

Knowledge and agent-based system for decentralised scheduling in manufacturing

Saeidlou, Salman; Saadat, Mozafar; Jules, Guiovanni D.; Lou, Ping

DOI:

[10.1080/23311916.2019.1582309](https://doi.org/10.1080/23311916.2019.1582309)

License:

Creative Commons: Attribution (CC BY)

Document Version

Publisher's PDF, also known as Version of record

Citation for published version (Harvard):

Saeidlou, S, Saadat, M, Jules, GD & Lou, P 2019, 'Knowledge and agent-based system for decentralised scheduling in manufacturing', *Cogent Engineering*, vol. 6, no. 1, 1582309, pp. 1-19.
<https://doi.org/10.1080/23311916.2019.1582309>

[Link to publication on Research at Birmingham portal](#)

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

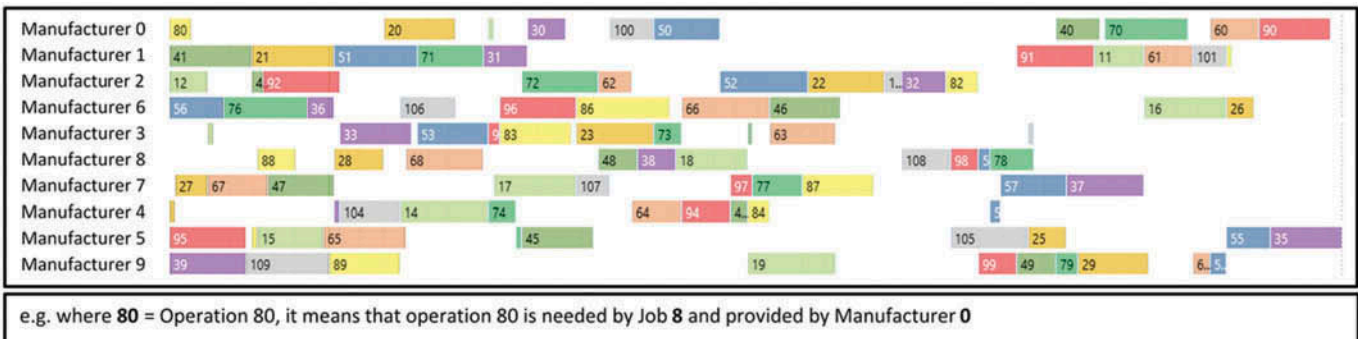
Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.



Manufacturer operation schedules for the LA19 scheduling problem with maximum make span of 1198 hours

PRODUCTION & MANUFACTURING | RESEARCH ARTICLE

Knowledge and agent-based system for decentralised scheduling in manufacturing

Salman Saeidlou, Mozafar Saadat and Guiovanni D. Jules

Cogent Engineering (2019), 6: 1582309



Received: 05 October 2018
Accepted: 05 February 2019
First Published: 18 February 2019

*Corresponding author: Salman Saeidlou, Department of Mechanical Engineering, School of Engineering, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK
E-mail: sxs989@bham.ac.uk

Reviewing editor:
Ping Lou, Wuhan University of Technology, China

Additional information is available at the end of the article

PRODUCTION & MANUFACTURING | RESEARCH ARTICLE

Knowledge and agent-based system for decentralised scheduling in manufacturing

Salman Saeidlou^{1*}, Mozafar Saadat¹ and Guiovanni D. Jules¹

Abstract: The aim of the research paper is to develop algorithms for manufacturers' agents that would allow them to sequence their own operation plans and to develop a multi-agent infrastructure to allow operation pair agents to cooperatively adjust the timing of manufacturing operations. The scheduling problem consisted of jobs with fixed process plans and of manufacturers collectively offering the necessary operations for the jobs. Manufacturer agents sequenced and pair agents timed each operation as and when required. Timing an operation triggered a cascade of conflicts along the job process plan that other pair agents would pick up on and would take action accordingly. The conventional approach performs conflict resolution in series and manufacturer agents as well as pair agents wait until they are allowed to sequence and time the next operation. The limiting assumption behind that approach was systematically removed, and the proposed approach allowed manufacturers to perform operation scheduling in parallel, cutting down tenfold on the computation time. The multi-agent infrastructure consists of the Protégé



Salman Saeidlou

ABOUT THE AUTHORS

Salman Saeidlou received the MEng (Hons.) degree in Mechanical Engineering from the University of Birmingham, Birmingham, U.K., and the Ph.D. degree in the same field of study from the University of Birmingham, Birmingham, U.K. He is currently working within the Automation and Intelligent Manufacturing (AIM) Research Group in the School of Mechanical Engineering. His research interests include holonic manufacturing, distributed systems, agent-based modeling and intelligent manufacturing systems.

Mozafar Saadat received the B.Sc. (Hons.) degree in Mechanical Engineering from the University of Surrey, Guildford, U.K., and the Ph.D. degree in industrial automation from the University of Durham, Durham, U.K. He is currently with the Department of Mechanical Engineering, School of Engineering, University of Birmingham, Birmingham, U.K.

Guiovanni D. Jules received the B.Eng. degree in mechanical engineering from the University of Birmingham, Birmingham, U.K., in 2010, and the Ph.D. degree also from the University of Birmingham. His current research interests include holonic manufacturing, distributed systems, agent-based modeling, and manufacturing networks of small and medium enterprises.

PUBLIC INTEREST STATEMENT

Research on decentralised manufacturing scheduling have been prominent during the past couple of years due to its flexibility and fault tolerant characteristic. The aim of the presented research is to develop simple algorithms for the sequencing of operations, following a framework of decentralised scheduling among manufacturers. One such framework is the network paradigm which has helped small and medium manufacturing enterprises (SMEs) to adapt to the global market trend. By combining unique and complementary knowledge as well as sharing resources as part of an inter-firm cooperative agreement, adaptability is achieved. Furthermore, decentralisation provides a cheaper alternative to accessing new technologies, in exchange of royalty fees and some efforts expended in searching, evaluating and coordinating technology providers.

knowledge base, the Pellet semantic reasoner and the Workflows and Agent Development Environment (WADE). The case studies used were the MT6, MT10 and LA19 job shop scheduling problems; and an industrial use case was provided to give context to the manufacturing environment investigated. Although there were benefits from the decentralised manufacturing system, we noted an optimality loss of 34% on the makespans. However, for scalability, our approach showed good promise.

Subjects: Artificial Intelligence; Automation; Cybernetics; Simulation & Modeling; Operations Research; Production Systems; Computer Aided Design & Manufacturing; Supply Chain Management; Systems Engineering; Intelligent Systems

Keywords: semantic web; ontology; graph database; multi-agent system; disturbance; conflict resolution

1. Introduction

The research aims to develop simple algorithms that interact for the decentralised scheduling of manufacturing operations in a network. The network paradigm has helped small and medium manufacturing enterprises (SMEs) to adapt to the global market trend by combining unique and complementary knowledge, as well as sharing resources as part of an inter-firm cooperative agreement. A decentralized network is an organisational arrangement where power is symmetrically distributed among its members, where decision making is consensus based and where the nature of leadership is informal (Müller-Seitz & Sydow, 2012). This paper is concerned with the boundary system that represents explicit rules and limits for the behaviour of a decentralised system and its constituents (Artto, Kulvik, Poskela, & Turkulainen, 2011). The boundary control systems work effectively when cognitive proximity is prevalent in the network i.e. the network needs a medium through which members can exchange their complementary knowledge (Li, Veliyath, & Tan, 2013). This medium would be in the form of a shared knowledge base. An example of a widely accepted knowledge base is the ontology (Jules, Saadat, & Saeidlou, 2013; Saeidlou, Saadat, Amini Sharifi, & Jules, 2017). In this context, the paper aimed to achieve decentralized scheduling using a multi-agent system and an ontology. On one hand, the multi-agent system is used as a framework in which simple algorithms could be designed for each agent, and new agents would easily be integrated into the existing software network that shadows the network of the manufacturers. A multi-agent system is a form of distributed artificial intelligence; consisting of multiple agents that are computational entities, capable of perceiving their environments and taking action regarding them. This software architecture allows virtual systems to be developed that represent complex and dynamic real-life systems. Such systems are required by the manufacturing industry where decentralised operations, open organisational structures and the need for market predictability are increasingly becoming important (Komma, Jain, & Mehta, 2011).

