Agent Cooperation Mechanism for Decentralised Manufacturing Scheduling

Jules, Guiovanni; Saadat, Mozafar

DOI:
10.1109/TSMC.2016.2578879

Agent Cooperation Mechanism for Decentralised Manufacturing Scheduling

Guiovanni Jules, Mozafar Saadat

***Abstract*—This paper presents an agent cooperation mechanism for scheduling operations in a manufacturing network, while allowing manufacturers to absolutely control their scheduling activities. The study includes a thorough review of recent publications, a real-life industrial use case of a manufacturing network, an agent-based model of the network simulated with Recursive Porous Agent Simulation Toolkit (REPAST), the Muth and Thompson (MT10) scheduling data set, and the visualisation of results in Microsoft Project. Results of a study of a four-layer cooperation mechanism showed that for the MT10 problem, manufacturer arrangement 0-5-7-2-3-8-1-9-6-4-0 was found to maximise the utilitarian social welfare of the manufacturing network. In terms of make-span, the network achieved a maximum of 1125 which was beyond the known optimal 930. Results suggest that manufacturers could express their scheduling goals and their preferences with whom they wanted to cooperate. These were measured by the time incentive and compatibility indicators. The latter could also be used to track the optimality loss in make-span optimisation when implementing the decentralised scheduling approach in the context of manufacturing networks.**

***Index Terms*—Decentralised, Distributed, Scheduling, Agent, Welfare, Optimisation**

I. Introduction

PRODUCTION systems have encountered a paradigm shift towards mass customisation and more flexible and reconfigurable value chains. The context in which these systems operate, include multi-site independently owned facilities. Decentralised scheduling is a promising paradigm, within the aforementioned context, for effective planning and scheduling of processes. Decentralised scheduling consists of distributed entities that have enough knowledge to make decisions but only part of the knowledge to fulfil the scheduling objectives of the system. Therefore collaboration is required. Centralised scheduling consists of an entity that has all the required knowledge to fulfil the scheduling objectives of the system [1]. Centralised scheduling can be a rigid activity with respect to the dynamic demand of an uncertain market.

A scheduling problem usually involves a system of entities competing for limited resources. Deterministic scheduling problem is one where the problem parameters are known, certain and the schedule is executed exactly as planned. This is in contrast with a stochastic scheduling problem where the parameters are random variables and executional uncertainties are taken into account, such as machine breakdowns, rush orders and reworks [2]. A scheduling solution is bounded by constraints such as conflicting access to operations at a given time and limited resource capacities which contribute to the NP-hardness of the problem. Heuristic methods are often used for such problem solving and involve a model that takes inputs, constraints and decision variables [3].

The case for decentralised scheduling can be supported on the basis of two reports from the consulting companies McKinsey and American Productivity and Quality Council (APQC). McKinsey [4] proposed decentralisation as the default organisation structure unless one of three criteria is met. First criterion states that unless centralisation is mandated by law or external stakeholders, decentralisation is adequate. Second criterion states that if centralisation increase a value by at least 10%, for instance, market capitalisation, then centralisation is recommended. The final criterion is concerned with the risks of increased bureaucracy, increased business rigidity and withered motivation. If implementing centralisation could reduce the risks, then it should be implemented. So, based on a survey of 96 manufacturers, production schedule reliability was only 5% better for centralisation compared to decentralisation. Therefore production scheduling does not need to be centralised when the production sites are inherently decentralised [5].

Independent facilities should be able to decide on how to schedule their tasks as they are subject to their own constraints and goals. High value production is becoming more personalised where the concepts of vertical alliance and economy of scale are not adequate. Vertical alliance is usually associated with supply chains and requires suppliers to exclusively devote their production capacities to the top tier company. Horizontal alliances involve independent companies that usually compete but occasionally collaborate on projects that they cannot handle individually [6]. Horizontal alliance and strong collaboration are essential when key innovation come from independent small and medium entrepreneurial companies that have developed advanced technologies and processes [7]. They form part of manufacturing networks to better reach the market with adaptability and innovative products [8]. Multi-site production is uneconomical and difficult if the transfers of semi-finished goods are not tightly coordinated, a problem described in literature as very complex. In a network-type manufacturing system, planning and scheduling of manufacturing operations effectively handled by a decentralised scheduling approach has some promise.

The concept of ring network in the manufacturing domain has been introduced by Owliya et al. [9]. The researchers investigated the use of agent based model for decentralised

G. Jules and M. Saadat are with the School of Mechanical Engineering, University of Birmingham, B15 2TT Birmingham, UK.

job allocation among machines on a shop floor with the token ring principle as backdrop. Our work repurposes the essence of their previous work to the context of networks of independent manufacturers. Furthermore, our work uses a case study where interdependencies of manufacturing operations are more prominent. The context of manufacturing network organisation was investigated in our previous work [10]. We demonstrated the formation of a community of networks using various selection mechanisms. In this work, however, we focus our research on micro-mechanisms of scheduling manufacturers in a single network. A problem involving multiple interdependent issues cannot be effectively solved unless it is decomposed into self-contained sub-problems, which is addressed by Fujita et al. [11].

Our work makes use of a Muth and Thompson scheduling problem. The case study has been cited in 646 publications. The MT10 problem is a flow shop scheduling problem and involves 10 manufacturers, 10 jobs and 100 operations as shown in Table 1.

TABLE 1
JOB PROCESS PLANS FROM MUTH AND THOMPSON 10 X 10 PROBLEM

Job	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
1	10 (29)	11 (78)	12 (9)	13 (36)	14 (49)	15 (11)	16 (62)	17 (56)	18 (44)	19 (21)
2	20 (43)	22 (90)	24 (75)	29 (11)	23 (69)	21 (28)	26 (46)	25 (46)	27 (72)	28 (30)
3	31 (91)	30 (85)	33 (39)	32 (74)	38 (90)	35 (10)	37 (12)	36 (89)	39 (45)	34 (33)
4	41 (81)	42 (95)	40 (71)	44 (99)	46 (9)	48 (52)	47 (85)	43 (98)	49 (22)	45 (43)
5	52 (14)	50 (6)	51 (22)	55 (61)	53 (26)	54 (69)	58 (21)	57 (49)	59 (72)	56 (53)
6	62 (84)	61 (2)	65 (52)	63 (95)	68 (48)	69 (72)	60 (47)	66 (65)	64 (6)	67 (25)
7	71 (46)	70 (37)	73 (61)	72 (13)	76 (32)	75 (21)	79 (32)	78 (89)	77 (30)	74 (55)
8	82 (31)	80 (86)	81 (46)	85 (74)	84 (32)	86 (88)	88 (19)	89 (48)	87 (36)	83 (79)
9	90 (76)	91 (69)	93 (76)	95 (51)	92 (85)	99 (11)	96 (40)	97 (89)	94 (26)	98 (74)
10	101 (85)	100 (13)	102 (61)	106 (7)	108 (64)	109 (76)	105 (47)	103 (52)	104 (90)	107 (45)

O1 = first operation, 90 = needed by Job 9, provided by Manufacturer 0, (76) = operation processing time

Each job consists of a process plan of 10 unique operations with operation dependencies and precedencies. For instance, Job-2 has a process plan similar to 20-22-24-29-23-21-26-25-27-28 where operation 20 is the first operation with a processing time of 43 hours and operation 28 is the last operation with a processing time of 30 hours. Operations 12, 22, 32, 42, 52, 62, 72, 82, 92 and 102 share Manufacturer 2 and will all form part of the manufacturer operation plan. The objective of the problem solving is to generate the operation plans of the 10 manufacturers with respect to the process plans of the 10 jobs so that an optimal lead time of 930 hours is reached. The operation research community use this problem as one of the benchmarks to validate their results.

II. RELATED WORK

A. *Solution searching algorithms*

Problems that are computationally hard to optimise (NP-hard) are often approximately solved by moving from solution to solution within specified time bounds. They typically have a number of candidate solutions that form the search space. Usually, the objective is to find a solution that maximises a criterion such as make span or total operating cost, in the case of scheduling. A solution can be a cycle, a path or a plan [12]. Local search algorithms offer multiple ways of formulating the problems and solving them with various degree of efficiency. The established approach is the genetic algorithm. More recently developed approaches involve flower pollination and chemical reaction optimisation.

