Distributed Job Scheduling Using Multi-Agent System

Sandeep Kumar and Bhupesh Kumar Lad
Industrial and Systems Engineering, Discipline of Mechanical Engineering, Indian Institute of Technology Indore, Indore, 453 552, India
kumar.s.iiti@gmail.com, bklad@iiti.ac.in

Maharshi Harshadbhai Dhada
Department of Engineering, Institute for Manufacturing, University of Cambridge, CB3 0FS, UK
mhd37@cam.ac.uk

Kshitij Bakliwal
Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA
kbakliwal@gmail.com

Abstract
The current paradigm shift in manufacturing, widely known as Industry 4.0, exists at the nexus of advances happening in computer science, sensing technologies, Information and Communication Technology, and big data analytics. It brings together Internet of Things, social networking, and advanced analytics to meet the growing need of personalized production at lower costs, by integrating human-like social capabilities into the assets of an industrial system. In this paper, we propose an innovative Multi-Agent System based distributed operations planning approach for scheduling of jobs in a parallel machine shop-floor. The approach harnesses the capabilities of Cyber-Physical Systems formed by bringing together physical machines, and various functional divisions, with their cyber space, or agents. These agents interact with one another to form a network of social machines. Using distributed decision-making and communications within the network of social assets, we tackle the complex, NP-hard problem of job scheduling, and compare the results with that of conventional centralized operations planning approach. The advantages of the proposed approach are clear in terms of reduction in computation time and lateness, and the flexibility offered by the distributed approach.

Keywords
Distributed Approach, Centralized Approach, Multi-Agent System, Job Scheduling, Industry 4.0

1. Introduction
A supervisory control system is essential for a manufacturing system to fulfill its utility. Dilts et al. (1991) have classified the manufacturing control systems into four types viz. centralized, hierarchical, semi-hierarchical and heterarchical. This classification is based on the distribution of the decision-making power. Centralized systems are characterized by a single node responsible for all the system level decision-making. The hierarchical and semi-hierarchical systems have several control levels, thereby allowing partial distribution of decision-making power. On the other hand, heterarchical systems have no fixed relationships within the constituent decision-makers.

Traditionally, the manufacturing control systems have been all centralized or hierarchical. Such systems are useful for mass production or for larger batch sizes. However they fail to address the challenges of growing consumer demands for high quality customized products with shorter life cycles, and the rising requirements of flexibility, expansibility, agility and re-configurability for the manufacturing systems (Leitão, 2009; Hu, 2013). Leitão (2009) points out that the distributed and intelligent control systems satisfying these needs are different from the ones conventionally used. Characterized by autonomous and local decision-makers, the distributed heterarchical systems have strong tolerance to the disturbances and ease of expansion, although at the cost of reduced global-optimization (Leitão, 2009). Of the distributed systems, Multi-Agent Systems (MAS) and Holonic Manufacturing Systems (HMS) have received major interest in academia, and in industry. MAS aim at dividing a system level task (or goal) into numerous sub-tasks being allotted to the entities which comprise the system. These entities, which interact with one another to achieve their goals, are called agents. HMS are a manufacturing-centric approach analogous to Koestler's idea of biological holons, where they interact with other holons at the same level, but are also a part of other higher level holons (Giret and Botti, 2004).

Owing to the path-breaking advances in the sensing technologies, computers and the information and communication technologies in the last decade, today's industries are capable of implementing the distributed control systems (Ganchev and O'Droma, 2016; Unland, 2015). The cost of both sensors, and that of storing the data, has decreased. This reduced cost has made it possible to extensively instrument the industrial assets and generate prodigious data, simulating the real-time operating condition of corresponding asset. However, it is both important, and technologically demanding, to handle and analyze this big data to optimize both asset and system-level performance. The reduced size and increased ability of the modern computers makes it feasible for the modern industries to use the Cyber-Physical Systems (CPS). CPS is the synergy between the physical and the cyber space, to analyze the asset data. In the form of a micro-controller, or a mobile device, a computer (cyber space) can be embedded in an asset (physical space) to analyze the asset data. This creates a feedback loop where the physical processes affect the computations and vice versa (Lee, 2008; Jeschke et al., 2017).

Apart from analyzing the asset data, the embedded computer is also capable of sharing the data, over cloud, with those associated with several other assets in the system to form a network of assets. This network of assets is called the Industrial Internet of Things (IIoT) and has proven beneficial for sectors such as healthcare, airlines, etc. (Annunziata and Evans, 2012; Da, 2014). There are scientific evidences for a group of people under certain conditions arriving at better solutions to a common problem than an individual person (Surowiecki, 2004), following which an IIoT can be enhanced to a Social Internet of Things (SIoT). In a SIoT the assets not only share the data with one another, but also identify other similar assets/ friends and collaborate with one another thus showing a human-like social behavior. SIoT, which has witnessed applications in areas like those of traffic-routing and product life-cycle management, can also be extended to improve system-level performance in an asset fleet (Li et al., 2018).

In words of Annuziata (2013), the machines today are not only intelligent, but brilliant. Brilliant in the sense that they are self-aware, reactive, social and predictive. Though holding great promise for various industrial systems, the resource intensive domain of manufacturing provides abundant data and scope to embrace the benefits of the above mentioned developments under the umbrella of Industry 4.0 or Smart Manufacturing. Industry 4.0 aims at bringing together physical, embedded and IT systems, and the internet to improve the traditional manufacturing systems (Lasi et al., 2014). The data-intensive Industry 4.0 harnesses these innovations, and uses connected assets and distributed decision-making to improve the system-level performance.

