SM* architecture. Therefore, automated task planning based on Ontology or semantic-based mapping can be considered. Market-based task assignment in *SM* potentially be improved with the following modifications:

1. Combinatorial auction: To improve optimization in allocation with near-optimal solutions.
2. Redundancy in allocation agent: To Reduce single-point failure.
3. Consensus in auctioning: To increase performances through peer exchanges of local state.
4. Field-based behavioural solution: To improve collision avoidance strategies.

The augmented TePSSI could encounter conflicts during global *STN* graph data structure in orders with an extensive list of *PIs*. This remains to be a subject of simulation in such scenarios. Alternatively, near-optimal meta-heuristics or even reinforcement learning methods could potentially improve the scheduling performance of an SPS.

Like SPS, LMAS relies on AMRs for dynamic factory configuration and product conveyance, and an *SM* framework can assist in compiling a system-level view for holistic development. Product flow flexibility in MMS and FLMS could suffice with AGVs with redundant tracks on the production floor for navigation, offering significantly less stochastic downtime than AMR-based production. An execution system like *SM* can effectively manage tasks, resources and failures during the production course.

6. Conclusion

The paper started with conceptualizing a *SM* architecture that modern production systems with heterogeneous multi-robot fleets in the Industry 4.0 era demand. The proposed *SM* architecture will be the basis for building a demonstrator to test the feasibility of the proposed architecture. An SPS simulation platform with AMR fleet orchestration would facilitate bench-marking different planning, allocation and scheduling algorithms for optimization of makespan, system utilization and cycle time.

CRediT authorship contribution statement

Akshay Avhad: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Visualization. **Casper Schou:** Conceptualization, Writing – original draft, Supervision– original & review. **Ole Madsen:** Conceptualization, Writing – original draft, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

This work has received funding from Innovation Fund Denmark within the Manufacturing Academy of Denmark (MADE FAST) programme.

References

Alami, R., Fleury, S., Herrb, M., Ingrand, F., Robert, F., 1998. Multi-robot cooperation in the MARTHA project. IEEE Robot. Autom. Mag. 5 (1), 36–47.
Avhad, A., Schou, C., Madsen, O., 2023. Topology planning in swarm production system: Framework and optimization. In: Advances in Automotive Production Technology – Towards Software-Defined Manufacturing and Resilient Supply Chains. Springer, Cham.
Bischoff, E., Meyer, F., Inga, J., Hohmann, S., 2020. Multi-robot task allocation and scheduling considering cooperative tasks and precedence constraints. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics. SMC, pp. 3949–3956.
Buckhorst, A., do Canto, M., Schmitt, R., 2022a. The line-less mobile assembly system simultaneous scheduling and location problem. Procedia CIRP 106, 203–208, 9th CIRP Conference on Assembly Technology and Systems.
Buckhorst, A.F., Grahn, L., Schmitt, R.H., 2022b. Decentralized holonic control system model for line-less mobile assembly systems. Robot. Comput.-Integr. Manuf. 75, 102301.
Caridi, M., Cavalieri, S., 2004. Multi-agent systems in production planning and control: an overview. Prod. Plan. Control 15 (2), 106–118.
Crespo, J., Castillo, J.C., Mozos, O.M., Barber, R., 2020. Semantic information for robot navigation: A survey. Appl. Sci. 10 (2).
Da Silva, R.M., Junqueira, F., Filho, D.J.S., Miyagi, P.E., 2016. Control architecture and design method of reconfigurable manufacturing systems. Control Eng. Pract. 49, 87–100.
De Ryck, M., Versteyhe, M., Debrouwere, F., 2020. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. J. Manuf. Syst. 54, 152–173.
Deroussi, L., Gourgand, M., Tchernev, N., 2008. A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles. Int. J. Prod. Res. 46 (8), 2143–2164.
Faruq, F., Parker, D., Laccrda, B., Hawes, N., 2018. Simultaneous task allocation and planning under uncertainty. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, pp. 3559–3564.
Fazlollahtabar, H., Hassanli, S., 2018. Hybrid cost and time path planning for multiple autonomous guided vehicles. Appl. Intell. 48 (2), 482–498.
Filz, M.-A., Gerberding, J., Herrmann, C., Thiede, S., 2019. Analyzing different material supply strategies in matrix-structured manufacturing systems. Procedia CIRP 81, 1004–1009, 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.
Florescu, A., Barabas, S.A., 2020. Modeling and simulation of a flexible manufacturing system—A basic component of industry 4.0. Appl. Sci. 10 (22).
Fragapane, G., de Koster, R., Sgarbossa, F., Strandhagen, J.O., 2021. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. European J. Oper. Res. 294 (2), 405–426.
Fries, C., Fechter, M., Ranke, D., Trierweiler, M., Assadi, A.A., Foith-Förster, P., Wiendahl, H.-H., Bauernhansl, T., 2021. Fluid manufacturing systems (FLMS). In: Weißgraeber, P., Heieck, F., Ackermann, C. (Eds.), Advances in Automotive Production Technology – Theory and Application. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 37–44.
Galindo, C., Fernández-Madrigal, J.A., González, J., Saffiotti, A., 2008. Robot task planning using semantic maps. Robot. Auton. Syst. 56 (11), 955–966.
Gao, P.A., Cai, Z.X., 2006. Multi-robot task allocation for exploration. J. Cent. South Univ. Technol. 13, 548–551.
Göppert, A., Grahn, L., Rachner, J., Grunert, D., Hort, S., Schmitt, R.H., 2023. Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems. J. Intell. Manuf. 34 (5), 2133–2152.
Greschke, P., Schönemann, M., Thiede, S., Herrmann, C., 2014. Matrix structures for high volumes and flexibility in production systems. Procedia CIRP 17, 160–165.
Hinrichsen, T.-F., Fries, C., Hagg, M., Fechter, M., 2023. Order management perspective on fluid manufacturing systems. Procedia Comput. Sci. 217, 413–422, 4th International Conference on Industry 4.0 and Smart Manufacturing.

