Decentralized Scheduling for Flexible Job Shop with Sequence-Dependent Setup Time

Manojkumar R. Pal, Murari L. Mittal, Gunjan Soni
Department of Mechanical Engineering
Malaviya National Institute of Technology (MNIT), Jaipur, India
2017rme9086@mnit.ac.in, mlmittal.mech@mnit.ac.in, gunjansoni.mech@mnit.ac.in

Satyendra S. Chouhan
Department of Computer Science Engineering
Malaviya National Institute of Technology (MNIT), Jaipur, India
sschouhan.cse@mnit.ac.in

Abstract

Scheduling is a persistent problem in manufacturing systems and is considered one of the most challenging combinatorial problems. The scheduling problem of flexible job shops where operations are scheduled on the available flexible machines is commonly referred to as FJSP and is considered as an important scheduling problem due to its nearness to advanced manufacturing systems. During production, the achines need setup for processing two different jobs consecutively. Such setup time is called sequence-dependent setup time (SDST). This paper presents a decentralized multi-agent-based model for the FJSP-SDST. For optimization, a popular swarm-based algorithm, particle swarm optimization (PSO), is embedded with the multi-agent model to achieve a global solution. An illustrative example has been explained along with the computational results for the ten standard test cases to show the effectiveness and robustness of the approach.

Keywords
Flexible job shop scheduling problem, Sequence-dependent setup time, particle swarm optimization, decentralized scheduling, multi-agent system and scheduling.

1. Introduction
A flexible job shop is a manufacturing system where a variety of jobs can be processed by flexible machines. Scheduling problems, considered in the flexible job shop, are often known as the Flexible Job-shop Scheduling Problem (FJSP). FJSP has attracted the attention of researchers due to its closeness to real-world manufacturing systems (Mokhtari et al., 2015). For solving FJSP, two sub-problems, machine assignment and operation sequencing, are generally solved. Along with these, investigations are also carried out to solve FJSP with various constraints. maintenance planning, transportation and setup time(Gao et al. 2006; Li et al. 2020, 2013; Li and Gao 2016; Nouri et al. 2016; Wang and Yu 2010; Hu et al. 2020). The setup time is needed to make a machine or resource ready for the processing of the job. However, the setup time soften dependent on the previous job processed on the machine. Such setup time is called as sequence-dependent setup time (SDST).

For solving the FJSP-SDST, two solution approaches, exact and approximate algorithms, have been applied. The exact algorithms, however, are not suitable for large real-life problems owing to the NP-Hard nature of FJSP. Thus, the approximate algorithms have been applied more often for the FJSP-SDST. These approaches are hybrid Genetic Algorithm (GA) (Wang and Zhu 2021), Imperialist Competitive Algorithm (ICA)(Li and Lei 2021a), self-adaptive evolutionary algorithm(Azzouz, Ennigrou and Said 2017)and Tabu Search (TS) (Abdelmaguid 2015), Iterated local search (ILS) (Mousakhani 2013). The above algorithms have been applied with a centralized decision-making approach. The centralized decision-making approach works on controlling decisions centrally, limiting the flexibility and scalability to solve complex combinatorial problems such as FJSP. Instead, decentralized approaches, such as multiagent systems (MAS), are more suitable for solving such issues due to their ability to simplify complex problems through disintegration(Barbati et al. 2012; Zhang et al. 2019). The agents in MAS are allocated different tasks which cooperate/compete with other agents to fulfil their objectives. In literature, different MAS have been applied to solve scheduling problems for FJSP(Ennigrou and Ghédira, 2008; Henchiri and Ennigrou, 2013; Nouri, 2018, Nouri et al. 2016). But, no decentralized approach has been applied to solve FJSP-SDST. As per author's knowledge, this paper is the first to develop decentralized multi agent system for the FJSP-SDST. The results of make span achieved by the proposed algorithm are compared with the other published algorithms. The objectives of the paper are:

- To develop a multiagent system for the FJSP-SDST
- To validate the model

The remainder of this paper is organized as follows. Section 2 presents the literature review. The problem definition and the proposed model are explained in section 3. The proposed approach is presented in section 4. Section 5 discusses the proposed decentralized algorithm's performance on a set of benchmarks problems; at last, section 6 concludes the results.