In genetic algorithm (GA), a sample of the search space of a problem is captured by a population of chromosomes. Chromosomes, by virtue of genes and alleles, map valid solutions to the problem. The fitness function quantifies the extent to which objective metrics are satisfied by these solutions. GA exploits or explores the search space by evolving the population. In exploitation, the rate of region sampling depends on the probability of good solutions in the vicinity. The population will consist of more chromosomes from higher payoff regions that from other regions. Exploration balances the mixture of chromosomes sampled from higher and lower payoff regions. Through mutation, crossover and selection, many solutions will fail and degrade the performance of the algorithm. However, after enough iterations, novel solutions are discovered [13]. Users can experiment with parameters such as population size, mutation rate and offset, to influence the exploitation and exploration of regions.

Chemical reaction optimisation is a search algorithm that mimics the nature of chemical reactions such as wall and inter-molecular collisions, synthesis and decomposition. A solution has a permutation-based and a vector-based representation and the ranking of its content is significant. In a permutation-based representation of a scheduling solution, the first vector has tasks that are separated, by delimiters, into clusters. Each cluster represents all the tasks that are allocated to a resource. The resource id is equivalent to the position, in the vector, occupied by the cluster. The second vector denotes the number of tasks allocated to each resource. It implicitly indicates the number of clusters and the position of the delimiters in the first vector. In a vector-based representation, a vector contains resources and their positions are equivalent to the allocated task id. All three vectors represent one solution. Within a chemical reaction, the solutions interact through several operations. The vectors undergo eight distinct operations and evaluation is performed, of the extent to which each solution achieves the objective metrics. This is determined by a fitness function [14].

Flower pollination algorithm has been claimed to have an exponential convergence rate. A solution vector is represented by a pollen gamete from a flower of a plant species in a patch. The pollination is associated with the transfer of pollen and its strength linked with the distance travelled by a pollen gamete. There is the long distance

step operation which is characterised by the Lévy distribution of pollen transfer via the Lévy flight of pollinators. There is the neighbourhood step operation which is characterised by a uniform distribution of transfer by rain and wind. Pollinators are attracted to flowers of the same plant species and will bypass other flowers. The combination of proximity probability, flower constancy and step operations mimic the behaviour. The proximity probability determines when a pollen gamete is transferred over long distance or when it is transferred into the neighbourhood. In other words, it is the probability of local or global pollination taking place. Flower constancy modulates the step sizes so that the means of pollination do not overreach flowers of the same plant species. Flower pollination has had significantly good performance on several test functions and has been applied in design/cost optimisation of pressure vessels [15].

Being a well-established algorithm with many developed features and integrations into simulation development platforms, GA is a simpler means of representing a scheduling problem and maintaining the representation integrity, during optimisation. Moreover, in terms of computation complexity which is relative to the number of operations per iteration, FPA and GA are equally good and better than CRO. Also, CRO and GA could be applied to various types of problems e.g. scheduling, while FPA would be best for root-finding problems. Within the context of root-finding, it has been reported that FPA can produce exact solutions in less time than GA. However, for the context of this paper, the application of GA was considered well enough for demonstrating the use of optimisation.

B. Decentralised Approaches in Manufacturing Scheduling

From literature, it is noted that the main problems that decentralised scheduling attempt to solve are the various particulars of decentralisation [16, 17, 18, 19], the intractability of black box optimisation [16], the lack of constraint relationships across multiple domains [19, 20] and dynamic scheduling [16, 18, 19].

The information used as scheduling criteria are localised and often incomplete due to decentralisation [16]. Decentralised scheduling is often attributed to task allocation to geographically distributed machines [17]. It can also be attributed to task allocation to a mix of local resources that are always available and shared resources that have limited availability. Scheduling resolves resource conflicts among shared resources by leveraging local resources [18]. And this mix of local and shared resources may constitute a single agent problem i.e. flow, open, job shop problem. Or it can be considered as a multi-agent problem i.e. production network where each task has its own objective [19]. Then there is the intractability of optimisation that is often used to determine the weight coefficients combining multiple criteria. If the information is global, complete and static, global optimisation is adequate. If the information is local, dynamic and globally incomplete, unambiguous local objectives must be achieved first [16]. In this manner, at multiple levels of decomposition, different constraints can be considered and be achieved [19, 20]. Finally, decentralised scheduling promises to tackle dynamic states [19] such as random project release times [18] and lot quantities [16].

Research in decentralised scheduling has tackled a few objectives broadly categorised into time and cost. Time objectives include cycle time, standard deviation of cycle time, and percentage of on-time completion [16], make span with transportation time [17], average project delay [18], maximum completion time and average job response time [19]. Cost objectives include total earliness and tardiness penalties, weighted tardiness cost, manufacturing costs and profit [19]. The objectives can be achieved while balancing system flexibility and solution optimality [20].

There are main methods that have been researched namely functional and physical decomposition [16, 20], interaction mechanisms [18] and system architecture [20]. In a type of physical decomposition approach, workstations have been classified based on utilisation and dispatching complexity i.e. entropy. These simpler sub problems are more tractable problems to which specific scheduling policies were applied such as decentralised WIP and speed control as well as workload control [16]. Interaction mechanism is concerned with the coordination of agents and negotiation of limited resources by agents, for overall scheduling to emerge [20]. Examples of interaction mechanisms are auction-based negotiation, multi-unit combinatorial auction [18], modified ring protocol for unsupervised task allocation in shop floors [9] and modified Contract Net Protocol (CNP) for the formation of a collaborative network organization [10]. This scheme allows the parallel generation of schedules, local decision making by agents [18], generation of alternative plans, and ranking of alternative plans, final selection of resources against plans and the merging of local schedules into a global one [20]. There are also bio-inspired interaction mechanism where potential fields whether attractive or repulsive, are used to control behaviours of the system. The potential field is formulated as a matrix correlating services and their availability [21]. Finally, system architecture is an important component of successful decentralised scheduling. Agents can be federated, form part of hierarchical structure or be autonomous. Federated agents may be associated with a broker, a mediator or a facilitator. In the case of federated facilitator architecture, agents communicate via an interface which processes incoming data from agents and routes outgoing data to appropriate agents. This contributes in limiting unintended interactions, communication overhead and facilitating complex agent management [20].

It was noted that workstation classification and the right policies can help to outperform rule-based scheduling and compound scheduling strategies by managing the number of look ahead and look back steps [16]. Through carefully designed interaction mechanisms, the proposed approach which was benchmarked against three decentralised and two centralised algorithms, outperformed with 82 out of 140 new best solutions. The approach could handle dynamic arrival of projects and was scalable on the basis of number of activities, resources and projects [18].

C. Mechanism of Agent Cooperation

Evolution and adaption in a distributed system of agents may trigger an endless cycle of chaotic behaviours. Barbosa

et al. [23] proposed a two layer stabilisation approach for a system of self-organising agents to reduce the nervous impulse of agents and the system to react to perturbations. The research used a Proportional, Integrative and Derivative (PID) controller derived from classical control theory. Applied to a manufacturing case study, a reduction in make-span performance degradation for behavioural self-organisation and a reduction in transportation times for structural self-organisation, were reported.

Wooldridge et al. [24] proposed a taxation scheme to impose different levels of costs on various agent actions while the agent seeks to minimize its expenditures. This mechanism can provide an incentive for an agent to steer clear of some actions or steer towards some actions with respect to its goals.

The social welfare, also known as collective welfare of the system measures how well agents had their goals met. The notion of utilitarian social welfare is the sum of utilities of agents and the aim is to maximise the sum without regard for the average of utilities. The score is not concerned with fairness as are egalitarian and Nash welfares [25]. A utility-based agent would compare different possible outcomes in terms of utilities and select the action that would produce the outcome with the highest utility [26]. Utility could be defined as either a measure of an agent preference for an outcome of the agent performing a particular action or simply a measure of an agent preference for an action [27]. Nguyen et al. [28] performed a computational complexity survey on social welfare optimisation namely utilitarian, egalitarian and the Nash product. It was reported that on all three notions, the complexity of optimisation is NP-complete. In other words, an exact solution can be achieved but there are no known algorithm that can efficiently solve the problem. Therefore, the computation time significantly increases with the size of the problem.