On the other hand, ontology has huge potential in automating reasoning in a boundary system of soft and hard constraints. An ontology is the explicit representation of structures, in a logic-based language, so that the structures hold domain information in a format that facilitates automated query, reasoning and cooperation among agents (Guo & Zhang, 2009; Lin, Zhang, Lou, Chu, & Cai, 2011; Monostori, Váncza, & Kumara, 2006). The Semantic Web is a form of ontology, based on the idea that decentralised and heterogeneous agents can use the web to share, integrate and process data (Leitão, 2009). The ontology semantically enriches existing sources of data and meta-data, so that machines can autonomously summon information and manipulate it (Guo & Zhang, 2009; Luck, McBurney, & Preist, 2003). Irrespective of their existing knowledge and purpose, agents that share concepts and a common language of expressing their knowledge; they can request information and services from each other (Monostori et al., 2006). However, both multi-agent systems and ontology lack the ability to optimize scheduling objectives on their own (Ouelhadj & Petrovic, 2009). The ontology and the multi-agent system can be combined to provide a sophisticated decentralised scheduling system.

Table 1. Job process plans from MT6 x 6 problem

Job	O1	O2	O3	O4	O5	O6
1	12 (1)	10 (3)	11 (6)	13 (7)	15 (3)	14 (6)
2	21 (8)	22 (5)	24 (10)	25 (10)	20 (10)	23 (4)
3	32 (5)	33 (4)	35 (8)	30 (9)	31 (1)	34 (7)
4	41 (5)	40 (5)	42 (5)	43 (3)	44 (8)	45 (9)
5	52 (9)	51 (3)	54 (5)	55 (4)	50 (3)	53 (1)
6	61 (3)	63 (3)	65 (9)	60 (10)	64 (4)	62 (1)

O1 = first operation, 45 = needed by Job 4, provided by Manufacturer 5,
 (9) = operation processing time

In this paper, a decentralized scheduling system was developed that used an ontology and a reasoner within each agent of a multi-agent system that reads and writes to the ontology. Decision making would take place using a simple indexing algorithm distributed within each job agent, and a re-indexing algorithm within each manufacturer’s agent that should result in the emergence of multi-manufacturer operation plans, from a collection of simple behaviours. The MT6, MT10 and the LA19 scheduling problems (Lawrence, 1984) were used, the former shown in Table 1. The operation research community uses these problems as benchmarks to validate their results against other researchers.

2. Related work

2.1. Multi-agent system in the manufacturing domain

Komma et al. (2011) developed an agent-based simulator for the manufacturing shop floor domain, modelling agents such as AGV-agent, machine-agent and part-agent. The components of their framework involved a knowledge base, reasoning capabilities and agent behaviours. The dispatch algorithm used in part-agents was the simple “first-come-first-served” rule. The work was developed on the Java Development Framework (JADE). Their work, in contrast with ours, did not consider tasks with sequence set-up constraints.

Barbosa, Leitão, Adam, and Trentesaux (2015) presented the ADACOR multi-agent system with a feature of stigmergy which allowed agents to pick upon message trails left in the environment. These messages signalled a plan deviation and an opportunity for self-reconfiguration. ADACOR is also built on the JADE infrastructure. In our paper, we looked at a knowledge base rather than a message trail, which we think is a more scalable approach.

Vrba and Marik (2010) presented MAST, which is a multi-agent system capable of structural reconfiguration when the layout of the factory floor changes. The factory consisted of a system of conveyors. The disturbance is simulated as a failed conveyor. The system reconfigured the virtual map and automatically searched for the shortest path for a product to reach its destination. Our work, on the other hand, looked at operation sequencing and timing in a manufacturing network.

2.2. Agent-based manufacturing scheduling

In agent-based manufacturing scheduling, research has been concerned with six main problems: shared resource conflicts; disturbance; dynamic effective capacity; resource selection criteria; the scalability of scheduling; and improvement in the state of art.

Shared resource conflicts involve independent projects that share a set of resources that are bounded by constraints of operation precedence and resource availability. The objective is to minimise delay (Adhau, Mittal, & Mittal, 2012; Jules & Saadat, 2017). In disturbance problems, uncertainty makes prior solutions suboptimal and even infeasible, unless stochastic optimisation is used to account for uncertainty, but there are drawbacks. A current research field is trying to find ways for deterministic reactive rescheduling to account for uncertainty (Harjunkoski et al., 2014). Then there is the case of flexible manufacturing, where the effective capacity is dynamic. The scheduling, in the context of a dynamic factory layout consisting of mobile robots, is a complex problem. It is a case of distributed scheduling (Giordani, Lujak, & Martinelli, 2013). Also, not all resources are adequate and therefore finding promising partners is a concern. In addition, this must be achieved in a more cost-effective manner than the conventional tendering process (Mohebbi & Shafaei, 2012). An issue is scalability, where for every increase in dependencies such as machines, operators, mobile robots, there is an exponential increase in potential solutions. Furthermore, there is detailed complexity downstream of factory job allocation, and uncertain events, such as order changes, machine failure, and operator absence. As scale increases, decisions become increasingly myopic (Alvarez, 2007; Jules & Saadat, 2017). Finally, there is the problem of making the agent technology more natural. Agent technology already inherits biological insights, such as its distributed nature, division of labour and emergence from collective simple behaviours. However, research has been focused on applying added properties of self-configuration, self-optimisation and self-healing to multi-agent systems (Leitão, Barbosa, & Trentesaux, 2012).

Three main categories of solutions can be noted across multiple literature sources: namely auction, negotiation and iteration. To solve shared resource conflicts, Adhau et al. (2012) used a multi-agent architecture consisting of agents, a communication bus and protocol and a database. The project agents bid for a time slot with the resource agents through a five-step auction-based negotiation algorithm; namely, virtual schedule and utility calculations, bid generation and bid modification by the project and resource agents. Then, the exchange agent performs the provisional winner determination, and the final resource allocation and winner determination. The results were measured in average project delay and total makespan. The data set used involved 140 multi-project instances.