2. Literature Review

FJSP has received the attention of researchers due to its closeness to the current advanced manufacturing systems. Brucker and Schlie (1990) pioneered the research on the FJSP by proposing a tabu search (TS) based heuristic. After that, various algorithms were developed to solve the different variants of FJSP. Most of the approaches are approximate algorithms due to the NP-hard nature of the FJSP. The approximate algorithms generally consist of evolutionary algorithms and heuristics, e.g. particle swarm optimization (PSO), ant colony optimization (ACO), grey wolf algorithm (GWO), genetic algorithm (GA), tabu search (TS), simulated annealing (SA).

To consider the real-life scheduling issues, the FJSP has been extended with various constraints. Sequence-dependent setup time (SDST), one of the essential constraints, depends on the sequence of the predecessor job and the upcoming job of the possibly allocated machine. It has been known that the FJSP-SDST was first considered by Choi and Choi (2002), who proposed a local search algorithm to minimize the make span. Guimaraes and Fernandes (2006) proposed GA operators for solving the FJSP-SDST to minimize the make span, total workload of the machines, maximum workload, and total setup time. Mousakhani (2013) proposed a metaheuristic based on iterated local search with insertion neighborhood to solve FJSP-SDST and minimize total tardiness. Abdelmaguid (2015) presented a multiple integer linear programming model (MILP) with a neighborhood search function, Ennigrou and Said (2017) developed a hybrid algorithm based on GA+ iterated local search (ILS) and Luo and Lin (2020) proposed selection hyper-heuristics (SHH) to solve FJSP-SDST to minimize make span, .Li and Lei (2021) proposed the imperialist competitive algorithm (ICA) to solve FJSP with transportation and SDST.

The literature review on the FJSP-SDS Travels that most approaches are based on centralized decision-making, which has limitations in terms of flexibility and expansibility. The decentralized approaches are becoming more popular in solving such complex scheduling problems. The multi-agent system, a decentralized approach, generally disintegrates the system's decision-making from a singular entity into multiple entities (called agents) (Barbati et al. 2012).These multiple agents handle smaller sub-problems of the main problem in groups to fulfil the objective(s) of the problem. There have been many applications of the multi-agent system to solve the FJSPs, such as multi-agent tabu search local optimization (MATSLO) by Ennigrou and Ghédira (2008), multi-agent model with TS and PSO (MATSPSO) by Henchiri and Ennigrou (2013), holonic multi-agent model with GA and TS (HMA-GATS) by Nouri et al. (2016).To the best of the knowledge of the authors, multi-agent systems have not been applied to solve FJSP-SDST. This paper presents a multi-agent model to solve FJSP-SDST.

3. Problem Definition

This section explains the mathematical formulation of the FJSP-SDST. A set of *n* jobs is considered with a set of *m* machines. Top recess job *i*, a number of operations (r_i) have to be carried out in a predefined sequence.$C_i$is the completion time of the job after completion of all its operations. Setup time is needed to make the machine ready to process the next job. The setup time, dependent on the preceding job, is called sequence-dependent setup time (SDST). If the subsequent operations are of the same job, the SDST is zero; otherwise, $S_{i,d,k}$ (setup time where two different job s*i* and *d* are performed on machine *k*).The objective is to minimize the make span (maximum completion time of the machines) defined as:

$$\text{Minimize } Z = \sum_{i=1}^{n} C_i$$

4. The proposed model

This paper proposes multi-agent based model to solve the FJSP-SDST. The decentralized decision-making model is implemented by using the auction-based strategy to allocate the job operations to machines(Saad et al. 1997; Xia and Wu 2005).The remainder of this section describes the proposed MAS.

4.1 Multi-agent Architecture

The proposed multi-agent model creates three different classes of agents: the coordinator, job, and resource (Figure 1). These agents are actively involved in finding the allocation of the machines to the sequence of operations. For the optimization, the coordinator agent employs a popular swarm-based algorithm known as Particle Swarm Optimization (PSO).