To solve the highly complex utility space with improved efficiency, Fujita et al. [29] proposed a mechanism to decompose the problem into distributed agents which, based on compatible issues, locally establish relationships with other agents to form issue clusters. A mediator aggregate the clusters into issue groups which undergo nonlinear optimisation to produce the final solution. A measurement was proposed for issue interdependency strength, optimality rate of issue grouping and quality factor. They use as their control method centralised simulated annealing. When the number of issues increased, they reported that the distributed mechanism improved the differential gradient of optimality rate as well as the quality factor.

D. *Statement of purpose*

In this paper, we introduce the idea of a mechanism for manufacturer interaction so that a desirable network schedule can emerge and lead time is minimised while manufacturers have absolute control over their scheduling activities. The study was limited to deterministic scheduling problem and decentralisation while other issues, such as executional uncertainties, lot sizing, order release strategies and dispatch rules, were not considered, in this case. The MT10 scheduling problem has always been solved by global optimisation and in a centralised manner which is the conventional way. In decentralisation, new constraints are involved namely incomplete information, federated decision making, and local entity goals in addition to the system global goals. This paper proposed four main functions involved in solving the MT10 in a decentralised manner. First, the local agent must define its goal by determining its ideal operation plan. Second, not all local agents have to interact and this is determined by their local objective scores. Third, agent federation is influenced by existing federated agents. And finally, information is explicitly exchanged for satisfying local information needs. Addressing these four functions would fulfil a gap in the decentralised scheduling literature.

Metrics were developed to measure the schedule performance at each interaction stage. We generated an agent-based model of a network, from an industrial use case, ran the model in the Recursive Porous Agent Simulation Toolkit (REPAST) with the operation research data set MT10. REPAST applied GA decentrally to produce manufacturer operation plans. GA was used for the purpose of demonstrating the use of optimisation in decentralised scheduling and is not claimed to be the best but yet is a strong method. The resulting operation plans and the fixed job-based process plans were plotted into Microsoft Project. Upon data entry, the latter automatically checks that the dependencies between the operations, either manufacturer-wise or job-wise, are valid. The scheduling of the validated operation plans is performed and the resultant lead time is benchmarked against a known optimum lead time for the MT10 data. Figure 1 shows the procedure.

Fig 1. Flowchart of proposed methodology

Merits of REPAST in the manufacturing domain have been conclusive especially when modelling distributed decision making, time scheduling and networks [26]. Moreover, the platform provides facilities for data collection, visualisation as well as an array of useful optimisation algorithms which outweigh similar platforms like MASON, NetLogo and Swarm. Furthermore, REPAST is versatile in applications ranging from industrial analysis, to social systems and evolutionary systems [29].

III. Methodology

A. Use Case of a Network-Type Manufacturing System

GFM Srl (Groupe Fabricazione Meccanica) is a private enterprise situated in the province of Bergamo, Italy. A provider of components and assembly equipment for the energy, naval, aerospace and oil/gas sector, GFM has 40 years of experience and employs 90 employees which qualifies it as a Small and Medium Enterprise (SME). The company acts as a service broker for more than 40 workshops and 500 specialised suppliers. It provides the service to the original equipment manufacturers. Moreover, it offers warehousing facilities for raw material, semi-finished and finished products transferred between workshops. However, the company owns only one production facility which it uses as a production and R&D facility. The production services consist of hundreds of operation combinations, a thousand technicians which translate into more than a million working hours per year (www.gfmsrl.com). In previous works, we have investigated the job allocation process of GFM to form a Manufacturing Network Organisation (MNO) [30]. In this paper, we focus on functions of the departments of Purchasing and Production for production scheduling in one network of MNO.

B. Agent-Based Model of a Network

The proposed model used three basic agents from the Holonic principles to investigate decentralised scheduling approach in a manufacturing network. The resource agent becomes an abstraction of the production means such as manufacturers and organisations of manufacturers. The product agent represents the product model which incorporates the complete operation plans for products specified for the customer. The order agent represents an operation state model which assures the correct execution of an operation and the on-time delivery to the next operation. For the rest of this section, the '*italic'* font will be used to refer to the class attributes of the resource, order and product agents e.g. *'jobDueDate'*.

1) Definition of main attributes

There are three new attributes introduced namely time budget, pair compatibility and network compatibility. Each operation has a time budget which determines how close to the time constraints of a job and a manufacturer, the timing of the operation is. Time budget is affected by the job due time, the operation duration and operation start time. Time budget indicates the utility of the manufacturer. Pair compatibility is an indication of how effective the timing of operations within a pair is. Pair compatibility increases when the sum of idle time within the pair and the time budget overdraft decrease. Time budget overdraft applies to a manufacturer pair and is the sum of time budget excess for the operations involved in the pair. A zero overdraft means that the operations of a pair of manufacturers have respected all the time constraints set by the jobs and the pair of manufacturers. The attributes, such as pair compatibility and network compatibility, monitor whether the operation plans of manufacturers are in harmony with the job process plans. Pair compatibility is an indication of the utility of the manufacturer pair. The network compatibility is the sum of the pair compatibilities. A valid network will have a much higher compatibility value that an invalid network and lower optimality loss. Network compatibility indicates the social welfare of the system-wide manufacturer network. In a valid network, the manufacturer operation plans have optimised operation dependencies i.e. predecessors and successors, that complement rather than conflict with the fixed operation dependencies of job process plans.

2) Cooperation mechanism of the agent-based model

The cooperation mechanism involved elements such as agents, their local objectives, a multi-agent system, its global objectives and regulations for agent behaviours. Cooperation was considered as a regulated system behaviour emerging from strategically pruned interactions of agents. Due to their decentralised nature, agents had incomplete information about the overall system and therefore cooperation is a means for agents to perceive and effectively react to the needs of the system. Cooperation allowed the agents to compromise between their local objectives and the global objectives of the system. In this paper, the compromise was regulated by the utilitarian social welfare which ensured that the global objectives prevail among agents that are also trying to achieve their local objectives. Other regulation alternatives included stabilisation from control theory [23] and taxation schemes [24]. The social welfare concept was used for its simplicity and because it aligns with the notion that it is the manufacturing networks and not individual manufacturers that compete for customer orders. The chosen method to prune agent interaction was the federated facilitator architecture [20] so that agents formed into little clusters to achieve local objectives that eventually led to a network that achieved global objectives.

3) Order Agent and Product Agent Specialisation

The agent-based model consists of the operation agent which is a specialisation of the order agent. Attributes of the operation agent form the process execution knowledge. The product agent has a specialisation called the job agent. Attributes of the job agent form the production knowledge.

4) Resource Agent Specialisation

The agent-based model proposes three types of resource agents namely Manufacturer, Manufacturer Pair and Manufacturing Network and all implement their own optimiser. Attributes of resource agents can be regarded as the process knowledge [22]. To maximise the chance of hitting target '*jobDueDate'*, manufacturer agents implement a Time Budget Utility (TBU) function. The TBU function incentivizes the rearrangement of the operation plan of the manufacturer, so that its schedule maximises the differences between *'proposedFinishTime'* and '*latestPossibleFinishTime*'. This results in the parameter *'timebudget'* created for each operation and parameter *'optimisedOperationPlan'* for the manufacturer agent.