Production scheduling is a critical task in a manufacturing facility. The overall system performance and other manufacturing activities, such as the maintenance plan, depend on the job schedule prepared. Theoretically, optimal or near-optimal job schedules for a fixed temporal horizon are prepared using several constraints and assuming complete information about the system. The constraints have to be set because of the limited computing power of a conventional centralized decision-making system. A schedule prepared in such a static environment using probabilistic models has limited application for the real-world systems because the real-world manufacturing facilities are dynamic in nature (Hall and Potts, 2004). Dynamic in the sense that many disturbances may arise while the plan is being executed- a machine may fail, rush hour demand may arise, etc. This renders a gap between theory and practice of job scheduling (Vieira et al., 2003). However, recent research in this field is focused on using the heterarchical control systems to address the practical challenges of job scheduling.

In this paper, we present a job scheduling algorithm for a parallel machine shop-floor which relies on a distributed control system- a MAS, with each asset and functional divisions of the system represented by an agent. In contrast to the conventional methods, where a single entity i.e. the Production Planning and Control (PPC) division, create job schedules for all the machines for a fixed temporal horizon, the algorithm described here involves coordination among multiple agents of the system, i.e. the machines and the PPC division. Such distributed approach for the complex combinatorial, more specifically a NP-hard problem (Ovacik and Uzsoy, 2012) of job scheduling has not been employed before. In the results section, we demonstrate the significant advantage this approach offers when compared with the conventional centralized approach.

Section 2 reviews the Multi-Agent systems in the Subsection 2.1 and their application in the Subsection 2.2. The problem description and the objective are given in the Section 3, followed by the distributed approach for job-sequencing in the Section 4. A centralized approach for making the job sequences is given in the Section 5, and the results are discussed in the Section 6. Lastly, conclusion with future work is presented in section 7.

2. Literature Review

2.1 The Multi-Agent Systems

Multi-Agent systems deal with “behavior management in collections of several independent entities, or agents”. MAS comprise of one or more agents, which interact with one another to achieve their goals and in turn the system approaches the overall global objective (Stone and Veloso, 2000). Leitão (2009) defines an agent as “an autonomous component that represents physical or logical objects in the system, capable to act in order to achieve its goals, and being able to interact with other agents, when it does not possess knowledge and skills to reach alone its objectives”. In a manufacturing facility, the decision-making ability of agents is realized by the use of CPS architecture, where the assets are each embedded with a computer. These agents can correspond to each machine, the functional divisions, the raw materials, the labor available, the products, etc. The distributed manufacturing systems offer advantages like parallel computing and tolerance to disturbances which reduce the computation time and make it possible for the system to accommodate the unforeseen disturbances such as a machine failure. The literature offers several examples of applying these for manufacturing operations planning.

2.2 MAS application in Manufacturing Operations Planning

The current research trends show increasing use of heterarchical control systems for dealing with the manufacturing operations planning. Duffie and Piper (1987) were the first to show and demonstrate the reduced complexity, flexibility and increased fault-tolerance and modularity offered by heterarchical architecture for manufacturing operations planning. Since then, we see several instances of agent based control- Giordani et al. (2013) made use of two-layer decentralized MAS to tackle the problem of production planning and scheduling. Here, the tasks and robots who perform those tasks are represented by agents which compete with one another for their preferences. Mönch and Drießel (2005) have showed that the parallel implementation of the simple algorithms such as the Shifting Bottleneck Heuristics is faster than the sequential versions. They use a two-layered hierarchical approach to decompose the scheduling problem into numerous sub-problems which are assigned to different machines. Lou et al. (2012) have used a proactive-reactive approach to tackle the job scheduling problem where a proactive schedule is prepared which can be modified while the system operates. This algorithm uses a blackboard-approach where the machines and the scheduler can communicate with one another. Upasani et al. (2017) implemented agent based control for maintenance planning for manufacturing shop-floor. Here the machine agents are intelligent enough to identify the best maintenance schedule for them, and collaborate with the Maintenance Department to ensure the system has enough labor to fulfill the same. They have compared the use of memetic, particle swarm, and brute force algorithms at the machine level, and shown the advantages of distributed maintenance scheduling over the conventional centralized ones.

Several more examples are found in the recent literature which glorifies the use of MAS for the manufacturing operations planning (Martin et al., 2016; Sahin et al., 2017; Xiong and Fu, 2018). The approaches discussed above all show the benefits of either the reduced algorithm run-time or the increased flexibility. Our algorithm differs from the above examples on two grounds. First, we consider the machine reliability also while generating the job sequences. The machine reliability is calculated by the machine agents using the data generated from the machine. Second, while generating the job sequence, we schedule the Preventive Maintenance Jobs (PM-Jobs) also.

Apart from directly using agents for parallel computation, many researchers have developed architectures which enable us to easily fuse the MAS approach with the manufacturing systems. Christensen (2003) proposed an architecture where agents focus on deliberative tasks on a higher level, while lower-level agents focus on real-time constrained control tasks. Bagheri et al. (2015) presented a step-wise approach to realize the application of CPS for the manufacturing systems. Bakliwal et al. (2018) have presented MAS architecture to implement collaborative learning for the industrial assets. They refer to the Digital Twins as the computational entities corresponding to the assets. These communicate with their friends (other similar assets) through a central platform.

The algorithm presented in this paper is a realization of using the emergent technologies for distributed control for scheduling of jobs.