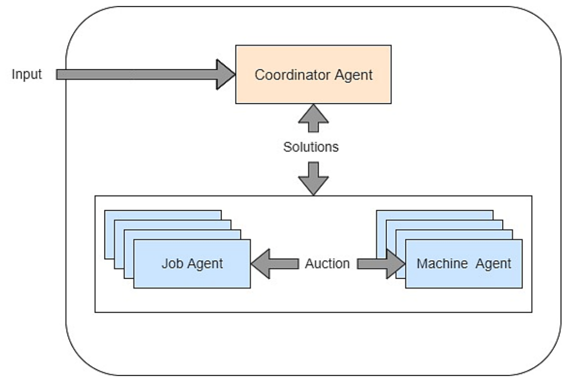

Figure 1.The architecture of the multi-agent model for the FJSP-SDST

A coordinator agent is a class of an individual and coordinating agent responsible for initiating the process and optimizing the solution through metaheuristics. The randomly generated OS vectors by the coordinator agent are sent to job agents for initiating the auction between job agents and machine agents for machine selection. The job agent chooses the best machine agent with the minimum completion time-based on the bids received from the available machine agents. Similarly, each OS vector is solved for its length of operations. After allocation of all OS vectors, job agent PSO for optimization.

4.2Particle Swarm optimization

Particle swarm optimization (PSO), developed by Kennedy and Eberhart (1995), is a popular bio-inspired algorithm inspired by the characteristics of swarm creatures like a group of birds. In PSO, a potential solution called a particle solution moves towards the best solution with its velocity, along with the self-best and global best. The position of the particle solution is updated as per the following equation (1) and (2).

$$v_h^{t+1} = wv_h^t + c_1r_1(p\_best_h^t - x_h^t) + c_2r_2(g\_best_h^t - x_h^t) \tag{1}$$
$$x_h^{t+1} = x_h^t + v_h^{t+1} \tag{2}$$

where,v_h^{t+1} and $x_h^t$are velocity and current position of the particle. w , c_1&$c_2$are inertia weight, cognition factor, and social factor. r_1&$r_2$are two random numbers.t is the present operation.$p\_best_h^t$&$g\_best_h^t$ are represents self-best and global-best position. For making PSO suitable to the discrete natured scheduling problems, discrete operators are used as per the following manipulated equation (3)(Huang et al. 2016).

$$x_h^{t+1} = w \otimes f_1(x_h^t) + c_1 \otimes f_2(x_h^t, p\_best_h^t) + c_2 \otimes f_2(x_h^t, g\_best_h^t) \tag{3}$$

where$\otimes$ is a probability operation which defines that the following operator is applied with the corresponding probability. "+" explains the implementation of the following operator.w , c_1&c_2 are the predefined probabilities, 0.8,0.5 and 0.5, respectively.f_1&f_2 are two discrete operators wheref_1 is applied through swapping of two random positions of the particle and f_2 is applied by crossing the particle with their self-best position and global-best position. The precedence-based crossover (POX) can be referred from Zhang et al. (2020).

To illustrate the approach, an example is considered with details of processing time and SDST as given in Table 1and Table 2. Dummy time represents a machine's initial setup time before processing a job's first operation. The proposed approach is applied to solve the example, e.g. Job 3 asks to auction for its first operation (denoted "31"). Each machine agent offers a bid for the operation. From the available bids, the third machine agent has been chosen for its lowest bid. Similarly, other operations have been auctioned and scheduled. The best make span value of 25 is achieved, as shown in the Gantt chart (Figure 2). In Figure 3, machines are defined on the ordinate by "M/C", and the time horizon is represented on the abscissa. The red block defines the SDST where the symbol "D" represents dummy SDST, "S" represents SDST and other color blocks define the processing of the job operations.