TABLE 2
NOMENCLATURE

Symbol	Description
P_p	Manufacturer pair where $0 \le p < P$
P	No. of valid pairs
M_{1,P_p}	Primary manufacturer of pair P_p
M_{2,P_p}	Secondary manufacturer of pair P_p
M_m	Manufacturer where $0 \le m < M$
m	Manufacturer instance ID
M	No. of manufacturers
$mo_{M_m,i}$	Operation of manufacturer M_m in position i
i	Position within operation plan where $0 \le i < I$
I	No. of operations offered by manufacturer
O_o	Operation where $0 \le o < O$
o	Operation instance ID
O	No. of operations
J_j	Job where $0 \le j < J$
j	Job instance ID
J	No. of jobs
$jo_{J_j,b}$	Operation of job J_j in position b
b	Position within process plan where $0 \le b < B$
B	No. of operations required by job
TB_{O_o}	Time budget of operation O_o
FT_{O_o}	Proposed finish time
LFT_{O_o}	Latest possible finish time
ST_{O_o}	Proposed start time
D_{O_o}	Duration of operation O_o
IT_{P_p}	Idle time of pair P_p
$TB_{overdraft_{P_p}}$	Time budget overdraft of pair P_p
C_{P_p}	Compatibility of pair P_p

$$O_o \overset{\text{has}}{\Rightarrow} \{D_{O_o}, LFT_{O_o}, ST_{O_o}, FT_{O_o}, TB_{O_o}\} \quad (1)$$

$$J_j \overset{\text{needs}}{\Longrightarrow} \{jo_{J_j,0}, jo_{J_j,1}, \dots, jo_{J_j,B-1}\} \quad (2)$$

$$M_m \overset{\text{provides}}{\Longrightarrow} \{mo_{M_m,0}, mo_{M_m,1}, \dots, mo_{M_m,I-1}\} \quad (3)$$

$$P_p \overset{\text{defined by}}{\Longrightarrow} \{M_{1,P_p}, M_{2,P_p}, \mathrm{IT}_{P_p}, TB_{overdraft_{P_p}}, C_{P_p}\} \quad (4)$$

$$mo_{M_m,i} \equiv jo_{J_j,b} \equiv O_o \quad (5)$$

$$M_{1,P_p} \equiv M_{2,P_p} \equiv M_m \quad (6)$$

TBU function: $\max \sum_i^I TB_{mo_{M_m,i}}$ (7)

$$TB_{O_o} = LFT_{O_o} - FT_{O_o} \quad (8)$$

$$FT_{O_o} = ST_{O_o} + D_{O_o} \quad (9)$$

$$ST_{O_o} = ST_{mo_{M_m,i+1}} = ST_{jo_{J_j,b+1}} \quad (10)$$

Where $ST_{mo_{M_m,i+1}} \ge FT_{mo_{M_m,i}}$

Where $ST_{jo_{J_j,b+1}} \ge FT_{jo_{J_j,b}}$

Where $O_o = mo_{M_m,i+1} = jo_{J_j,b+1}$

Sample problem:

Given Job 1 is defined as J_1 with operations $jo_{J_1,1} = O_{10}, jo_{J_1,2} = O_{11}$ where $D_{O_{10}} = 1, LFT_{O_{10}} = 9$ and $D_{O_{11}} = 1, LFT_{O_{11}} = 10$

Given Job 2 is defined as J_2 with operations $jo_{J_2,1} = O_{20}, jo_{J_2,2} = O_{21}$ where $D_{O_{20}} = 2, LFT_{O_{20}} = 6$ and $D_{O_{21}} = 4, LFT_{O_{21}} = 10$

Given Manufacturer 1 is defined as M_1 with operations $mo_{M_1,\mathrm{i}} \in \{O_{10}, O_{21}\}$ and Manufacturer 2 is defined as M_2 with operations $mo_{M_2,\mathrm{i}} \in \{O_{11}, O_{20}\}$

TABLE 3
WORKED SOLUTIONS

	$\sum_i^I TB_{mo_{M_m,\mathrm{i}}}$	O_o	ST_{O_o}	FT_{O_o}	TB_{O_o}
$M_1(O_{10}, O_{21})$	12	O_{10}	0	1	8
		O_{21}	2	6	4
$M_2(O_{20}, O_{11})$	11	O_{20}	0	2	4
		O_{11}	2	3	7
$M_1(O_{21}, O_{10})$	4	O_{21}	3	7	3
		O_{10}	7	8	1
$M_2(O_{11}, O_{20})$	12	O_{11}	0	1	9
		O_{20}	1	3	3
$M_1(O_{10}, O_{21})$	10	O_{10}	0	1	8
		O_{21}	4	8	2
$M_2(O_{11}, O_{20})$	10	O_{11}	1	2	8
		O_{20}	2	4	2
$M_1(O_{21}, O_{10})$	6	O_{21}	2	6	4
		O_{10}	6	7	2
$M_2(O_{20}, O_{11})$	6	O_{20}	0	2	4
		O_{11}	7	8	2

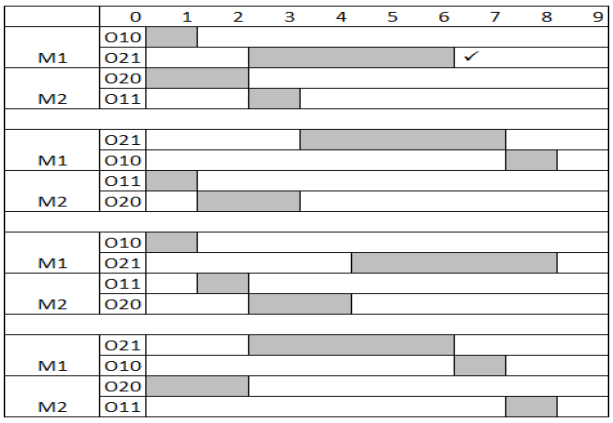

Fig 3. Gantt charts of the worked solutions

The selected operation plans for Manufacturer1 is $mo_{M_1,1} = O_{10}, mo_{M_1,2} = O_{21}$ and for Manufacturer2 is $mo_{M_2,1} = O_{11}, mo_{M_2,2} = O_{20}$, so that $max\left(\sum_i^I TB_{mo_{M_1,\mathrm{i}}}\right) = 12$ and $max\left(\sum_i^I TB_{mo_{M_2,\mathrm{i}}}\right) = 12$. The other operation plans are shown in Table 3. However, the operation plans are not compatible with each other, in terms of timing and instead Manufacturer2 should have $mo_{M_2,1} = O_{20}, mo_{M_2,2} = O_{11}$ as its operation plan. This would have resulted in the best lead

time of 6 and the least idle time as marked in Figure 3. Therefore operation plan compatibility needs to be accounted for, in selecting the right operation plan for Manufacturer2.

To maximise the utility *'pairCompatibility'* between two manufacturers, Manufacturer Pair (MP) agents implement a Pair Compatibility Utility (PCU) function. The pair compatibility is the inverse of the sum of idle time between operations and the overdraft of the time budget for rearranging operations. It is also a measure of optimality loss. A manufacturer pair consists of two manufacturers offering some operations each. The PCU function incentivizes a pair of manufacturer to reshuffle its operation positions *i* to maximise the pair compatibility. This results in an optimized arrangement of operations which is stored in the parameter *'optimisedCombinedOperationPlan'*.