3. Problem Description and Objective

In this paper, we have considered a job shop having parallel machines. We assume that the machines are made up of single component. These machines are identical in the sense that the jobs in demand can be processed on any of the machines. In addition to that, the time taken for production of the jobs is same irrespective of the machine which processes the job. However, the machines can be different from one another in terms of their reliability, and their age. The reliability of the machines is characterized using the two-parameter Weibull probability distribution, viz.

eta (Ƞ)- the scale parameter and beta (β) - the shape parameter. These Weibull parameters are computed, using the historical data pertaining to the failure of the asset, by their corresponding agents. One way of evaluating the Weibull parameters for the machines is by the use of RNN demonstrated by Palau et al. (2018). We also assume that there is only one mode of failure for the machines and hence a set of Weibull parameters completely describe a machine's operating condition.

The PPC department receives the job demand and is responsible for creating final job schedule for all the machines in the industry. The job demand consists of the job names, along with their corresponding job descriptions. The job descriptions comprise of their processing times, the due date of the jobs, and the penalty costs per hour associated with late production of the jobs. In our distributed approach, the PPC coordinates with the machine agents to fulfill its task of producing the enterprise-level job schedule.

In addition, we consider the Preventive Maintenance (PM) tasks. The PM task, which we refer to as a PM-Job, can be scheduled between the processing of two consecutive jobs. This possibility of scheduling a PM-Job is referred to as a PM opportunity. It is assumed in our problem that there is no shortage of labor and, the PM-Job can be scheduled whenever needed at a PM opportunity. The PM-Job is characterized using two parameters- the Time to Repair (TTR), and the Restoration Factor (RF). The TTR is the downtime a machine would encounter when the PM-Job is scheduled. The RF ranges from 0 to 1. It is the factor by which the life of the machine gets restored at the end of that PM-Job i.e. the factor of reduction in the machine age (Kijima, 1989). For example, if machine age is 1000hr, then a PM-Job with RF of 0.6 would reduce the machine' age to 400hr.

It is important to schedule PM-Jobs in an optimal way. If the frequency of PM-Jobs is high, then the machine would not be able to process enough jobs in its shift duration due to increased maintenance downtime. At the same time, too few PM-Jobs would imply the increased probability of machine failure and unplanned downtime.

The PPC and machines are the two levels which collaborate with one another via their agents, and operate local computations over their asset data to produce an enterprise-level job-schedule. The jobs description and the machine failure/repair characteristics we used for our illustration are described in the Tables 1 and 2 respectively.

Table 1: Job's Properties

Job Name	J1	J2	J3	J4	J5	J6	J7	J8	J9
Processing Time (t_i^{PT}) in hours	55	85	205	105	155	185	225	135	225
Due Time (t_i^d) in hours	260	260	260	260	260	260	260	260	260
Penalty cost (per job per hour) in Rupees	10	10	10	10	10	10	10	10	10

Table 2: Machine Operating Condition Parameters

Machine id	M1	M2	M3	M4	M5	Time to Repair (t_i^{pm}) in hours	Restoration Factor
beta (β)	2	2	2	2	2	8	0.6
eta (Ƞ) in hours	1000	1000	1000	1000	1000		
age (t) in hours	0	973	1969	2319	2497		

Assumptions

Following generic assumptions are made in the problem: a) A job cannot be pre-empted by another job; b) Each job is available at the start of production schedule; c) At the beginning of production schedule machines are available; d) Machine can process only one job at a time; e) Machine always produces items of acceptable quality.

Our objective is to prepare enterprise-level job-schedule which minimizes the penalty cost. As seen in the job descriptions, the penalty cost is the same for all our jobs, hence minimizing the penalty would mean minimizing the lateness of production of the jobs. Each machine would finally be allotted a sequence of jobs. In the following sections, we present a distributed approach and a centralized approach for comparison of the results.

4. Distributed Approach

The assets, or the departments, in modern industries are each provided with their own computational entities in the form of an embedded micro-controller, or a processor which make it possible for us to represent that asset using what is called an 'agent' of the asset. The asset data reflects its working condition, on which the local computations are based. Thus, the corresponding agent is capable of independent decision-making based on the asset's data. An agent further linked to the agents of other assets to enable human-like interaction. Interaction here means the sharing of data, or the computational results at different levels.

The distributed job scheduling algorithm proceeds in the below listed steps:

1. The job demand reaches the PPC department.
2. PPC Department circulates the job demand across all machine agents.
3. Machine agent analyzes their respective past time-to-failures data and fits a reliability model (Weibull distribution, in the present case) to represent the health of each machine.
4. The machine agent uses heuristics to generate a set of optimal job sequences, which also include the PM-Jobs, according to the required objective.
5. Each job-sequence generated in the above step is assigned an index of feasibility for comparing with other sequences, generated by the same machine or other machines, at the enterprise-level. This index, or the Intensity Factor (IF) is explained in subsection 4.1.
6. List of job-sequences from each machine is sent back to the PPC department which evaluates them and prepares the enterprise-level job schedule.
7. PPC communicates these final schedules to all the machines for execution.

The above described steps are shown in the Figure 1. Computations happening at each level are explained under in the subsequent subsections.