Table 1. The processing time for the operation

Operation	M_1	M_2	M_3
O_{11}	6	10	-
O_{12}	5	8	5
O_{13}	4	-	3
O_{21}	-	6	-
O_{22}	5	7	-
O_{23}	9	5	5
O_{31}	4	-	3
O_{32}	5	6	8
O_{33}	5	4	9
O_{34}	-	6	8

Table 2. The sequence-dependent setup time details

To	M_1			M_2			M_3		
From	Job1	Job2	Job3	Job1	Job2	Job3	Job1	Job2	Job3
Dummy	2	1	4	4	3	3	3	1	2
Job1	0	2	1	0	2	4	0	4	2
Job2	3	0	4	2	0	1	3	0	4
Job3	2	2	0	3	2	0	1	4	0

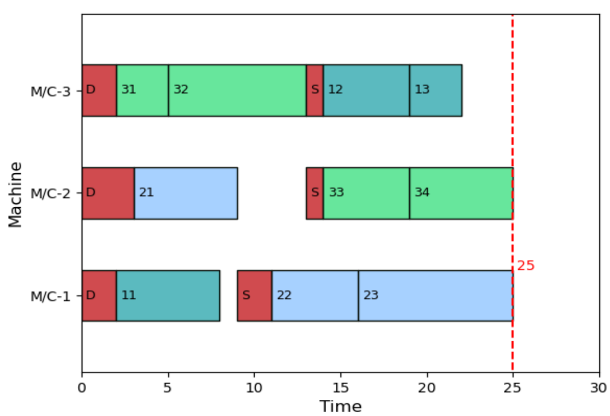

Figure 2. The Gantt chart for the best solution of the Example

5. Experimental Evaluation

This section discusses the comparison of the results achieved by the proposed algorithm with some of the existing approaches. The proposed algorithm is coded in python programming (3.7.7) and runs on an Intel Core i5-8250U CPU at 3.6 GHz with an 8 GB RAM machine. Ten test instances are generated from Bagheri and Zandieh (2011) with details of a number of jobs (n), a number of operations (n_i) and a number of machines (m)as per Table 3 and table 4. Population size and the number of generations are the parameters considered 50 and 50, respectively. Each problem instance is run by ten independent trials to get the best makespan. The results of the proposed algorithm are compared with benchmark algorithms HGA(Azzouz, Ennigrou and Ben Said 2017) and GA, SAHA(Azzouz, Ennigrou and Said 2017).

Table 3.The details of the three instance characteristics

	n x n_ix m	Average Available machine	Processing time	SDST	Dummy Jobs
Class	10x5x5	U(1,5)	U(20,100)	U(20,60)	U(20,40)

Table 4. The comparison of makespan results of the algorithms

Instance Problem	Flexibility	Size	GA	HGA	SAHA	MAS-PSO
BZ01	2.64	10 x 5 x 5	1082	865	846	813
BZ02	3.22		981	818	807	778
BZ03	3.00		886	762	753	773
BZ04	3.26		917	840	807	796
BZ05	2.92		961	898	902	900
BZ06	4.69	15 x 8 x 5	959	870	831	746
BZ07	4.55		884	896	820	769
BZ08	4.12		927	886	910	834
BZ09	4.42		871	870	818	758
BZ10	4.56		893	875	825	770

It is visible from the results that the performance of the proposed multi-agent models superior to the GA algorithm (Figure 3). Compared to the SAHA and HGA approach, our decentralized approach outperforms almost all instances except BZ03, as shown in Figure 3. It is also seen in the results that the proposed approach performs better for large problems. In summary, the proposed approach has proved its effective performance against other algorithms in terms of overall results.