To solve the issue with disturbances, Harjunkoski et al. (2014) proposed that deterministic scheduling be used in a closed-loop schedule, thus allowing frequent rescheduling. This is because deterministic scheduling is solved in a more reasonable time compared to stochastic scheduling. This approach resulted in the possibility of a longer scheduling horizon, amid more uncertainty and a faster reaction time to changes (Harjunkoski et al., 2014). In the context of dynamic effective capacity, Giordani et al. (2013) proposed a two-level scheduling algorithm, where in the first level, the tasks compete for robots through iterative auctioning; and in the second level, using the Hungarian method, robots reallocate themselves among the tasks. This approach, compared to a centralised approach, showed some weaknesses in terms of sub-optimal control policy and under-utilised resources (Giordani et al., 2013). To address the selection of resources, Mohebbi and Shafaei (2012) proposed an approach involving a rich profile of the suppliers and buyers, a feasibility analyser combined with a negotiation protocol. By consulting the profiles, the feasibility analyser generates a decision matrix with preference priority and practical feasibility. Practical feasibility is judged based on the material flow and the network welfare, determined from the profiles of the buyers and suppliers. The results were measured in terms of backlog order cost, un-utilised capacity cost and total cost; where their proposed method outperformed the conventional tendering process (Mohebbi & Shafaei, 2012). To address the issue of scalability, Alvarez (2007) proposed an iterative price adjustment method to fender off resource conflicts and improve system performance. Key to that approach is an auctioning process, supported by the contract net protocol that allows time-critical data and constraints to be communicated and used in the bid calculation model that is executed by each agent (Alvarez, 2007). Finally, concerning the state of the art, Leitão et al. (2012) proposed a potential field control architecture, where a potential field acts as a vector of resource services. The field intensity is reduced with the

distance of the resources away from the job. A decisional node consists of a matrix of potential fields. From that node, the resource that has the highest potential field gets the job. This approach compared to the contract net protocol has resulted in an average production time gain of 10% (Leitão et al., 2012).

2.3. Multi-agent system negotiation mechanism

For scheduling activities to be delegated, there are two main components required, namely interaction protocol and negotiation mechanism. The interaction protocol governs the structure and timing of the data exchange between agents; contract net protocol and modified ring protocol are useful examples (Jules, Saadat, & Saeidlou, 2015; Owliya, Saadat, Jules, Goharian, & Anane, 2013). The negotiation mechanisms can be categorized into market-based or threshold-based approaches. The market-based approach caters for agents with self-interested goals. Agents compete and are rewarded if they exhibit desirable system-wide behaviours. The threshold-based approach is based on the probability of an agent accepting a preferred type of task when some events take place. However, market-based negotiation, which is a direct negotiation mechanism, has problems with communication overhead due to the constant exchanges of bids and the processing of those bids (Goldingay & Van Mourik, 2013; Shen, 2002). Threshold-based mechanisms which are often associated with indirect negotiation mechanisms such as stigmergy and bio-inspired coordination, do not suffer from communication scalability issues. The knowledge would be in the form of pheromone-type traces in the case of the threshold-based mechanism; and in the case of the market-based mechanism, knowledge would come from agent bids. Agents would operate within a context bound by rules, implicit data and inferred data (Yilmaz & Erdur, 2012).

2.4. Knowledge-based representation of domain problem

Kotulski, Sędziwy, and Strug (2014) presented a graph transformation system for handling the storage and exchange of knowledge between agents. Due to the increasing complexity of the graph data that needed to be handled and the increasing workload imposed by multi-agent systems, the authors developed algorithms to maintain graph cohesion and to speed up graph processing. Our research made use of the open source ontology editor Protégé for knowledge storage and inline analytics of scheduling data.

In order to better understand the decision-making process with the urban goods movement that is damaging the environment, Anand, van Duin, and Tavasszy (2014) developed an agent-based model. The authors identified the stakeholder agents and their interactions and formed a model that worked in tandem with a knowledge base that represented the city logistics' domain. Model simulation enabled the authors to understand how to consolidate goods and coordinate different types of goods' movers, to improve efficiency and reduce the environmental downsides of logistics.

2.5. Collective behaviour in manufacturing systems

In order to eliminate the communication overhead that exists in manufacturing control systems, Wang et al. (2012) proposed a pheromone-based coordination approach. The approach consists of a potential field which emits a negative or positive and a strong or weak field. This allows signals to be picked up by agents that can act on them.

In our paper, we used ontology as the source of knowledge because of its scalability in terms of the variety of signals it can store.

Wang and Choi (2014) proposed a decomposition-based holonic approach (DBHA) to scheduling. The approach consisted of a genetic algorithm control (GAC) and a shortest processing time contract net protocol (SPT CNP) with the ability to switch between them. A backpropagation network was used to estimate the switch-over threshold because the GAC generates better makespan in low stochastic conditions, while SPT CNP prefers high stochastic processing times. The context was a flexible flow

shop problem with stochastic processing times. The contrast with our work is that we used fixed processing times throughout but in disturbance situations, the DBHA method has merit.

2.6. The state of the art and research tools

Workflows and Agent Development Environment (WADE) is the next generation of JADE. Coupled with the Workflow Lifecycle Management Environment, WADE allows scalable software systems to be visually programmed using workflows, actors, tasks, activities and relationships. Fundamentally, WADE enables the development of decentralised agents with unique behaviours which can send and receive synchronous and asynchronous messages, request agent services from the directory facilitator and event listening. WADE has been used in mission critical applications, notably on projects for Telecom Italia (Bergenti, Caire, & Gotta, 2012).

Conceptbase, Protégé, Racer and Pellet are tools for storing knowledge and inferring new ones using logical reasoning. Ludwig (2010) drew a comparison between a deductive database system and Semantic Web reasoning. Though he demonstrated that in-database analytics is much faster than Semantic Web reasoning, the file format used by Protégé, Racer and Pellet is more portable and particularly designed to be easily stored and retrieved from the Web. Also, the Pellet reasoner is mature software with Pellet API bindings for OWL API, the semantic file format, (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007) which enables its practical implementation in multi-agent systems.

3. Methodology

3.1. Experiment overview

In this experiment, decentralised scheduling of manufacturing tasks using a multi-agent system was investigated. The multi-agent system could read and write to an ontology. Moreover, the system implemented Pellet, which enabled agents to reason about the linked data. The reasoning is dictated by rules that are written in Semantic Web Rule Language (SWRL). Case studies of scheduling problems such as MT6, MT10 and LA19 were used to test the proposed scheduling algorithm that was embedded in job and manufacturing agents.

3.2. Industry use case

The decentralised organisation is an idealistic model which presupposes the good nature of partnering businesses in boosting the performance of the organisation. The Emilian model is such an example. Emilia-Romagna has one of the leading GDPs per capita in Europe and an export ratio with a trade surplus of 6 billion euros in 2008 for mechanical engineering products alone (Mosconi & Mantovi, 2012). Their most notable output comes from the automotive industry. The Lombardy model is another example. The region's research and innovation strategy encourages knowledge transfer and cooperation along the value chain; secondly it moves towards a horizontal integration approach while improving coordination; and finally, it co-designs and tests innovations. Lombardy is now the first Italian region in terms of number of patents registered with the European Patent Office (European Commission, 2018). We carried out our use case in Bergamo in the Lombardy region at the manufacturing group GFM s.r.l.

GFM s.r.l is a small to medium enterprise (SME) with a strong market hold in mechanical engineering services. The company has an entrepreneurial background in metal machining and design of special equipment. It has later evolved into a company focusing on upstream and downstream business activities; and for some time, outsourced all its production processes. Its business activities include customer service, planning, procurement, purchasing, quality control, Information Technology (IT), warehousing and logistics. It recently acquired a production facility to add research and development (R&D) and to reinstate engineering design to its expertise portfolio. GFM s.r.l has a network of regional SME partners with partnerships that vary in strength. Strong partnerships go beyond basic procurement, to include technology transfer, training and financial support. Strong partners operate on principles of reciprocity and weak partnerships operate on contract and profit principles. GFM acts as a provider of systemic support for co-operative and non-cooperative

organisations, feeding global knowledge into local production and helping SMEs to compete on the global stage. Such global knowledge includes technology developments, market trends and manufacturing best practices. GFM brings to a manufacturing network the strength of a large corporate structure while respecting the strengths of SMEs. The SMEs are usually family-owned or cooperatives, and are efficient and flexible at scale, innovative, independent and technologically specialised.