PCU function: maximise C_{P_p} (11)

$$IT_{P_p} = \left|\sum_i^I \left(ST_{O_y} - FT_{O_x}\right)\right| \quad (12)$$

$$TB_{overdraft_{P_p}} \xRightarrow{if\ TB<0} \left|\sum_i^I \left(TB_{O_x} + TB_{O_y}\right)\right| \quad (13)$$

Where $O_x = mo_{M_m,i}$ where $M_m = M_{1,P_p}$

Where $O_y = mo_{M_m,a}$ where $M_m = M_{2,P_p}$

Where $a \equiv i$

Where $\{O_x,\ O_y\} \xRightarrow{part\ of} J_j$

$$C_{P_p} = \frac{1}{1+IT_{P_p}+TB_{overdraft_{P_p}}} \quad (14)$$

TABLE 4
WORKED SOLUTIONS

	P_p	O_o	IT_{P_p}	$TB_{Overdraft_{P_p}}$	C_{P_p}
$M_1(O_{10}, O_{21})$	P_1	O_{10}	(0-0) +	0	$\frac{1}{1+1+0} = 0.5$
		O_{21}	(2-1) +		
$M_2(O_{20}, O_{11})$		O_{20}	(0-0) +		
		O_{11}	(2-2) = 1		
$M_1(O_{21}, O_{10})$	P_2	O_{21}	(3-0) +	0	$\frac{1}{1+3+0} = 0.25$
		O_{10}	(7-7) +		
$M_2(O_{11}, O_{20})$		O_{11}	(0-0) +		
		O_{20}	(1-1) =3		
$M_1(O_{10}, O_{21})$	P_3	O_{10}	(0-0) +	0	$\frac{1}{1+5+0} = 0.17$
		O_{21}	(4-0) +		
$M_2(O_{11}, O_{20})$		O_{11}	(1-0) +		
		O_{20}	(2-2) = 5		
$M_1(O_{21}, O_{10})$	P_4	O_{21}	(2-0) +	0	$\frac{1}{1+7+0} = 0.13$
		O_{10}	(6-6) +		
$M_2(O_{20}, O_{11})$		O_{20}	(0-0) +		
		O_{11}	(7-2) = 7		

With the highest pair compatibility, pair P_1 was chosen from the list presented in Table 4. That pair of manufacturers has the most compatible operation plans. This means that time budget is positive and idle time is lowest. This resulted in the lowest lead times for Manufacturer1 and Manufacturer2 as noted in Figure 3. This increases the likelihood of the pair to participate in the formation and become part of a manufacturing network.

Just like two manufacturers form a manufacturer pair, several manufacturer pairs form a network. Each network agent starts with a pair group called 'optimisedpairlist'. To form a valid network, each two pairs must have a common node. The characteristics of the common node is that it is one manufacturer with an operation plan. The operation plan is shared by two pairs. The objective is to find the best shared operation plan for every two pairs so that their pair compatibilities are maximised. Each network agent runs the NCU function that triggers the right two pairs. The pair agents then run their PCU functions within the constraint of a shared operation plan. The resulting compatibility *'networkcompatibility'* is quantified by the NCU function. Network compatibility is considered as the utilitarian social welfare of a network because it is an aggregation of objective scores of manufacturer agents and pair agents.

NCU function: maximise $\sum_p^P \sum_a^P \left(C_{P_p} + C_{P_a}\right)$ (15)

Where $a \equiv p$

Where $M_{2,P_p} = M_{1,P_a}$

Where $M_{1,P_p} \neq M_{2,P_a}$

TABLE 5
SAMPLE PROBLEM

P_p	M_{1,P_p}	M_{2,P_p}	C_{P_p}
P_1	M_1	M_2	0.2
P_2	M_1	M_3	0.5
P_3	M_2	M_1	0.1
P_4	M_2	M_3	0.3
P_5	M_3	M_1	0.6
P_6	M_3	M_2	0.5

TABLE 6
WORKED SOLUTION

Network	*NCU*
$M_1M_2M_3$	0.2+0.3=0.5
$M_1M_3M_2$	0.5+0.5=1.0
$M_2M_1M_3$	0.1+0.5=0.6
$M_2M_3M_1$	0.3+0.6=0.9
$M_3M_1M_2$	0.6+0.2=0.8
$M_3M_2M_1$	0.5+0.1=0.6

The valid network $M_1M_3M_2$ consists of manufacturer pairs P_2 and P_6 and has the highest network compatibility among the list of options provided in Table 6. The social welfare of the network is the highest due to the selection of two pairs with the best shared operation plan.

5) *Decentralised Optimiser*

Every instance of resource agents executes a genetic algorithm. The decentralised feature allows optimisation to take place in parallel on separate computing threads. The optimisation among similar agents is a concurrent activity and saves computational time. However, the optimisation across different resource agent types is performed in sequence, firstly manufacturer optimisation, next manufacturer pair optimisation and finally manufacturing network optimisation. This ensures that the exchange of information across the three phases of optimization is correct and therefore the sequence needs to be respected.

C. *Simulation of Decentralised Optimisation Model*

Recursive Porous Agent Simulation ToolKit (Repast) Simphony uses Eclipse as the primary development environment. Repast Simphony (RS) provides, through the simphony application framework, the user interface tools.

In addition to core simulation functions such as scheduling, RS supports a set of independent third-party applications such as Java Genetic Algorithm Package (JGAP), Microsoft Excel spreadsheet and so on.

1) Optimisation Configuration for Manufacturer Agent

The manufacturer agent has a population of sample potential solutions called genotypes. These solutions are encoded in chromosomes made up of integer genes. The integer within the genes are called alleles and are bounded from 1 to 10 because we are dealing with operation plans with 10 operations. Genotypes are made of chromosomes where genes, in Figure 4a, have been shuffled around for each chromosome. Figure 4b represents operation ID substitution for genes of a chromosome where allele 1 represents Operation 11, allele 2 represents Operation 25 and so on.

a)	1	2	3	4	5	6	7	8	9	10
b)	11	25	30	40	52	61	70	82	91	100

Fig 4. Chromosome (a) representing operation plan (b) of Manufacturer 1.

2) Optimisation Configuration for Pair Agent

The integer within genes are bounded from 1 to 20 because we are dealing with two operation plans. The manufacturer pair agent does not perform the same operation several times and therefore the chromosome will not contain duplicate integer genes. Figure 5 presents the combined operation plan of two manufacturers encoded.

a)	1	2	3	4	5	6	7	8	9	10
	11	12	13	14	15	16	17	18	19	20
b)	16	26	37	44	59	67	74	85	96	103
	19	23	38	48	58	65	76	87	95	108

Fig 5. Chromosome (a) representing decoded combined operation plans (b) of Manufacturer 6 and 9

3) Optimisation Configuration for Network Agent

The integer within the genes are bounded from 1 to 10 because there are 10 manufacturers. The manufacturing network agent does not use the same manufacturer pair twice and therefore the chromosome will not contain duplicate integer genes. Furthermore, it is very important that no more than two pairs contain the same manufacturer. Figure 6 represents the relationship between alleles and the decoded manufacturer pair IDs whereby the combination of two adjacent alleles represent a manufacturer pair.

a)	1	2	3	4	5	6	7	8	9	10
b)	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-1
c)	4	2	9	1	5	8	6	7	3	10
d)	4-2	2-9	9-1	1-5	5-8	8-6	6-7	7-3	3-10	10-4

Fig 6. Chromosomes (a, c) representing the decoded networks (b, d)

4) Interaction mechanism between agents

In order for agents to have the right information at the right time for their optimisation process, an interaction protocol was developed. The protocol directs the outputs of six main actions to the right agents at the right time, as illustrated in Figure 7.

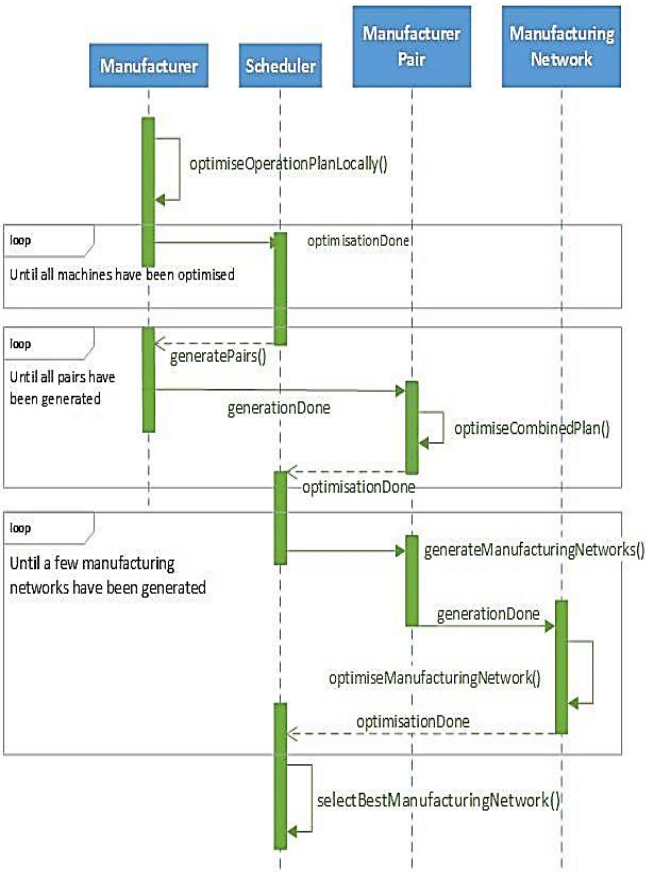

Fig 7. UML activity diagram of the interactions between agents

The object *ChromosomeApplicationData* is an optional feature of the genetic algorithm that is used to pass inputs, received by agents, to their fitness functions. For optimising a manufacturer agent, its operation plan is passed via the parameter, for manufacturer pair agent, a pair of operation plans is passed and for manufacturing network agent, a list of linked pairs is passed. These constitute the messages that were exchanged between agents.