Hsieh, F.-S., 2022a. An efficient method to assess resilience and robustness properties of a class of cyber-physical production systems. Symmetry 14 (11).
Hsieh, F.-S., 2022b. A theoretical foundation for context-aware cyber-physical production systems. Appl. Sci. 12 (10).
Hüttemann, G., Buckhorst, A.F., Schmitt, R.H., 2019. Modelling and assessing line-less mobile assembly systems. Procedia CIRP 81, 724–729, 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.
Hyland, M.F., Mahmassani, H.S., 2017. Taxonomy of shared autonomous vehicle fleet management problems to inform future transportation mobility. Transp. Res. Rec. 2653 (1), 26–34.
Jerald, J., Asokan, P., Saravanan, R., Rani, A.D.C., 2006. Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm. Int. J. Adv. Manuf. Technol. 29 (5), 584–589, 2006 29:5.
Kaiser, J., McFarlane, D., Hawkridge, G., André, P., Leitão, P., 2023. A review of reference architectures for digital manufacturing: Classification, applicability and open issues. Comput. Ind. 149, 103923.
Kattepur, A., Purushotaman, B., 2020. RoboPlanner: a pragmatic task planning framework for autonomous robots. Cogn. Comput. Syst. 2 (1), 12–22.
Khamis, A., Hussein, A., Elmogy, A., 2015. Multi-robot task allocation: A review of the state-of-the-art. In: Koubâa, A., Martínez-de Dios, J. (Eds.), Cooperative Robots and Sensor Networks 2015. pp. 31–51.
Koren, Y., Gu, X., Guo, W., 2018. Reconfigurable manufacturing systems: Principles, design, and future trends. Front. Mech. Eng. 13 (2), 121–136.
Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Van Brussel, H., 1999. Reconfigurable manufacturing systems. CIRP Ann. 48 (2), 527–540.
Koren, Y., Shpitalni, M., 2010. Design of reconfigurable manufacturing systems. J. Manuf. Syst. 29 (4), 130–141.
Kulatunga, A.K., Skinner, B.T., Liu, D.K., Nguyen, H.T., 2007. Distributed simultaneous task allocation and motion coordination of autonomous vehicles using a parallel computing cluster. In: Tarn, T.-J., Chen, S.-B., Zhou, C. (Eds.), Robotic Welding, Intelligence and Automation. pp. 409–420.
Lacomme, P., Larabi, M., Tchernev, N., 2013. Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles. Int. J. Prod. Econ. 143 (1), 24–34.
Lagoudakis, M., Berhault, M., Koenig, S., Keskinocak, P., Kleywegt, A., 2004. Simple auctions with performance guarantees for multi-robot task allocation. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), Vol. 1. pp. 698–705, vol.1.
Leitão, P., Restivo, F., 2006. ADACOR: A holonic architecture for agile and adaptive manufacturing control. Comput. Ind. 57 (2), 121–130.
Liang, H., Kang, F., 2016. A novel task optimal allocation approach based on contract net protocol for agent-oriented UUV swarm system modeling. Optik 127 (8), 3928–3933.
Mathews, J.B., Rachner, J., Kaven, L., Grunert, D., Göppert, A., Schmitt, R.H., 2023. Industrial applications of a modular software architecture for line-less assembly systems based on interoperable digital twins. Front. Mech. Eng. 9, 12.
Messing, A., Neville, G., Chernova, S., Hutchinson, S., Ravichandar, H., 2022. GRSTAPS: Graphically recursive simultaneous task allocation, planning, and scheduling. Int. J. Robot. Res. 41 (2), 232–256.
Mosteo, A.R., Montano, L., 2010. A Survey of Multi-Robot Task Allocation. Tech. Rep., Instituto de Investigacin En Ingenierła de Aragn (I3A).
Nielsen, C.P., Avhad, A., Schou, C., Ribeiro da Silva, E., 2023. Control system architecture for matrix-structured manufacturing systems. Comput. Ind. 146, 103851.
Nishi, T., Hiranaka, Y., Grossmann, I.E., 2011. A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles. Comput. Oper. Res. 38 (5), 876–888.
Nunes, E., McIntire, M., Gini, M., 2017. Decentralized multi-robot allocation of tasks with temporal and precedence constraints. Adv. Robot. 31 (22), 1193–1207.