In the scope of manufacturing operation scheduling, GFM and its partners present a good case for decentralised scheduling for the following reasons. First, in the spirit of cooperation, GFM cannot impose a schedule onto its partners. Second, the partners are independent and do not produce exclusively for GFM. Therefore, they have other customers to consider when agreeing on a schedule. Third, companies can be more responsive in solving scheduling disturbances and communicating the solutions to affected partners while GFM settles the penalties. It is difficult for either a centralised or decentralised approach to find the optimal schedule for the use case presented. However, pushing forward in the direction of decentralised scheduling can help find useful insights.

3.3. Proposed concept for decentralised scheduling

There are two types of scheduled operation; namely, job-scheduled and manufacturer-scheduled operations. A job-scheduled operation is one which is scheduled irrespective of manufacturer constraints. Such a schedule is characterised as partial, ideal, not having idle times, having theoretical lead times and being generated by forward scheduling. That operation has a theoretical start time and finish time. A manufacturer-scheduled operation is a job-scheduled operation which is also scheduled with respect to its manufacturers' constraints. That operation has a practical start time and finish time.

A wave is a good analogy to discuss how the proposed mechanism creates a good schedule in a decentralised fashion. The creation of a manufacturer operation plan sends a wave of schedule discrepancies across a field of jobs that constitutes customer orders. A manufacturer offers operations which are the sources of discrepancies, each systematically anchored in jobs, specifically in job process plans. The delay of a single operation sends a wave where its strength depends on the properties of the transmission media. The medium is the field of jobs. The wave strength is dampened by the defects lodged in the medium, and the wave energy dissipation changes the size of the defects and displaces them. These defects are actually the idle times lodged among operations. The proposed approach dampens one wave of conflict at a time. Given that the creation of an operation plan certainly creates conflicts and probably affects idle times among operations, it becomes important that manufacturer agents and job agents systematically and cooperatively address conflict resolution with regards to those interdependencies. This is achieved in three main steps: one executed by job agents, another by each manufacturer agent and the third by both.

The first step requires all job agents to agree on which manufacturer should perform scheduling next and in consequence, send a wave of conflict. The second step requires the target manufacturer agent to index and re-index its operations in various ways, with the aim of dampening the wave as fast as possible. Then the third step requires the job agents and the target manufacturer agent to adjust operation start times. These steps are designed to be executed in series, i.e. one execution at a time, as they rely on updated information from job agents and other manufacturer agents.

3.4. Proposed multi-agent system architecture

The support agent provides a foundation for the formation of manufacturer agents and job agents. It utilises manufacturer and job information from an ontology and registers the agent details to the resource management agent (RMA) in WADE. The RMA acts as a Yellow Pages' facility where agents can look up operations and touch base with providers. Once the agents are created, the support agent terminates.

At this point, the manufacturing scheduling process is triggered. Manufacturer agents do not have operation plans yet. An operation plan is derived from the operation pairs that a manufacturer agent decides to generate and validate. The creation process of the operation pairs follows two distinct steps: 1) pair generation; and 2) pair split. These steps are empirically investigated and presented in greater detail in the Results and Discussion section.

Manufacturer agents then assert the properties of their operation pair agents into the ontology, where operation plans are automatically inferred from the pair data. There are explicitly written rules that the Pellet reasoner interprets to infer such data. Manufacturer agents and operation pair agents implement Pellet.

3.5. Ontology rule formulation overview

The ontology consists of asserted facts and rules that a reasoner, such as Pellet, uses to infer new facts. These rules define relationships among facts and set up boundary conditions within which new facts might exist. The sources of facts are as follows and defined in Table 2:

- (1) Facts about the scheduling problem include operations $o_{j,x}$ & $o_{m,y}$, precedent $spp(o_{j,x})$, successor $ppp(o_{j,x})$, processing time pt_o .

Table 2. Properties of multi-agent system and nomenclature of knowledge-based components

Role	Task	Result
Support agent	Create agents using facts from ontology	
Manufacturer agent	Represents a manufacturer (M_m)	(ma_m) Manufacturer agent for manufacturer (M_m) (M) Total number of manufacturers
Job agent	Represents a job (J_j)	(ja_j) Job agent for job (J_j) (J) Total number of jobs
Ontology	Give job (J_j) facts	$(O_{j,x})$ Operation required $(O_{j,x=1})$ First operation $(O_{j,x})$ Last operation where X is the total number of operations for J_j $(O_{adj,x})$ Operation triggering adjustment of a job schedule $(O_{zero,b})$ Operation that has no idle time $(O_{zero,B})$ First operation that has idle time
	Give manufacturer (M_m) facts	$(O_{m,y})$ Operation offered $(O_{target,y})$ Operations offered by $M_{target} \in M_m$ $(O_{target,-y})$ Operations offered by M_{target} except $O_{target,y}$ $(P_p \in P)$ Operation pairs created (O_{split}) Tentative operation splitting operation pair $[OP()]$ Its operation plan
	Give operation ($O_{m,y}$, $O_{target,y}$, $O_{adj,x}$, O_{split} , $O_{zero,b}$ & $O_{j,x}$) facts	(Y) Total number of operations for M_m $[PRE()]$ Its predecessor in the job process plan $[SUC_b()]$ Its successors in the job process plan $[PT()]$ Its processing time $[ST()]$ Its start time $[FT()]$ Its finish time (FT_{max}) Maximum finish time $[N()]$ Its normalised time budget $[I()]$ Its operation sequence $[A()]$ Its start time adjustment $[fn()]$ Function indexing operations in descending order of parameter $[W()]$ Has idle time: true or false
	Give operation pair (p) facts where ($p \in P$) is the set of all valid pairs	(po) Primary operation (so) Secondary operation

- (2) Further facts about the operations $o_{m,y}$ are inferred by the reasoner constrained by the proposed equations and guided by the proposed rules.

Manufacturers are allowed to generate their own operation plans $OP(M_m)$ with respect to their constraints, such as a rush operation or a miscellaneous operation o_{alt} . A rush operation may be a rework. A miscellaneous operation may be a machine breakdown or maintenance. However, they must all use some important scheduling rules such as rules 1–5. Manufacturers use these rules to decide how to break an indices tie, how to split an operation pair and how to avoid redundant adjustments. These rules ensure that decisions are made at all times. The bottom line is that when all operation plans are generated and superposed, the plans would emerge into complete optimal schedules. We set out to achieve such operation plans by using the proposed algorithm in Figure 1.

The proposed scheduling algorithm is unique for two main reasons. The first reason is that manufacturers can criticize the indexing of its operations in hindsight. A manufacturer can change its operation indices twice in each algorithm loop. For instance, given two operations are tied by the same operation index, the manufacturer applies a rule to break the tie. In another instance, given that two operations with consecutive indexes are adjacent, the operations form an operation pair. The operation pair is engaged by the remaining operations, aiming to split the pair. The manufacturer applies the set of rules that determines whether or not a split is achieved. The original operation indices are either validated with no split or overridden with a split.

The second reason concerns the distinct layers that lead to a complete schedule. Each manufacturer takes care of a scheduling layer. The layer can be memorised and act as a restoring point. A restoring point is a valuable asset when a disturbance occurs. By determining the source of the disturbance, the last unhealthy layer can be retrieved. The predecessor layer is certainly healthy and can be used to restore the schedules to a point where the aforementioned algorithm can perform scheduling as usual. Layers can also become training data sets for machine learning, aiming to find patterns that can reduce recovery time from disturbances.