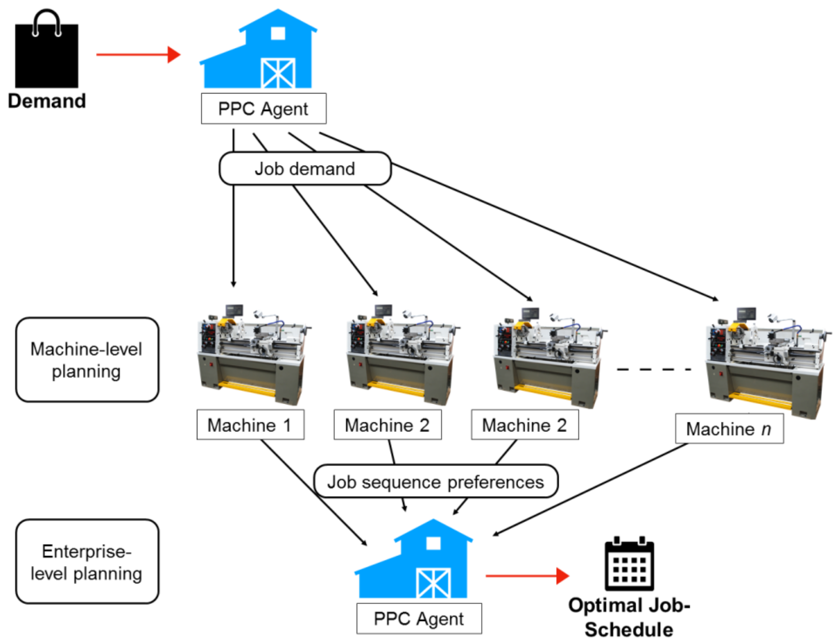

Figure 1. Schematic description of the data flowing within the Manufacturing environment

4.1 Machine Agent

The job demand first reaches the PPC. The PPC agent sorts the job list in increasing order of their processing times. This is done to reduce the computation load at machine level as it is a common step for the subsequent calculations at each machine agent level. This sorted job demand is then circulated across the machine agents.

The machine agent identifies a set of optimal sequences for its corresponding asset. It also includes PM job. Machine agent also evaluates the health of the machine in terms of probability of failure $F(t)$ using the Weibull parameters (η; β). We evaluate $F(t)$ for the machine after each job in the sequence, i.e. if a 1000 hours old machine sequence has the first job J3 with processing time 205 hours, we will calculate the machine's probability of failure after the processing time of J3 i.e. at the age 1205 hours. This $F(t)$ is evaluated using the Equation 1:

$$F(t) = 1 - e^{\left(-\frac{t}{\eta}\right)^{\beta}} \qquad (1)$$

If, while calculating $F(t)$, we find that it goes above a certain limit, we schedule a PM-Job. While this limiting value of the probability of failure can be adjusted according to the needs of the system, in our algorithm this limit is set at 0.5. The machine agent is responsible for evaluating the machine health after each demand job scheduled, and also scheduling the PM-Jobs.

'Optimal' sequence for our illustration refers to the sequence corresponding to the minimum cumulative delay/ lateness in production. Delay in production of a job is the difference between the due time and the time at which the job has been produced i.e. completion time (t_i^c). Cumulative delay is the sum of all the jobs in the demand received.

This cumulative delay of jobs is called 'lateness' corresponds to the sequence. The lateness in production of jobs when produced in a given sequence is termed as Intensity Factor (IF) of the sequence. IF is used first by the machine agent, and later by the PPC agent while preparing the enterprise-level job-schedule. Calculation of IF is explained in the following subsections.

It is solely to the manager to decide upon the algorithm which the machine agent would use to generate the set of optimal job sequences. The machine agent may resort to basic brute force algorithm for smaller problem sizes, or to other advanced heuristics and meta-heuristics for more complex problems. These may not lead to the best solution, but a near-optimal solution (Upasani et al., 2017).

IF is a common measure throughout the system. The objective in our algorithm is to minimize the lateness i.e. IF. The method of calculating the IF is shown below and the formula is given in the Equation 2:

$$IF = \sum_{i=1}^{n} t_i^c - \quad t_i^d \tag{2}$$

where, n is the total number of jobs in demand, and t_i^d is due time of the i^{th} job. The t_i^c is completion time of i^{th} job. It is sum of operations time (t_i^o) of the same job sequenced at k^{th} place and operation times of preceding jobs on the machine and estimated as:

$$t_i^c = \sum_{k=1}^{k} [t_i^o]_k \tag{3}$$

The operation time of i^{th} job sequenced on the machine at k^{th} place includes job processing time (t_i^{PT}), machine downtime due to PM (t_i^{pm}). It is calculated as:

$$t_i^o = t_i^{PT} + t_i^{pm} \tag{4}$$

An example describing the IF calculation for a sequence is shown below:

Consider a sequence of jobs out of a total demand of 9 jobs described in the Table 1:

J1, PM-Job, J4, J3, PM-Job, J6

The IF for this sequence will be calculated as:

Here, the due time for all jobs is same (see, table 1), and if a job manufactured before its due time, its lateness will be zero.

$$IF = \sum_{i=1}^{n} t_i^c - \quad t_i^d = [(55 - 260) + (168 - 260) + (373 - 260) + (566 - 260)] = 419\ hours$$

In summary, the machine agent has to perform the following tasks after receiving the demand job list from the PPC:

1. Evaluate time-to-failures distribution parameters.
2. Generate a list of the optimal job sequences which the machine can perform in its shift duration and based on its reliability characteristics.
3. Assign the PM-Jobs whenever necessary and calculate the IF for each sequence.
4. Assign the IF to the pool of job sequences generated in the task-1, and send these to the PPC agent.

4.2 PPC agent

When the PPC receives the job sequences from the machine agents it has to produce an enterprise-level job schedule. The PPC agent arranges all the job sequences in decreasing order of IFs (because lower the IF, more optimal is the corresponding sequence). The PPC then starts assigning the job sequences to each of the machines, starting with assigning the first sequence to the corresponding machine. Once this is done, the remaining sequences corresponding to this machine are deleted (because the machine is already assigned a job sequence). For similar reasons, the remaining sequences having jobs of this sequence are also deleted. The PPC then assigns the next sequence on the list to the corresponding machine, and the procedure keeps on repeating itself until all the jobs are assigned to the machines. This way, an enterprise-level job-schedule is prepared using inputs from machines considering PM requirements. The computations steps at the PPC level after receiving the job-preferences from the machines are shown in the Figure 2.