Some of the inputs did not undergo processing and were used instead for influence. For instance, given pairs 1-6, 6-3 and 3-5. On one hand, for pair 1-6, its combined operation plan is optimised so that manufacturers 1 and 6 have jointly optimised operation plans. On the other hand, the jointly optimised operation plan of manufacturer 6 is kept fixed during the optimisation of pair 6-3. The operation plan of manufacturer 3 is optimised within the constraints of the fixed operation plan of manufacturer 6. The reason is that this prevents conflicts from occurring between pair 1-6 and pair 6-3. The same takes place for the next pair 3-5. This ensures that the optimised manufacturing network is congruent and schedules of the pairs are aligned.

5) Decentralised Optimisation Algorithm

The generic algorithm is used by all resource agents with

minor modifications for each agent type. The GASolver is an object that executes genetic algorithm on a separate thread. The solver is configured with a TBU fitness function for manufacturer agents, PCU fitness function for manufacturer pairs, and NCU for manufacturing network agents. The solver is further configured with the GreedyCrossoverOperator [31] for the crossover stage and the SwappingMutationOperator for the mutation stage. These operators will avoid duplicate alleles and are often found applied to the Travelling Salesman Problem (TSP) for that reason. Furthermore, an offset parameter can be specified to keep a part of a chromosome fixed and devoid of mutation and swapping manipulation. The population of solutions is initially created and is evolved for a set number of cycles. To increase the problem solving efficiency, the previous population is passed back to the solver at each iteration and a fraction of solutions are replaced by a new population. Best solutions have higher chances to be retained.

IV. Results

Time incentive is the performance indicator that was maximised. Fundamentally, the utility function ($\sum TB$ = 1960 hrs) ensures that manufacturers maximise the overall distance of their operations from the critical time path. Figure 9 shows the local optimisation of the operation plan for Manufacturer 1 by its agent to satisfy the utility function. The variants for optimisation were set for a population size of 500 and a swapping mutation rate of 50%.

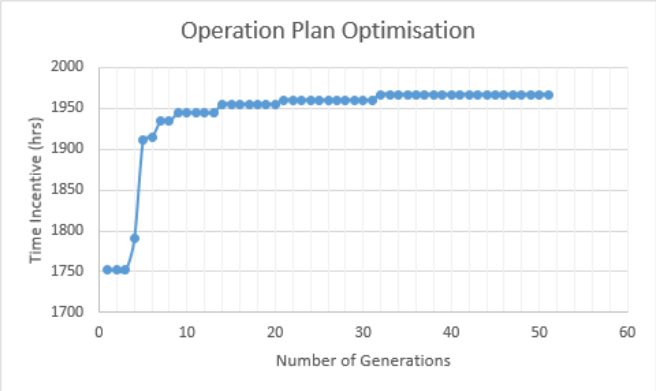

Fig. 9. GA optimization of the time incentive available to manufacturers.

After a number of GA iterations (35), the time incentive for Manufacturer 1 indicates that the ideal operation plan for Manufacturer 1 clocked a total distance of nearly 2000 hours from the critical path for the MT10 problem.

In the same scenario, the next phase carries out pairing of manufacturers, aiming at discovering the manufacturer pair that inherently will generate the best schedule. The performance indicator to maximise is compatibility which is an inverse function of idle time between two operations of the same job and the manufacturers' deficits in time incentive. The focus of this optimisation has shifted from the manufacturer alone to both a job and two manufacturers. In this phase, constraints involve the process plan of the job, operation plans of manufacturers, the job due date and the manufacturer time incentive. As expected, there are ${}^{10}C_2$ (45) pairs of manufacturers associated with compatibility scores. Figure 10 shows that the manufacturing pairing has a lower optimisation gradient than that of single manufacturer.

In the first phase, 10 operations are optimised compared to 20 operations in the second phase. Figure 11 shows the results of phase 2 where utility function is maximised and where manufacturers 6 and 9 do not have conflicting schedules. A population size of 200 and a swapping mutation rate of 25% were set and returned the best compatibility after 40 generations.

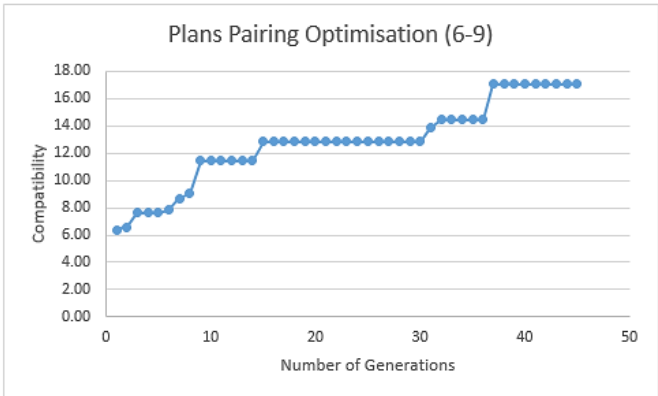

Fig. 10. GA optimisation of the compatibility of the operation plans for Manufacturer 6 and 9.

The third phase gathers valid manufacturer pairs with high compatibility. The scheduler agent then carries out GA optimisation to maximise the compatibility of the ring network being formed. The objective function takes into account the labelling of the pairs which is indicative of the manufacturers involved. For instance, given the machine pairs presented in Table 5, the pairs aggregate to form ring network 0-5-7-2-3-8-1-9-6-4-0. The network has the highest compatibility achieved using a population size of 50

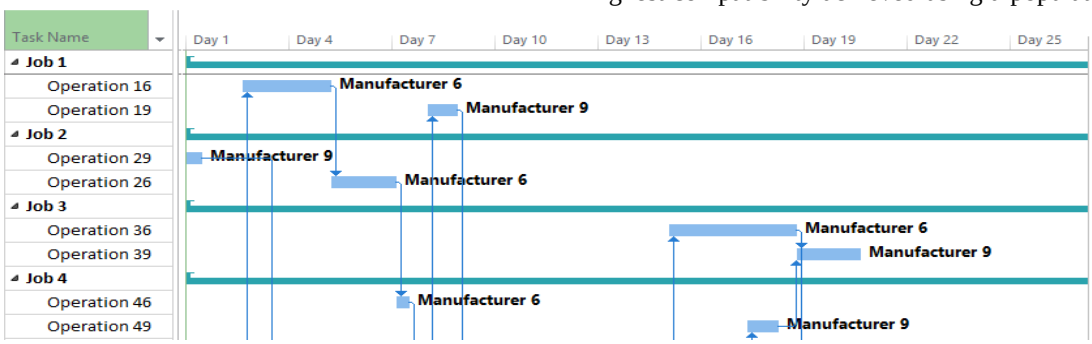

Fig. 11. Combined operation plans of Manufacturer 6 and 9 after pairing optimisation.

and a swapping mutation rate of 5%.

Finally, the fourth phase re-uses the objective function of the second phase to synchronise and re-optimise the operation plans of the ring network. New pair compatibilities (C = 39 +- 24; n = 10) are generated with coefficient of variation 0.6 compared to their initial optimisation round.

The job data and job sequences from Table 7 were used in Microsoft Project, to create a Gantt chart. Based on the methodology presented, the proposed operations plans should produce a valid Gantt chart. If not valid, Microsoft project automatically flags errors. The data from Table 7 is invalid if it creates a situation where a series of dependent operations contains an operation that links back in a way to the first operation. If the Gantt chart is valid, the results extracted from the Gantt chart are then presented in Table 8 and Table 9.