Finding the latest operation finish time:

$$FT_{max} = \max(FT(O_{j,x})) \text{ where } 1 \leq j \leq J \tag{1}$$

Calculating the operation normalised time budget:

$$N(O_{j,x}) = \frac{FT(O_{j,x}) - FT(O_{j,x})}{FT_{max}} \tag{2}$$

Finding the job-scheduled operation having the earliest start time:

$$O_{target,1} = \{O_{j,x} | \min[ST(O_{j,x})]\} \text{ where } 1 \leq x \leq X \text{ and } 1 \leq j \leq J \tag{3}$$

Finding the manufacturer that provides $o_{target,1}$

$$M_{target} = \{M_m | O_{m,y} \equiv O_{target,1}\} \text{ where } 1 \leq m \leq M \tag{4}$$

Finding all operations that M_{target} provides:

$$O_{target,y} = \{O_{m,y} | M_{target} \equiv M_m\} \text{ where } 2 \leq y \leq Y \tag{5}$$

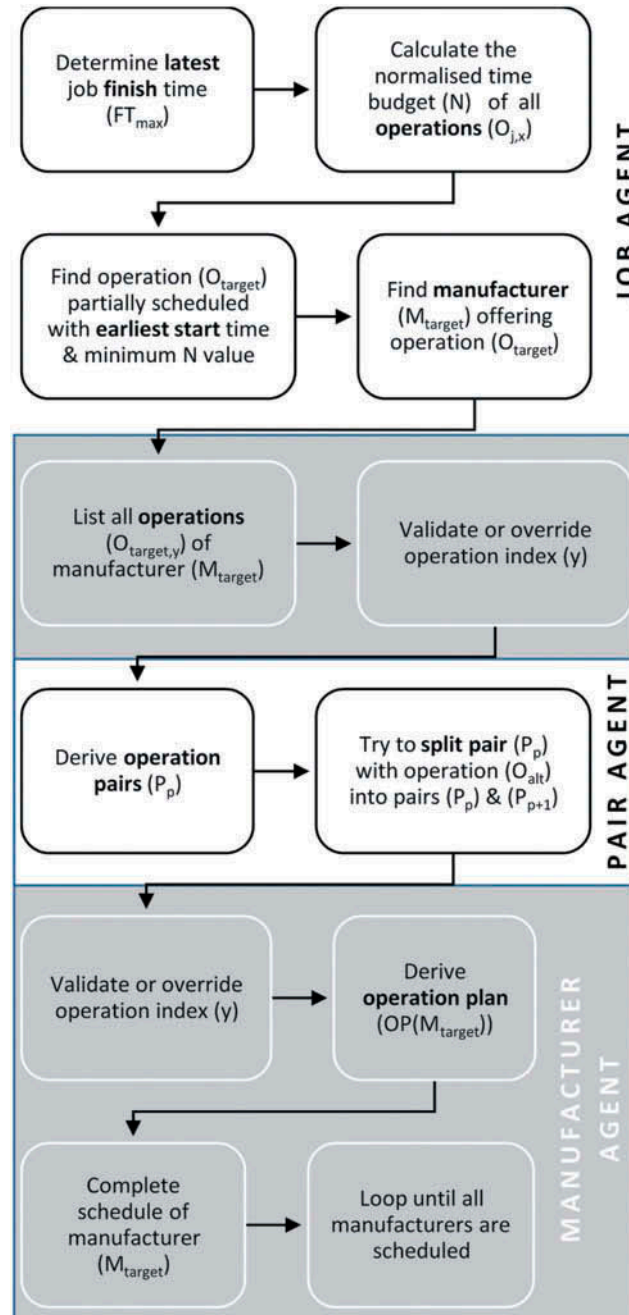
Setting operation index of $o_{target,1}$ to 1 by default:

$$I(O_{target,1}) = 1 \tag{6}$$

Determining operation indices of $o_{target,y}$:

$$I(O_{target,y}) = fn[N(O_{target,y})] \tag{7}$$

Figure 1. Interaction flowchart of system components.



Rule 1: operation with the earliest start time is prioritised if the normalised time budget of two or more operations are the same (breaking the tie among $O_{target,y}$)

$$\text{IF } N(O_{target,y}) \equiv N(O_{target,-y}) \text{ and } ST(O_{target,-y}) < ST(O_{target,y}), \text{ THEN } I(O_{target,y}) = I(O_{target,-y}) \text{ and } I(O_{target,-y}) = I(O_{target,y+1})$$

Setting operation indices of $O_{target,y}$:

$$y = I(O_{target,y}) \text{ where } 2 \leq y \leq Y \tag{8}$$

Deriving operation pairs:

$$P_p = (O_{target,y}, O_{target,y+1}) \tag{9}$$

$$po_p = O_{target,y} \tag{10}$$

$$po_p = (O_{j,x} | O_{target,y} \equiv O_{j,x}) \tag{11}$$

$$so_p = O_{target,y+1} \tag{12}$$

$$so_p = (O_{j,x} | O_{target,y+1} \equiv O_{j,x}) \tag{13}$$

Finding the best operation or a rush operation or a miscellaneous operation, to split P_p :

$$O_{split} = (O_{target,y} | \neg po_p, \neg so_p) \tag{14}$$

$$O_{zero,b} = \{SUC_b(SO_p) | W(SUC_b[SO_p]) = FALSE\} \tag{15}$$

Rule 2: the idle time is reduced by replacing a new operation acting between two adjacent operations

$$\text{IF } FT(po_p) + PT(O_{split}) + PT(so_p) + \sum_0^{B-1} PT(O_{zero,b}) \leq ST(O_{zero,B}) \text{ AND } FT(po_p) \equiv ST(O_{split}), \text{ THEN } P_p = (O_{target,y}, O_{split}) \text{ and } P_{p+1} = (O_{split}, O_{target,y+1})$$

Rule 3: validating or overriding indices of operations

$$\text{IF } O_{split} \neq \text{null}, \text{ THEN } I(O_{split}) = I(O_{target,y}) + 1 \text{ and } I(O_{target,y+1}) = I(O_{split}) + 1$$

Deriving operation plan:

$$OP(M_{target}) = \{O_{target,y}\} \tag{16}$$

Performing manufacturer-scheduling of $O_{adj,x}$:

$$A(O_{target,y}) = ST(O_{target,y}) - FT[PRE(O_{target,y})] \text{ where } A(O_{target,y}) \geq 0 \tag{17}$$

$$O_{adj,x} = (O_{j,x} | O_{j,x} \equiv O_{target,y}) \tag{18}$$

$$FT(O_{adj,x+1}) = ST[O_{adj,x}] + A(O_{adj,x}) \text{ where } x + 1 \leq 9, A(O_{adj,x}) \geq 0 \tag{19}$$

Avoiding duplicated adjustments with successors of $O_{adj,x}$:

Rule 4: adjustment of predecessors and successors in order to eliminate idle time

$$\text{IF } A(O_{adj,x}) \geq A(O_{adj,z}) \text{ THEN } A(O_{adj,z}) = -A(O_{adj,z}) \text{ where } x < z \leq X$$

Rule 5: adjustment of predecessors and successors in order to eliminate idle time

$$\text{IF } A(O_{adj,x}) \leq A(O_{adj,z}) \text{ THEN } A(O_{adj,z}) = -A(O_{adj,z}) \text{ where } x < z \leq X$$

4. Results and discussion

We investigated the hypotheses whereby we asked “How scalable is operation scheduling using operation pair selection?” and “Is operation pair selection based on time budget sufficient?” We experimented with four main functions of the manufacturer agent; namely, 1) pair generation; 2) pair pre-selection; 3) pair selection; and 4) pair conflict resolution. For operation pair generation,

we tested two approaches to reduce the space of valid solutions. Then, we looked at the feasibility of parallel execution of the selection processes; and finally, we investigated scalability with and without conflict resolution.