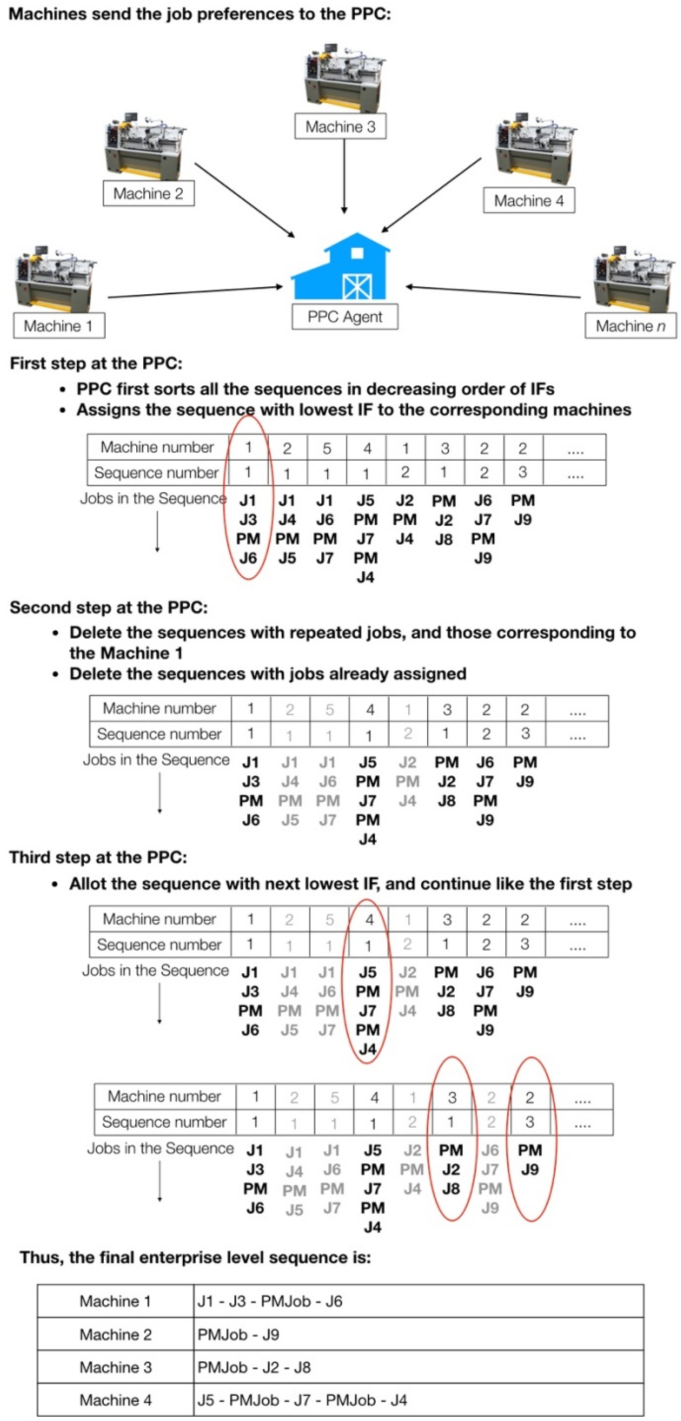

Figure 2. Computations at the PPC level after receiving the job-preferences from the machines

5. Centralized Approach

In the conventional centralized approach, the computations are done at a single level. There is only one entity to govern the operations involving several assets. For example, in our case the computations are done at the PPC level only. Although the operation of job scheduling involves several machines, the PPC solely generates the optimal job scheduling decisions for all the machines.

Conventionally, an enterprise-level scheduling algorithm is applied at the PPC level. This algorithm generates the job sequences for all the machines. In this approach, the PPC receives the job demand. Then, it generates the optimal job sequences for the machines such that lateness is minimized. Here, the lateness is calculated using Eq. (2); however, operation time calculation does not include PM time (t_i^{pm}). The new operation time (t_i^{co}) equation is estimated as:

$$t_i^{co} = t_i^{PT} \tag{5}$$

The maintenance decisions are superimposed on the optimal job sequences generated by the PPC based on the machine health calculated in terms of $F(t)$. $F(t)$ is calculated using Eq. (1) based on the reliability characteristics of machines after each job in the sequence. If the $F(t)$ value reaches to more than the limit i.e. 0.5, a PM-Job is assigned. The lateness of final schedule (considering PM -jobs) is calculated using Eq. (2).

6. Results

The betterment seen in terms of the reduced computation times and in terms of decreased lateness while using the above described distributed scheduling algorithm, is discussed here. For our example, we consider the job scheduling problem for different combinations of number of machines and the jobs to be scheduled, starting from scheduling 2 jobs into 2 machines till scheduling 9 jobs into 5 machines. All the cases (1-9) are shown in the table 3.

We use both- our distributed approach and conventional centralized approach to schedule the jobs in the above cases. The job scheduling problems are well-known and proved NP-hard (Kumar and Lad, 2016; Xie and Chen, 2018) and are of combinatorial non-linear optimization in nature (Kim et al., 2013). To solve such problems, simulation coupled with optimization is most widely used method (Sharma et al., 2011). Thus, the same is utilized here to solve the above cases. Jobs characteristics and machines properties are coded in Witness 14 simulation platform for all cases. For optimization, Adaptive Thermo-statistical Simulated Annealing (ATSA) and brute force (evaluate all combinations) techniques have been used. The Brute Force (BF) technique guarantees the optimality of the solution while ATSA provide near-optimal solution in less computation time. The results of all the cases in terms of computation time and lateness have been shown in table 3 and are summarized in figure 3.