Based on the results, manufacturers and jobs had almost equal average idle times and theoretical lead times while the average actual lead time was almost twice the average theoretical lead times as presented in Table 10. Moreover, the spreads of idle times and actual lead times at manufacturers were trice the standard deviation of theoretical lead time. At the jobs, the spreads of actual lead times and theoretical lead times were almost the same. The maximum duration of the order was 1125 hrs.

TABLE 7
COOPERATIVE SCHEDULING OF PAIRS TO FORM A NETWORK

Pair	Job Sequence 1	Job Sequence 2	C
0-5	9-7-10-8-1-4-2-5-6-3-	6-7-10-8-9-5-1-3-2-4-	79
7-5	4-7-9-3-5-8-1-2-6-10-	6-7-10-8-9-5-1-3-2-4-	70
2-7	8-6-5-10-4-7-2-1-9-3-	4-7-9-3-5-8-1-2-6-10-	61
3-2	7-6-9-10-1-3-5-2-4-8-	8-6-5-10-4-7-2-1-9-3-	42
8-3	10-6-4-7-8-3-5-9-1-2-	7-6-9-10-1-3-5-2-4-8-	32
1-8	7-10-6-4-9-8-3-1-5-2-	10-6-4-7-8-3-5-9-1-2-	32
9-1	10-6-7-2-9-8-4-5-3-1-	7-10-6-4-9-8-3-1-5-2-	26
6-9	10-7-4-8-9-1-2-6-3-5-	10-6-7-2-9-8-4-5-3-1-	19
4-6	4-8-2-1-5-9-10-6-7-3-	10-7-4-8-9-1-2-6-3-5-	15
0-4	9-7-10-8-1-4-2-5-6-3-	4-8-2-1-5-9-10-6-7-3-	13

C = Compatibility

TABLE 8
RESULTS OF THE DECENTRALISED SCHEDULING OF JOBS BY NETWORK

Task Name	TD (h)	AD (h)	I (h)	Last Manufacturer
Job 1	395	1047	652	Manufacturer 9
Job 2	510	1085	575	Manufacturer 8
Job 3	568	1047	479	Manufacturer 4
Job 4	655	919	264	Manufacturer 5
Job 5	393	1022	629	Manufacturer 6
Job 6	496	1080	584	Manufacturer 7
Job 7	416	973	557	Manufacturer 4
Job 8	539	1006	467	Manufacturer 3
Job 9	597	915	318	Manufacturer 8
Job 10	540	1125	585	Manufacturer 7

TD = Theoretical Duration, AD = Actual Duration, I = Idle Time

TABLE 9
RESULTS OF THE DECENTRALISED SCHEDULING OF OPERATIONS BY THE MANUFACTURERS

Manufacturer	TD (h)	AD (h)	I (h)
Manufacturer 0	493	585	92
Manufacturer 1	548	754	206
Manufacturer 2	556	730	174
Manufacturer 3	631	1006	375
Manufacturer 4	534	1047	513
Manufacturer 5	416	919	503
Manufacturer 6	491	1022	531
Manufacturer 7	499	1125	626
Manufacturer 8	531	1085	554
Manufacturer 9	410	1047	637

TD = Theoretical Duration, AD = Actual Duration, I = Idle Time

TABLE 10
AVERAGE AND DATA SPREAD IN THE SCHEDULES

Criteria	Manufacturer	Job
Idle time (h)	421 ∓ 198	511 ∓ 130
Actual duration (h)	932 ∓ 180	1021 ∓ 70
Theoretical duration (h)	510 ∓ 66	510 ∓ 88

V. DISCUSSION

A. *Repurposing a JSSP benchmark to the network context*

In this work, we repurpose a job shop scheduling problem (JSSP) that has been used many times in research. We found that it is possible to adapt the problem to a manufacturing network scheduling problem. Considering the MT10 problem, our scheduling problem consists of ten manufacturers. Each manufacturer can perform ten unique operations. There are ten jobs that each consists of ten unique operations. Every manufacturer is involved in processing one unique operation within every job. Based on the best results obtained by previous approaches for the MT10 problem, the optimal solution is known to be 930 hrs [32]. There is little evidence to suggest that this optimal solution was not generated by centralised algorithms. Our work uses a decentralised set of algorithms, implemented by distributed agents.

B. *Discussion of results from the multi-phase scheduling mechanism*

Our concept decomposes the scheduling problem into self-contained optimisation problems. The first phase encourages manufacturer agents to sequence their manufacturing operations in order to maximise their utility functions regardless of the social welfare of the whole manufacturing network. The first phase gives manufacturers the incentive to resist change to their ideal operation plan thus limiting the solution space for an operation plan. The steep gradient of Figure 8 suggests that resistance as GA converged to solution within a few generations.

The second phase promotes the cooperation of manufacturers in pairs. Pairs maximise their utility functions regardless of the welfare of the network but now with regards to objectives of their respective manufacturers.

By maximising their utility functions, the operation plans of manufacturers synergise and scheduling conflicts are reduced to increase operation plan compatibility as shown in Figure 11. Scheduling conflicts include operation overlapping, idle time and being on critical paths. After several optimisation iterations, some manufacturer pairs significantly stand out from the rest and are more likely to proceed to the next phase. Judging by the gentle gradient of Figure 9, the solution space is much larger. The size of the optimisation problem is doubled due to involvement of two operation plans from a pair of manufacturers.

In the third phase, the network agent combines manufacturer pairs to form a ring network. Pairs that can be merged and that have high compatibilities are selected. The selected pairs have combined operation plans and in GA terms, the plans give access to good search regions where idle time, conflict, time budget overdraft and optimality loss were lowest.

In the final phase, the manufacturer pairs of the ring network perform optimisation in series. The pair that heads the network is considered to be optimised and is left as is. The next annexed pair is re-optimised relative to the previous pair where the primary manufacturer of the former is the secondary manufacturer of the latter. Consequently one operation plan is kept fixed while the other is re-optimised. This significantly narrows the regions to sample and search for a good operation plan. Eventually, all pairs are optimised relative to each other, for the benefit of the network. The network compatibility, which is the sum of pair compatibilities, is referred to as the utilitarian social welfare of the network. It is a network with less conflict, less idle time and less time budget overdraft and therefore with better lead time than alternative networks.

C. Analysis of useful outcomes

The results, from Table 10, meant that for every operation, there was an associated idle time of approximately equal length to the operation processing time. The use of the egalitarian social welfare, which is a useful indicator of fairness, would probably reduce standard deviations of idle time compared to utilitarian welfare [25]. However, the average idle times might increase. This is where the Nash product could help to reach a compromise between two notions of social welfare [25]. At the jobs, the standard deviation was possibly kept under control by the defined due times for each job. At the manufacturers, there was no such control and there lies an opportunity, for future work, to limit lead time deviations during the optimisation of manufacturer pairs and manufacturer networks. Also, the maximum actual duration of the schedule is 1125 hrs which is therefore above the optimal 930 hrs. However, optimality loss of 21% is believed to be a reasonable compromise for implementing a decentralised approach to scheduling of multi-site production in the context of manufacturing networks. And we have presented useful performance indicators, at multiple phases of the scheduling process, to enable tractability of optimality loss.

D. Contribution of the approach

The paper introduced manufacturer pair agents which act as facilitators between two manufacturer agents. All agents are able to share data via a pair agent. Some of the shared data are universal and some are unique and owned by two agents and their pair agent. This novelty enables interactions to be developed in new ways. For instance, interaction is now an agent with scalable data structure. Next, the interaction has the ability to reason about the data. Also, interaction becomes tractable and therefore can be scientifically enhanced. The federated facilitator architecture limits the pool size of pair agents and agent interactions. In doing so, it also limits the solution space for possible configuration of networks, leading to better convergence.