4.1. Results from job agents for MT6

Each loop accounts for an execution of the main steps and involves all job agents and a target manufacturer agent such as M1 for Loop A and M4 for Loop F. The manufacturer agent becomes targeted when one of its operations has the earliest start time of all job-scheduled operations. That operation is indexed as 1 which is the first position in an operation plan. Once the earliest starter is planned in, a solid foundation is laid down and it being affected by the next series of conflict waves is unlikely. If two or more operations are tied on start times, the normalised time budget (N) breaks the tie; this is similar to what happens with op21 and op12 of Table 3.

4.2. Results from manufacturer agents for MT6

Figure 2 shows the first index which is the initial indexing by a normalised time budget. The second index indicates the optimal re-indexing when Rules 1 and 2 are applied. Please note that the chart is a simple rectangular line plot that has been warped into a radial plot in order to save space. The chart illustrates that 80% of the operations were optimally indexed and that only 20% of operations necessitated correction through re-indexing.

Manufacturer agents consider the normalised time budget ($N_{O_{j,x}}$) as a rough guide for the prioritisation of its manufacturer operations. As mentioned previously, the first operation of an operation plan is decided by the start time and normalised time budget. The next operations are planned based on the normalised time budget. High N value means high operation priority within a manufacturer’s operation plan. However, in solving the MT6 scheduling problem, this approach alone indexed 80% of operations optimally but accounted for sub-optimal indexing of the remaining operations. Therefore, the agent needed to override the approach 20% of the time.

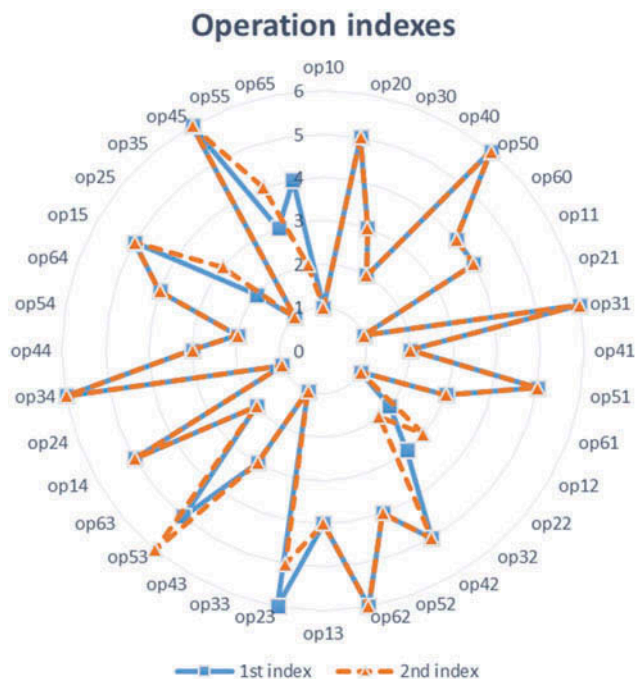
There are two problems identified with that approach. First, in the event where two or more operations scored the same N value, to break the tie the operation start times are considered. The operation with the earliest start time is prioritised which is most likely the optimal solution. This is Rule 1. However, it is still not guaranteed optimum because it may be affected by a second problem. The normalised time budget does not indicate where idle times (defects) are situated within the job process plans. An optimal prioritisation should aim to reduce the idle time between two adjacent operations. It does so by replacing an idle time with an operation. The operation splits an operation pair into two new alternative pairs, so that both the job agents as well as the manufacturer agents benefit from a reduced or no idle time. This is Rule 2. Normalised time budget, Rule 1 and 2 guarantee the optimisation of the MT6 scheduling problem.

Table 3. Mutually agreed facts sent by job agents

LOOP	FT_MAX	JOB	ST_MIN	O_TARGET	N	M_TARGET
A	47	J2	0	op21	0.83	M1
B	47	J2	0	op12	0.79	M2
C	51	J2	1	op10	0.67	M0
D	52	J2	6	op33	0.48	M3
E	53	J5	10	op35	0.32	M5
F	53	J5	13	op24	0.55	M4

$FT_MAX = FT_{max}; ST_MIN = \min[ST(O_{j,x})]; O_TARGET = O_{target, y}; N = N_{O_{j,x}}; M_TARGET = M_{target}$

Figure 2. Manufacturer agents indexing and re-indexing of operations.



4.3. Scalability of operation pair generation

Two approaches for the generation of operation pairs can be considered. The first approach generates all possible operation pairs and the manufacturer agent performs pair selection. The second proposed approach limits the number of pairs generated. The operation plan is developed from front to end. When the precedent pair is selected, the next generated or split pair must have a primary operation that is equal to the secondary operation of the previous selected pair. The total number of pair generations for the first and second approaches are $n(n - 1)$ and $\frac{n}{2}(n - 1)$ respectively.

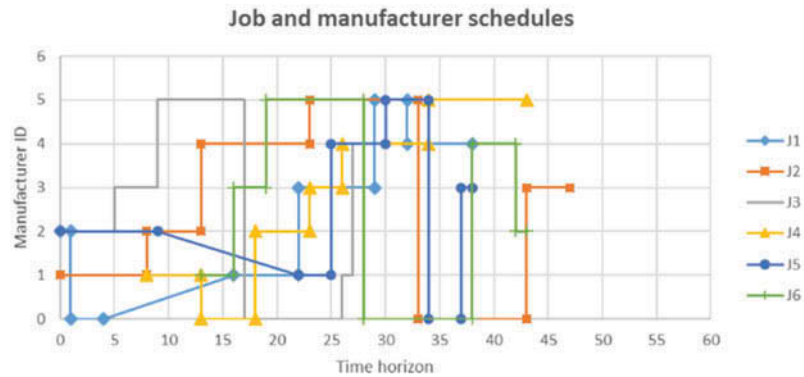
4.4. Operation pair selection

Of all the operation pairs generated, the indices of the primary and secondary operations determine which operation pair will be selected next by the manufacturer agent. The indexing of operations by manufacturer agents is defined as a function of the operations' normalised time budget, as assigned by job agents. Ascending indexing was used so that first operations are sequenced first hand, all the way down to the last operation due to forward scheduling. For the MT6 problem, a maximum makespan of 55 hours was achieved, which amounts to an optimal solution. The results for MT10 showed a maximum makespan of 1297; while the best-known solution from the literature is 930. This is comparable to an optimality loss of 33%. For the LA19 scheduling problem, a maximum makespan of 1198 was achieved, which amounts to an optimality loss of 35%, with the best-known solution of 842.

4.5. Operation pair conflict resolution

By the end of a loop, the target manufacturer operation plan is final. However, the timing of operations needs to be updated to avoid schedule overlaps for the target manufacturer. Also, the update is a pre-requisite for the next loop. Therefore, the start times of operations of the target manufacturer are adjusted until they are conflict-free with respect to the manufacturer's operation plan and jobs' process plans. Figure 3 illustrates the end of Loop A, where the operations of target manufacturer M1 are conflict-free, i.e. no overlap. Moreover, if necessary, job agents force adjustments onto the remaining manufacturer agents to become conflict-free relative to target manufacturer M1.

Figure 3. Manufacturer-scheduled operations for M1 at the end of Loop A.