Distributed and centralized approaches are evaluated for all the cases using ATSA technique. While using Brute Force (BF) technique, centralized approach is evaluated for smaller problem size cases (1-4); for larger problem size cases (5-9), it takes more than 24 hours of computation time which is impractical. Thus, cases (5-9) are not evaluated utilizing Centralized Brute Force (CBF) approach. The Distributed Brute Force (DBF) evaluates cases (1-8) and for last case (9), it takes more than 3 hours of computation time due to large solution space, thus is not evaluated.

Though the centralized Brute Force is expected to produce most optimal solution as it evaluates all the possible combination in the optimization problem, the time take for such approach is major limitation. In fact, it will be impractical to use brute force approach for most of the real life problems. The same can also be seen from figure 3 (left) where the time taken for CBF approach increases exponentially after case number 2 (problem size 2^7). However, the results for smaller problem size (up to case number 4) help us in getting an idea about the effectiveness of various approaches in terms of objective function (lateness). It can be seen from table 3 that, for smaller problem size distributed brute force provides optimal solution closer to the centralized brute force. Thus, in the absence of CBF results we may consider DBF results as the benchmark for further comparison. The centralized and distributed ATSA are slightly deviated from the most optimal solution for case number 2. From figure 3 we can see that distributed ATSA provides better solution (in terms of lateness) than that of the centralized ATSA. This is happening because the centralized approach is more of interrelated nature, as the maintenance schedule is superimposed on the final optimal production schedule. This is equivalent of saying that the production planning department is not aware of the machine health while making the production schedule and it superimposes the maintenance schedule received from the maintenance department. On contrary, the distributed approach gets the opportunity to evaluate the maintenance jobs for each of the possible solutions checked at machine level as it is aware about the health of the machine. Thus, distributed approach is more of integrated nature from the decision point of view. Further, advantages of distributed approach can be seen from the figure 3 (middle) which shows significantly lesser computation time taken by the distributed ATSA compared to centralized ATSA. Thus, distributed approach can be considered as a promising approach to solve complex shop floor scheduling problem.

Table 3: Comparison of results obtained from centralized and distributed approaches for different cases

Case No.	No. of Machines	No. of Jobs	All possible combinations	Computation time in seconds (ATSA)		Computation time in seconds (BF)		Lateness in hours (ATSA)		Lateness in hours (BF)	
				Centralized (CATSA)	Distributed (DATSA)	Centralized (CBF)	Distributed (DBF)	Centralized (CATSA)	Distributed (DATSA)	Centralized (CBF)	Distributed (DBF)
1	2	2	2^3	16	5	16	5	0	0	0	0
2	2	3	2^7	52	14	159	15	8	8	0	0
3	3	3	2^9	66	17	924	17	0	0	0	0
4	3	4	2^{13}	99	19	1937	72	0	0	0	0
5	3	5	2^{17}	179	22	NA	621	63	48	NA	28
6	4	4	2^{16}	167	20	NA	74	0	0	NA	0
7	4	5	2^{21}	193	23	NA	688	8	0	NA	0
8	5	6	2^{31}	226	26	NA	7367	8	8	NA	0
9	5	9	2^{49}	247	28	NA	NA	136	118	NA	NA

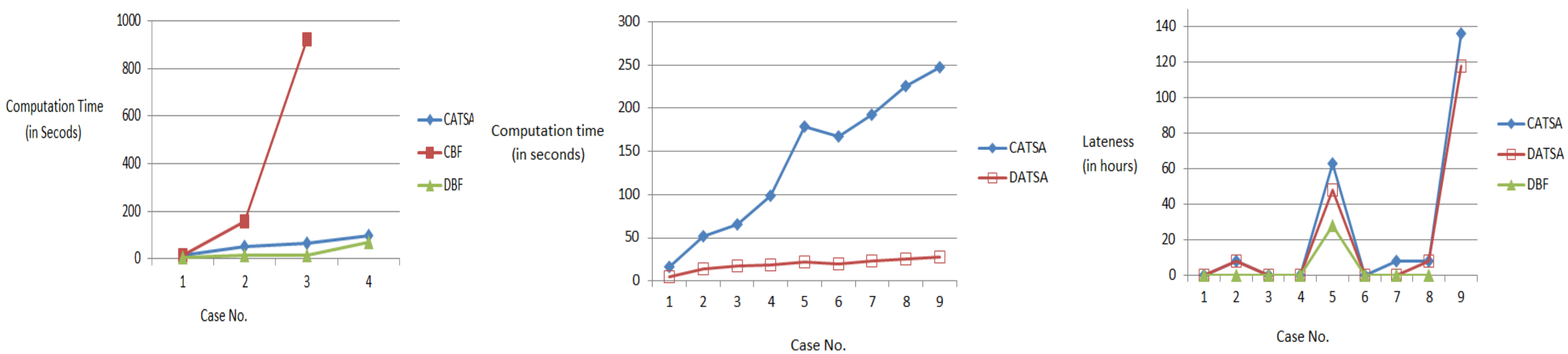

Figure 3. Comparison between centralized approach and distributed approach

7. Conclusion and Future Work

This paper proposes an innovative Multi-Agent System based distributed operations planning approach for scheduling of jobs in a parallel machine shop-floor. The approach is demonstrated for various cases. Each case is solved by the proposed distributed approach and conventional centralized approach to compare the results. Here, two different optimization techniques Adaptive Thermo-statistical Simulated Annealing (ATSA) and brute force are utilized for optimization. The proposed distributed approach utilizes cyber-physical-systems, parallel computing and communication capabilities of various agents (machine and PPC). This is one of the essential requirements of decision-making in smart factories under Industry 4.0. For complex problems, distributed approach provides best solution in least computation time. Therefore, distributed approach can be one of the alternatives for shop-floor operations planning for next generation manufacturing systems. In an event of failure, sudden surge in demand, or any other disturbance the system can go for rescheduling of jobs. This is made possible by the highly reduced computation time.