VI. Conclusion

This paper presented agent cooperation mechanism to allow manufacturers to schedule operations for a manufacturing network while they retain complete control over how scheduling is performed. The mechanism used time incentives and compatibility indicators to allow manufacturers to express their scheduling goals as well as their preferences for cooperation. The study approach included a review of primary research and review articles and a use case of a manufacturing network. The case study informed the design of an agent-based model of a manufacturing network which was simulated using Recursive Porous Agent Simulation Toolkit (REPAST). Muth and Thompson (MT10) scheduling data set was used as inputs to the model and the outcome was visualised in Microsoft Project. The results show that the maximisation of the utilitarian social welfare of a manufacturing network results into a maximum make span which is not the optimal but is a reasonable optimality trade-off for achieving the decentralisation of scheduling. Decentralisation of scheduling allows independently owned manufacturers to respect their own constraints and goals while engaging into cooperative scheduling. Future works would investigate functions that will incentivise agents to reduce standard deviations in pair and network lead times. An investigation of the Nash product and egalitarian social welfare would further our understanding of cooperative scheduling of decentralised agents. Also, a study would be performed on how the proposed cooperation mechanism could enable a manufacturing network to self-repair, in the context of a stochastic scheduling problem.

Acknowledgment

The authors would like to acknowledge with thanks GFM S.r.l. for the use case. The authors would also like to thank the anonymous reviewers for their constructive comments.

References

[1] G.Lo Nigro, S.Noto La Diega, G. Perrone, P. Renna (2003). Coordination policies to support decision making in distributed production planning. Robotics and Computer-Integrated Manufacturing, 19(6), pp. 521-531.

[2] H. Aytug, Mark A. Lawley, K. McKay, S. Mohan, R. Uzsoy (2005). Executing production schedules in the face of uncertainties: A review and some future directions, European Journal of Operational Research, 161(1), pp. 86-110.

[3] S. Pratap , M. Kumar B , D. Saxena , M.K. Tiwari (2015). Integrated scheduling of rake and stockyard management with ship berthing: a block based evolutionary algorithm. International Journal of Production Research, pp. 1-23

[4] A. Campbell, S. Kunisch, and G. Müller-Stewens (2011). To centralize or not to centralize? McKinsey Quarterly [Online] Available: http://bit.ly/1ORFwVL. Accessed on 17/11/2015
[5] APQC (2010). Manufacturing: Centralization versus Decentralization: Which is better? Target Third Issue [Online] Available: http://bit.ly/2146UDK. Accessed on 17/11/2015
[6] E. Todeva, D. Knoke, (2005). Strategic alliances and models of collaboration, Management Decision 43(1), pp. 123 – 148.
[7] C.A. Soosay, P.W. Hyland, and M. Ferrer (2008). Supply chain collaboration: capabilities for continuous innovation. Supply Chain Management: An International Journal 13(2), pp. 160-169.
[8] D. Mourtzis and M. Doukas (2013). Decentralized manufacturing systems review: challenges and outlook. Robust Manufacturing Control. Springer Berlin Heidelberg, pp. 355-369.
[9] M. Owliya, M. Saadat, G. Jules, M. Goharian, & R. Anane (2013). Agent-Based Interaction Protocols and Topologies for Manufacturing Task Allocation. Systems, Man, and Cybernetics: Systems, IEEE Transactions on, 43(1), 38-52.
[10] G. Jules, M. Saadat, S. Saeidlou (2015). Holonic Ontology and Interaction Protocol for Manufacturing Network Organisation," Systems, Man, and Cybernetics: Systems, IEEE Transactions Repository, Paper SMCA-13-10-0535.
[11] K. Fujita, T. Ito, & M. Klein (2014). Efficient issue-grouping approach for multiple interdependent issues negotiation between exaggerator agents. Decision Support Systems, 60, pp. 10-17.
[12] W. Michiels, Emile Aarts, and Jan Korst (2007). Theoretical aspects of local search. Springer Science & Business Media.
[13] J. Holland (1992). Genetic algorithms.
[14] J. Xu, Albert Lam, and Victor OK Li (2011). Chemical reaction optimization for task scheduling in grid computing. Parallel and Distributed Systems, IEEE Transactions on, pp. 1624-1631.
[15] X. Yang (2012). Flower pollination algorithm for global optimization, in: Unconventional Computation and Natural Computation 2012, Lecture Notes in Computer Science, 7445, pp. 240-249.
[16] S. Yao, Z. Jiang, N. Li, N. Geng, & X. Liu (2011). A decentralised multi-objective scheduling methodology for semiconductor manufacturing. International Journal of Production Research, 49(24), pp. 7227-7252.
[17] F. T. Chan, A. Prakash, H. L. Ma, & C. S. Wong (2013). A hybrid Tabu sample-sort simulated annealing approach for solving distributed scheduling problem. International Journal of Production Research, 51(9), pp. 2602-2619.
[18] S. Adhau, M. L. Mittal, & A. Mittal (2012). A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach. Engineering Applications of Artificial Intelligence, 25(8), pp. 1738-1751.
[19] J. Behnamian, & S. F. Ghomi (2016). A survey of multi-factory scheduling. Journal of Intelligent Manufacturing, 27(1), pp. 231-249.
[20] W. Shen, L. Wang, & Q. Hao (2006). Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 36(4), pp. 563-577.
[21] 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.
[22] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, & P. Peeters (1998). Reference architecture for holonic manufacturing systems: PROSA. Computers in industry, 37(3), pp. 255-274.
[23] J. Barbosa, P. Leitão, E. Adam & D. Trentesaux. (2015). Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. Computers in Industry, 66, 99-111.
[24] Wooldridge, M., Endriss, U., Kraus, S., & Lang, J. (2013). Incentive engineering for Boolean games. Artificial Intelligence, 195, 418-439.
[25] S. R. K. Doost (2008). Nash Social Welfare in Multiagent Resource Allocation (Doctoral dissertation, Universiteit van Amsterdam).
[26] K. Matsumoto, M. Matsumoto, H. Abe (2006). Goal-based action selection and utility-based action bias. Neural Networks, 19(8), 1315-1320
[27] A. Blokhin (2015). *What is the utility function and how is it calculated? | Investopedia*. [online] Investopedia. Available at http://bit.ly/1SbxGr8 [Accessed 10 Jan. 2016].
[28] N. T. Nguyen, T. T. Nguyen, M. Roos, & J. Rothe (2014). Computational complexity and approximability of social welfare optimization in multiagent resource allocation. Autonomous agents and multi-agent systems, 28(2), pp. 256-289.
[29] M. J. North, N. T. Collier, J. Ozik, E. R.Tatara, C. M. Macal, M. Bragen, & P. Sydelko (2013). Complex adaptive systems modeling with repast simphony. Complex adaptive systems modeling, 1(1), pp. 1-26.
[30] G. Jules, M. Saadat, N. Li (2012). Holonic multi-agent system: Simulation of manufacturing network formation," Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on, pp. 686,691, 14-17 Oct. 2012
[31] J. Grefenstette, R. Gopal, R. Rosmaita, and D. Gucht (1985). Genetic algorithms for the traveling salesman problem. In Proceedings of the Second International Conference on Genetic Algorithms. Lawrence Eribaum Associates, Mahwah, NJ
[32] B. J. Park, H. R. Choi & H. S. Kim (2003). A hybrid genetic algorithm for the job shop scheduling problems. Computers & industrial engineering, 45(4), pp. 597-613.

Guiovanni D. Jules (M'10) received the B.Eng. degree in Mechanical Engineering from the University of Birmingham, Birmingham, U.K. in 2010, where he is currently working towards the Ph.D. degree 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 modelling and manufacturing networks of small and medium enterprises.

Mozafar Saadat received the B.Sc. (Hons.) degree in Mechanical Engineering from the University of Surrey, Surrey, U.K., and the Ph.D. degree in industrial automation from the University of Durham, Durham, U.K. He has received various research funding in aerospace, automation and manufacturing industries, and published a wide range of peer-reviewed technical papers and editorial articles. He is currently with the Department of Mechanical Engineering, School of Engineering, University of Birmingham, Birmingham, U.K., where he leads the Automation and Intelligent Manufacturing (AIM) Research Group.