Usually, it is not a necessity for operations that precede M1 operations to be adjusted, but it may happen. Table 4 illustrates three such instances. In the first instance, op20 was adjusted in Loop C but its predecessor op25 was adjusted in Loop E. In the second instance, op51 was adjusted in Loop A but its predecessor op52 was adjusted in Loop B. Finally, op50 was adjusted in Loop C but its predecessor op55 was adjusted in Loop E. Therefore, op20, op51 and op50 had to be readjusted; and in so doing, the idle times of operation pairs P(25,20) of job J2, P(52,51) and P(55,50) of job J5, were eliminated. These are Rules 4 and 5.

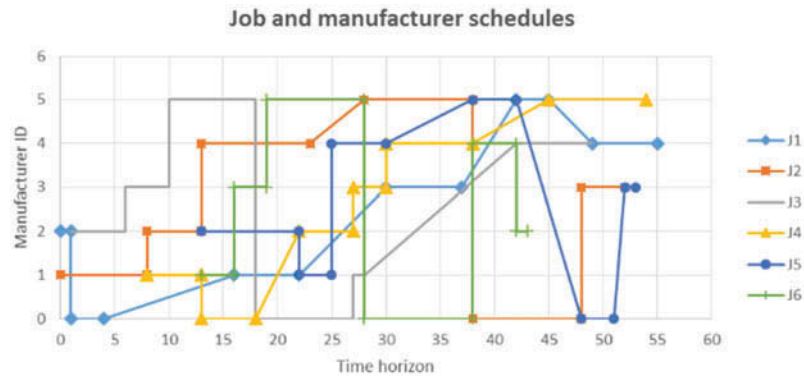
Figure 4 illustrates the end of Loop F, where all six manufacturers have adjusted their operation timings to be conflict-free. The jobs are also conflict-free and the idle times within the job process plans have been reduced as much as possible. When comparing manufacturer M1 in Figure 3, it is noted that op31 has shifted by 1 unit on the time horizon; it is due to op32 which was adjusted in Loop B, as shown in Table 4, and on which op31 depends.

Table 4. Mutually agreed adjustments between agents

O_ADJ	j	m	A	B	C	D	E	F
OP11	1	1	12					
OP13	1	3				8		
OP15	1	5					5	
OP14	1	4						4
OP25	2	5					5	
OP20	2	0			5		-5	
OP32	3	2		1				
OP34	3	4						14
OP41	4	1	8					
OP42	4	2		4				
OP43	4	3						
OP45	4	5					7	
OP52	5	2		13				
OP51	5	1	13	-13				
OP55	5	5					8	
OP50	5	0			14		-8	
OP61	6	1	13					

O_ADJ = O_{adj} ; j = job; m = manufacturer; A = Loop A; B = Loop B; C = Loop C; D = Loop D; E = Loop E; F = Loop F

Figure 4. Manufacturer-scheduled operations at the end of Loop F.



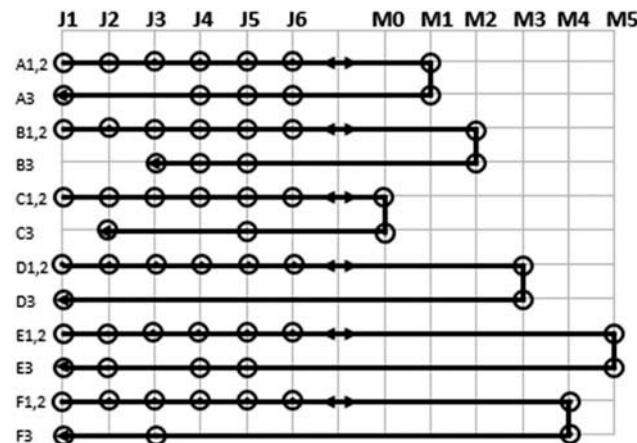
4.6. Agent interaction analysis

In each loop, the job agent communicates to the next agent about an operation which has the earliest start time and the highest N value. In Loop A and step 1, job agent J1 informs J2 agent about the operation which has the earliest start time, i.e. op12 and the operation’s N value. The J2 agent realises that it has an operation that has a start time equal to that of op12. So, to break the tie, it considers the N value. Consequently, J2 agent decides to inform J3 agent about op21. Eventually, because there was no operation having a better combination of start time and N value than op21, J6 agent communicates with the provider of op21, namely manufacturer M1 and M1 agent.

In Loop A and step 2, manufacturer agent M1 indexes its operations basing the index on N values. Op21 is indexed as 1 because, as previously mentioned, all job agents agreed on it. Operation pairs are formed from adjacently indexed operations. M1 has five operation pairs which are speculative and not yet definitive. One by one, M1 validates or splits its operation pairs. A validated operation pair also means that the indices of its primary and secondary operations have been validated. A split operation pair always means that its secondary operation has been replaced by a higher priority operation and M1 re-indexes the former secondary operation and the new one.

In Loop A and step 3, if there are conflicts with their operations, M1 agent suggests job agents to adjust the start times of the operations succeeding M1 operations. For instance, M1 agent asks J1 agent to adjust the start times of operations that succeed op11 by 12 units, i.e. op13, op15 and op14. Loops B–F are repeated in the same manner as illustrated in Figure 5.

Figure 5. Agents’ interactions during production scheduling process.



Furthermore, an investigation into the MT10 and LA19 problems has illustrated that the sequencing, timing and conflict resolution of every pair requires 480 interactions between pair agents, compared to 90 interactions if no conflict resolution is performed. This will result in a significant reduction in the computation time, reducing from 288 minutes to 29 minutes.

5. Conclusion

In this paper, we presented a scalable mechanism for the emergence of complete manufacturer operation schedules from a collection of simple decentralised algorithms. The mechanism enabled manufacturer agents to sequence their operations with respect to their own constraints, and enabled operation pair agents to cooperate in operation timing. Job shop scheduling problems MT6, MT10 and LA19 were used as case studies and an industrial use case of a decentralised manufacturing network was presented. We investigated four main functions of the manufacturer agent for scalability opportunities. It was found that agents could still generate good operation plans by executing the pair pre-selection and selection algorithms in parallel, reducing the computation time tenfold and the communication overhead fivefold. We also found that theoretically, the algorithms would systematically address disturbances such as rush jobs and delays.

Acknowledgements

The authors would like to acknowledge with thanks GFM S.r.l. for providing the use cases for the research paper.

Funding

The authors received no direct funding for this research.

Author details

Salman Saeidlou¹
E-mail: sxs989@bham.ac.uk
Mozafar Saadat¹
E-mail: M.SAADAT@bham.ac.uk
Guiovanni D. Jules¹
E-mail: gdj039@bham.ac.uk

¹ Department of Mechanical Engineering, School of Engineering, University of Birmingham, Birmingham, UK..

Citation information

Cite this article as: Knowledge and agent-based system for decentralised scheduling in manufacturing, Salman Saeidlou, Mozafar Saadat & Guiovanni D. Jules, *Cogent Engineering* (2019), 6: 1582309.

Cover image

Source: Author.