The future prospects are two-fold. First, to resemble the real-world manufacturing shop-floor more closely, the number of components of the machines can be increased and the processing times of the jobs can be different for different machines. The Maintenance Department can also know the quality of labor available and evaluate the variances in the PM-Jobs description, like the changes in restoration factors when using skilled or semi-skilled workforce. Secondly, the benefits of collaborative and distributed decision-making presented in this work can be carried forward to represent the entire shop-floor using agents, which collaborate with one another to govern several different operations. The agents can be used to represent the raw-materials, the consumers, the AGVs, etc.

Acknowledgement

The authors would like to thank The Royal Academy of Engineering, London (Newton Bhabha Project, HEPI\1516\10) for providing the necessary support for this research work.

References

Dilts, D. M., Boyd, N. P., and Whorms, H. H., The evolution of control architectures for automated manufacturing systems, *Journal of Manufacturing Systems*, vol. 10, no. 1, pp. 79-93, 1991.

Leitão, P., Agent-based distributed manufacturing control: A state-of-the-art survey, *Engineering Applications of Artificial Intelligence*, vol. 22, no. 7, pp. 979-991, 2009.

Hu, S. J., Evolving paradigms of manufacturing: From mass production to mass customization and personalization, *Procedia CIRP*, vol. 7, pp. 3-8, 2013.

Giret, A., and Botti, V., Holons and agents, *Journal of Intelligent Manufacturing*, vol. 15, no. 5, pp. 645-659, 2004.

Ganchev, I., Ji, Z., and O'Droma, M., Designing a low-cost data transfer unit for use in IoT applications, *8th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 85-88, 2016.

Unland, R., Industrial Agents, in *Industrial Agents*, pp. 23-44, Elsevier, 2015.

Lee, E. A., Cyber Physical Systems: Design Challenges, *11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pp. 363-369, 2008.

Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., and Eschert, T., Industrial Internet of Things and Cyber Manufacturing Systems, pp. 3-19. *Cham: Springer International Publishing*, 2017.

Annunziata, M., and Evans, P. C., Industrial Internet: Pushing the Boundaries of Minds and Machines, *General Electr*ic, 2012.

Da Xu, L., He, W., and Li, S., Internet of things in industries: A survey, *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233-2243, 2014.

Surowiecki, J., Wisdom of Crowds, *The Wisdom of Crowds*, Anchor, 2004.

Li, Hao, Adrià Salvador Palau, and Ajith Kumar Parlikad, A social network of collaborating industrial assets. Proceedings of the Institution of Mechanical Engineers, Part O: *Journal of Risk and Reliability,* vol. 232, no. 4, pp. 389-400, 2018.

Annunziata, M., Welcome to the age of the industrial internet, *TED Talks*, 2013.

Lasi, H., Fettke, P., Kemper, H. G., Feld, T., and Hoffmann, M., Industry 4.0, *Business and Information Systems Engineering*, vol. 6, no. 4, pp. 239-242, 2014.

Hall, N. G., and Potts, C. N., Rescheduling for New Orders, *Operations Research*, vol. 52, no. 3, pp. 440-453, 2004.

Vieira, G. E., Herrmann, J. W., and Lin, E., Rescheduling manufacturing systems: A framework of strategies, policies, and methods, *Journal of Scheduling*, vol. 6, no. 1, pp. 39-62, 2003.

Ovacik, I. M., and Uzsoy, R., Decomposition methods for complex factory scheduling problems, *Springer Science & Business Media*, 2012.

Stone, P., and Veloso, M., Multi-Agent systems: a survey from a machine learning perspective, *Autonomous Robots,* vol. 8, no. 3, pp. 345-383, 2000.

Duffie, N. A., and Piper, R. S., Non-hierarchical control of a flexible manufacturing cell, *Robotics and Computer-Integrated Manufacturing*, vol. 13, no. 2, pp. 175-179, 1987.

Giordani, S., Lujak, M., and Martinelli, F., A distributed multi-agent production planning and scheduling framework for mobile robots, *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 19-30, 2013.

Mönch, L., and Drießel, R., A distributed shifting bottleneck heuristic for complex job shops, *Computers and Industrial Engineering*, vol. 49, no. 3, pp. 363-380, 2005.

Lou, P., Liu, Q., Zhou, Z., Wang, H., and Sun, S. X., Multi-agent-based proactive-reactive scheduling for a job shop, *International Journal of Advanced Manufacturing Technology*, vol. 59, no. (1-4), pp. 311-324, 2012.

Upasani, K., Bakshi, M., Pandhare, V., and Lad, B. K., Distributed maintenance planning in manufacturing industries, *Computers and Industrial Engineering*, vol. 108, pp. 1-14, 2017.

Martin, S., Ouelhadj, D., Beullens, P., Ozcan, E., Juan, A. A., and Burke, E. K., A multi-agent based cooperative approach to scheduling and routing, *European Journal of Operational Research*, vol. 254, no. 1, pp. 169-178, 2016.