Parunak, H.V.D., 1996. Applications of distributed artificial intelligence in industry. Found. Distrib. Artif. Intell. 2 (1), 18.
Qian, C., Zhang, Y., Ren, S., 2019. Evolution of a self-organizing manufacturing network with homophily and heterophily. Procedia CIRP 83, 800–804, 11th CIRP Conference on Industrial Product-Service Systems.
Qin, Z., Lu, Y., 2021. Self-organizing manufacturing network: A paradigm towards smart manufacturing in mass personalization. J. Manuf. Syst. 60, 35–47.
Rodriguez, I., Mogensen, R.S., Schjørring, A., Razzaghpour, M., Maldonado, R., Berardinelli, G., Adeogun, R., Christensen, P.H., Mogensen, P., Madsen, O., Møller, C., Pocovi, G., Kolding, T., Rosa, C., Jørgensen, B., Barbera, S., 2021. 5G swarm production: Advanced industrial manufacturing concepts enabled by wireless automation. IEEE Commun. Mag. 59 (1), 48–54.
Schillinger, P., Bürger, M., Dimarogonas, D.V., 2018. Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems. Int. J. Robot. Res. 37 (7), 818–838.
Schmidtke, N., Rettmann, A., Behrendt, F., 2021. Matrix production systems-requirements and influences on logistics planning for decentralized production structures. In: 54th Hawaii International Conference on System Sciences 2021. Proceedings.
Schönemann, M., Herrmann, C., Greschke, P., Thiede, S., 2015. Simulation of matrix-structured manufacturing systems. J. Manuf. Syst. 37, 104–112.
Schou, C., Avhad, A., Bø gh, S., Madsen, O., 2022. Towards the swarm production paradigm. In: Andersen, A.-L., Andersen, R., Brunoe, T.D., Larsen, M.S.S., Nielsen, K., Napoleone, A., Kjeldgaard, S. (Eds.), Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems. pp. 105–112.
Souto, A., Prates, P.A., Lourenço, A., Al Maamari, M.S., Marques, F., Taranta, D., Doó, L., Mendonça, R., Barata, J., 2021. Fleet management system for autonomous mobile robots in secure shop-floor environments. In: 2021 IEEE 30th International Symposium on Industrial Electronics. ISIE, pp. 1–6.
Trierweiler, M., Foith-Förster, P., Bauernhansl, T., 2020. Changeability of matrix assembly systems. Procedia CIRP 93, 1127–1132, 53rd CIRP Conference on Manufacturing Systems 2020.
Ueda, K., 1992. A concept for bionic manufacturing systems based on DNA-type information. In: Olling, G., Kimura, F. (Eds.), Human Aspects in Computer Integrated Manufacturing. Elsevier, Amsterdam, pp. 853–863.
Ueda, K., 2007. Emergent synthesis approaches to biological manufacturing systems. pp. 25–34.
Ueda, K., Kito, T., Fujii, N., 2006. Modeling biological manufacturing systems with bounded-rational agents. CIRP Ann. 55 (1), 469–472.
Ueda, K., Ohkura, K., 1994. A modeling of biological-oriented manufacturing systems with two types of populations. pp. 75–80.
Valckenaers, P., 2020. Perspective on holonic manufacturing systems: PROSA becomes ARTI. Comput. Ind. 120, 103226.
Valckenaers, P., Van Brussel, H., Bongaerts, L., Wyns, J., 1997. Holonic manufacturing systems. Integr. Comput.-Aided Eng. 4 (3), 191–201.
Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998. Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. 37 (3), 255–274.
Vidal, T., Bidot, J., 2001. Dynamic sequencing of tasks in simple temporal networks with uncertainty. In: CP 2001 Workshop in Constraints and Uncertainty. pp. 1–10.
Wu, H., hui Tian, G., Li, Y., yu Zhou, F., Duan, P., 2014. Spatial semantic hybrid map building and application of mobile service robot. Robot. Auton. Syst. 62 (6), 923–941.
Wu, W., Lu, J., Zhang, H., 2023. A fractal-theory-based multi-agent model of the cyber physical production system for customized products. J. Manuf. Syst. 67, 143–154.
Zheng, Y., Xiao, Y., Seo, Y., 2014. A tabu search algorithm for simultaneous machine/AGV scheduling problem. Int. J. Prod. Res. 52 (19), 5748–5763.