References

- Adhau, S., Mittal, M. L., & Mittal, A. (2012). A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach. *Engineering Applications of Artificial Intelligence*, 25(8), 1738–1751. doi:10.1016/j.engappai.2011.12.003
- Alvarez, E. (2007). Multi-plant production scheduling in SMEs. *Robotics and Computer-Integrated Manufacturing*, 23(6), 608–613. doi:10.1016/j.rcim.2007.02.006
- Anand, N., van Duin, R., & Tavasszy, L. (2014). Ontology-based multi-agent system for urban freight transportation. *International Journal of Urban Sciences*, 18(2), 133–153. doi:10.1080/12265934.2014.920696
- Artto, K., Kulvik, I., Poskela, J., & Turkulainen, V. (2011). The integrative role of the project management office in the front end of innovation. *International Journal of Project Management*, 29(4), 408–421. doi:10.1016/j.ijproman.2011.01.008
- Barbosa, J., Leitão, P., Adam, E., & Trentesaux, D. (2015). Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Computers in Industry*, 66, 99–111. doi:10.1016/j.compind.2014.10.011
- Bergenti, F., Caire, G., & Gotta, D. (2012). Interactive workflows with WADE. In *2012 IEEE 21st International Workshop on Enabling technologies: Infrastructure for Collaborative Enterprises* (pp. 10–15). Hammamet: IEEE.
- European Commission. (2018). COMM/ENTR/R3. Innovation policy in Lombardy - a top manufacturing region in Europe. Retrieved from <http://bit.ly/1JqLsSA>
- Giordani, S., Lujak, M., & Martinelli, F. (2013). A distributed multi-agent production planning and scheduling framework for mobile robots. *Computers and Industrial Engineering*, 64(1), 19–30. doi:10.1016/j.cie.2012.09.004
- Goldingay, H., & Van Mourik, J. (2013). The effect of load on agent-based algorithms for distributed task allocation. *Information Sciences*, 222, 66–80. doi:10.1016/j.ins.2011.06.011
- Guo, Q., & Zhang, M. (2009). A novel approach for multi-agent-based intelligent manufacturing system. *Information Sciences*, 179(18), 3079–3090. doi:10.1016/j.ins.2009.05.009
- Harjunoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., ... Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. *Computers and Chemical Engineering*, 62, 161–193. doi:10.1016/j.compchemeng.2013.12.001
- Jules, G., & Saadat, M. (2017). Agent cooperation mechanism for decentralized manufacturing scheduling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(12), 3351–3362. doi:10.1109/TSMC.2016.2578879
- Jules, G. D., Saadat, M., & Saeidlou, S. (2013). Ontological extension of PROSA for manufacturing network formation. In V. Mařík, J. L. M. Lastra, P. Skobelev (Eds.), *International conference on industrial applications of holonic and multi-agent systems. Lecture notes in computer science*, Vol. 8062. Berlin, Heidelberg: Springer.
- Jules, G. D., Saadat, M., & Saeidlou, S. (2015). Holonic ontology and interaction protocol for manufacturing network organization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(5), 819–830. doi:10.1109/TSMC.2014.2387099
- Komma, V. R., Jain, P. K., & Mehta, N. K. (2011). An approach for agent modeling in manufacturing on JADE™ reactive architecture. *The International*

- Journal of Advanced Manufacturing Technology*, 52 (9–12), 1079–1090. doi:10.1007/s00170-010-2784-2
- Kotulski, L., Sędziwi, A., & Strug, B. (2014). Translation of graph-based knowledge representation in multi-agent system. *Procedia Computer Science*, 29, 1048–1056. doi:10.1016/j.procs.2014.05.094
- Lawrence, S. (1984). *Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques (Supplement)*. Pittsburgh, PA: Graduate School of Industrial Administration, Carnegie-Mellon University.
- Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7), 979–991. doi:10.1016/j.engappai.2008.09.005
- Leitão, P., Barbosa, J., & Trentesaux, D. (2012). Bio-inspired multi-agent systems for reconfigurable manufacturing systems. *Engineering Applications of Artificial Intelligence*, 25(5), 934–944. doi:10.1016/j.engappai.2011.09.025
- Li, W., Veliyath, R., & Tan, J. (2013). Network characteristics and firm performance: An examination of the relationships in the context of a cluster. *Journal of Small Business Management*, 51(1), 1–22. doi:10.1111/jsbm.2013.51.issue-1
- Lin, L. F., Zhang, W. Y., Lou, Y. C., Chu, C. Y., & Cai, M. (2011). Developing manufacturing ontologies for knowledge reuse in distributed manufacturing environment. *International Journal of Production Research*, 49(2), 343–359. doi:10.1080/00207540903349021
- Luck, M., McBurney, P., & Preist, C. (2003). *Agent technology: Enabling next generation computing (a roadmap for agent based computing)*. Southampton: AgentLink.
- Ludwig, S. A. (2010). Comparison of a deductive database with a semantic web reasoning engine. *Knowledge-Based Systems*, 23(6), 634–642. doi:10.1016/j.knsys.2010.04.005
- Mohebbi, S., & Shafaei, R. (2012). e-Supply network coordination: The design of intelligent agents for buyer-supplier dynamic negotiations. *Journal of Intelligent Manufacturing*, 23(3), 375–391. doi:10.1007/s10845-009-0377-4
- Monostori, L., Váncza, J., & Kumara, S. R. (2006). Agent-based systems for manufacturing. *CIRP Annals-Manufacturing Technology*, 55(2), 697–720. doi:10.1016/j.cirp.2006.10.004
- Mosconi, F., & Mantovi, A. (2012). The 'Emilian Model' for the twenty-first century. Retrieved from <http://ssrn.com/abstract=1986036> 10.1094/PDIS-11-11-0999-PDN
- Müller-Seitz, G., & Sydow, J. (2012). Maneuvering between networks to lead—A longitudinal case study in the semiconductor industry. *Long Range Planning*, 45 (2–3), 105–135. doi:10.1016/j.lrp.2012.02.001
- Ouelhadj, D., & Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, 12(4), 417. doi:10.1007/s10951-008-0090-8
- Owliya, M., Saadat, M., Jules, G. G., Goharian, M., & Anane, R. (2013). Agent-based interaction protocols and topologies for manufacturing task allocation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(1), 38–52. doi:10.1109/TSMCA.2012.2192263
- Saeidlou, S., Saadat, M., Amiri Sharifi, E., & Jules, G. D. (2017). An ontology-based intelligent data query system in manufacturing networks. *Production and Manufacturing Research*, 5(1), 250–267. doi:10.1080/21693277.2017.1374887
- Shen, W. (2002). Distributed manufacturing scheduling using intelligent agents. *IEEE Intelligent Systems*, 17 (1), 88–94. doi:10.1109/5254.988492
- Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2), 51–53. doi:10.1016/j.websem.2007.03.004
- Vrba, P., & Marik, V. (2010). Capabilities of dynamic reconfiguration of multiagent-based industrial control systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(2), 213–223. doi:10.1109/TSMCA.2009.2034863
- Wang, K., & Choi, S. H. (2014). A holonic approach to flexible flow shop scheduling under stochastic processing times. *Computers and Operations Research*, 43, 157–168. doi:10.1016/j.cor.2013.09.013
- Wang, L., Tang, D. B., Gu, W. B., Zheng, K., Yuan, W. D., & Tang, D. S. (2012). Pheromone-based coordination for manufacturing system control. *Journal of Intelligent Manufacturing*, 23(3), 747–757. doi:10.1007/s10845-010-0426-z
- Yılmaz, Ö., & Erdur, R. C. (2012). iConAwa—An intelligent context-aware system. *Expert Systems with Applications*, 39(3), 2907–2918. doi:10.1016/j.eswa.2011.08.152



© 2019 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

You are free to:

Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.

You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

***Cogent Engineering* (ISSN: 2331-1916) is published by Cogent OA, part of Taylor & Francis Group.**

Publishing with Cogent OA ensures:

- Immediate, universal access to your article on publication
- High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
- Download and citation statistics for your article
- Rapid online publication
- Input from, and dialog with, expert editors and editorial boards
- Retention of full copyright of your article
- Guaranteed legacy preservation of your article
- Discounts and waivers for authors in developing regions

Submit your manuscript to a Cogent OA journal at www.CogentOA.com