Sahin, C., Demirtas, M., Erol, R., Baykasoğlu, A., and Kaplanoğlu, V., A multi-agent based approach to dynamic scheduling with flexible processing capabilities, *Journal of Intelligent Manufacturing*, vol. 28, no. 8, pp. 1827-1845, 2017.

Xiong, W., and Fu, D., A new immune multi-agent system for the flexible job shop scheduling problem, *Journal of Intelligent Manufacturing*, vol. 29, no. 4, pp. 857-873, 2018.

Christensen, J. H., HMS/FB Architecture and its Implementation, Agent Based Manufacturing: Advances in the Holonic Approach*, Springer-Verlag Berlin,* Heidelberg, 2003.

Bagheri, B., Yang, S., Kao, H. A., and Lee, J., Cyber-physical systems architecture for self-aware machines in industry 4.0 environment, *IFAC-Papers OnLine*, vol. 48, no. 3, pp. 1622-1627, 2015.

Bakliwal, K., Dhada, M. H., Palau, A. S., Parlikad, A. K., and Lad, B. K., A Multi Agent System architecture to implement Collaborative Learning for social industrial assets, in 16th *IFAC Symposium on Information Control Problems in Manufacturing*, vol. 51, no. 11, pp. 1237-124, 2018.

Palau, A. S., Bakliwal, K., Dhada, M. H., Pearce, T., and Parlikad, A. K., Recurrent Neural Networks for real-time distributed collaborative prognostics, *IEEE Conference on Prognostics and Health Management (ICPHM),* pp. 1-8, 2018.

Kijima, M., Some Results for Repairable Systems with General Repair, *Journal of Applied Probability*, vol. 26, no. 1, pp. 89-102, 1989.

Kumar, S., and Lad, B. K., Integrated production and maintenance planning for parallel machine system considering cost of rejection, Journal of the Operational Research Society, vol. 68, no. 7, pp. 834-846, 2016.

Xie, N., and Chen, N., Flexible job shop scheduling problem with interval grey processing time, *Applied Soft Computing*, vol. 70, pp. 513-524, 2018.

Kim, B. S., and Ozturkoglu, Y., Scheduling a single machine with multiple preventive maintenance activities and position-based deteriorations using genetic algorithms, The International Journal of Advanced Manufacturing Technology, vol. 67, no. (5-8), pp. 1127-1137, 2013.

Sharma, A., Yadava, G. S., and Deshmukh, S. G., A literature review and future perspectives on maintenance optimization, Journal of Quality in Maintenance Engineering, vol. 17, no. 1, pp. 5-25, 2011.

Witness 14 Manufacturing Performance Edition, Lanner Group, Accessed August 18 2018. https://www.lanner.com/insights/blog/witness-14-has-arrived.html.

Biographies

Sandeep Kumar received the Ph.D. degree in the area of industrial engineering from the Department of Mechanical Engineering, Indian Institute of Technology (IIT) Indore, Indore, India, in 2018. He received the Master's degree in maintenance engineering from the National Institute of Technology Bhopal, India, in 2013. He is currently working as postdoctoral fellow at the Industrial and Systems Engineering in IIT Indore, India. His research focuses on agent based decision-support system to upgrade the today's factories to smart factories by utilizing the advanced data analytics tool, CPS, and IIoT. His major research interests are in the areas of smart manufacturing, CPS, IIoT, simulation modeling and optimization, reliability engineering, maintenance scheduling, and operation planning.

Maharshi Harshadbhai Dhada is a PhD student at the University of Cambridge, working at the Institute for Manufacturing (IfM). His research is focused on applying machine learning algorithms for distributed decision making in the manufacturing systems. Before joining IfM, he had worked under the supervision of Dr. Bhupesh K. Lad at the Industrial and Systems Engineering in IIT Indore as an undergraduate researcher. At IIT Indore he has developed algorithms for distributed operations planning, and prognostics by analyzing the time-series data from the assets. Apart from publishing papers in leading international conferences like INCOM, and IEEE PHM he has also participated in various national level exhibitions. He received his undergraduate degree in Mechanical Engineering from the Department of Mechanical Engineering, IIT Indore, India.

Kshitij Bakliwal is pursuing a Master's degree in Computer Science from Purdue University. He has recently completed his Bachelor's in Electrical Engineering from IIT Indore. He has interned with SAP Labs, McDonald's Global Digital Team, and the Institute of Manufacturing (IfM) at University of Cambridge. Kshitij joined the Industrial and Systems Engineering at IIT Indore to be a part of the team working on advanced optimization problems in Manufacturing, and to explore and learn about the Industry 4.0 paradigm. While under Dr. Bhupesh K. Lad as an undergraduate researcher, he enjoyed researching on and finding solutions for Prognostics and Health Management of Smart Machines. The Distributed Job-Scheduling problem allowed him to work, along with his colleagues, on Directed Acyclic Graphs and Digital Twins. He received his undergraduate degree in Electrical Engineering from the Department of Electrical Engineering, IIT Indore, India.

Bhupesh Kumar Lad received the Ph.D. degree in the area of reliability engineering from the Department of Mechanical Engineering, IIT Delhi, New Delhi, India, in 2010. He worked with GE Global Research Centre, Bangalore, India, as a Research Engineer from 2010 to 2011. He is currently an Associate Professor in the discipline of Mechanical Engineering in IIT Indore, India. He is the author of the book Machine Tool Reliability (Hoboken, NJ, USA and Salem, MA, USA: Wiley and Scrivener, 2016). He is an investigator of various research projects funded by national and international funding agencies. His major research interests include smart manufacturing, reliability engineering, and prognostics. Dr. Lad received the Hamied-Cambridge Visiting Lecture Fellowship of the University of Cambridge, U.K., in 2016.