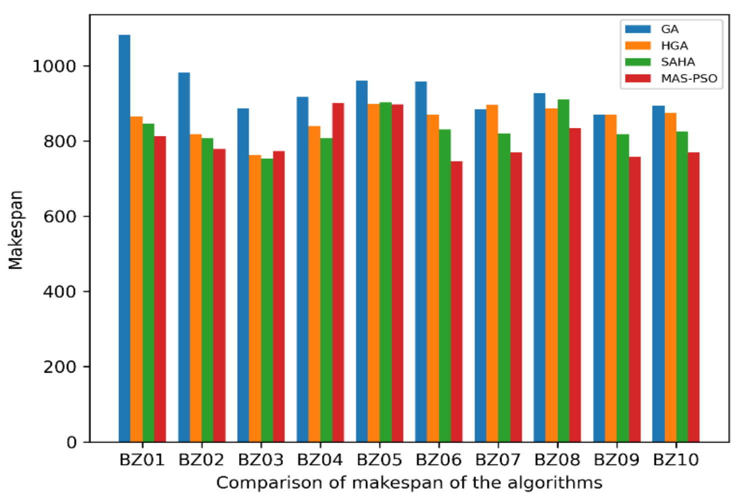

Figure 3. The comparison of the makespan of the algorithms

6. Conclusion

This paper has addressed the FJSP with sequence dependent setup time (SDST) with an objective to minimize the makespan. The decentralized multi agent-based model is presented to solve the FJSP-SDST. The approach consists of three agents: coordinator agent, job agents and machine agents. These agents are employed with auction strategy to solve the two sub-problems of operation sequencing and machine selection along with fulfilling the SDST constraint. To perform global search, the multi-agent system is embedded with optimization algorithm Particle swarm optimization (PSO) to achieve minimization of Makespan. An illustrative example is solved. The performance of the algorithm is compared with benchmark algorithms by solving ten standard benchmark instances. The results are competitive for small problems and are much better for large problems. This approach shows the effectiveness of decentralized approach to solve complex problems of FJSP. In future works, the proposed decentralized algorithm will be applied on the other standard benchmark instances for testing the large problem size with multiple objectives and constraints.

References

Abdelmaguid, T.F., A neighborhood search function for flexible job shop scheduling with separable sequence-dependent setup times, *Applied Mathematics and Computation*, Elsevier Ltd., vol. 260, pp. 188–203, 2015.

Azzouz, A., Ennigrou, M. and Said, L. Ben, A self-adaptive hybrid algorithm for solving flexible job-shop problem with sequence dependent setup time, *Procedia Computer Science*, Vol. 112, IEEE, pp. 457–466, 2012.

Azzouz, A., Ennigrou, M. and Ben Said, L., A hybrid algorithm for flexible job-shop scheduling problem with setup times, *International Journal of Production Management and Engineering*, vol. 5, no. 1, p. 23, 2017.

Bagheri, A. and Zandieh, M., Bi-criteria flexible job-shop scheduling with sequence-dependent setup times - Variable neighborhood search approach, *Journal of Manufacturing Systems*, The Society of Manufacturing Engineers, vol. 30, no. 1, pp. 8–15, 2011.

Barbati, M., Bruno, G. and Genovese, A., Applications of agent-based models for optimization problems: A literature review, *Expert Systems with Applications*, Elsevier Ltd, vol. 39, no. 5, pp. 6020–6028, 2012.

Brucker, P. and Schlie, R., Job-shop scheduling with multi-purpose machines, *Computing*, vol. 45, no. 4, pp. 369–375, 1990.

Choi, I. and Choi, D., A local search algorithm for jobshop scheduling problems with alternative operations and sequence-dependent setups, vol. 42, 2002.

Ennigrou, M. and Ghédira, K., New local diversification techniques for flexible job shop scheduling problem with a multi-agent approach, *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 2, pp. 270–287, 2008.

Ennigrou, M. and Said, L. Ben, ScienceDirect A self-adaptive self-adaptive hybrid hybrid algorithm algorithm for for solving solving flexible flexible job-shop problem with sequence dependent setup time problem with sequence dependent setup time, 2017.

Gao, J., Gen, M. and Sun, L., Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm, *Journal of Intelligent Manufacturing*, vol. 17, no. 4, pp. 493–507, 2006.

Guimaraes, K.F. and Fernandes, M.A., An Approach for Flexible Job-Shop Scheduling with Separable Sequence-Dependent Setup Time, pp. 3727–3731, 2006.

Henchiri, A. and Ennigrou, M., Particle swarm optimization combined with tabu search in a multi-agent model for flexible job shop problem, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7929 LNCS, no. PART 2, pp. 385–394, 2013.

Huang, S., Tian, N., Wang, Y. and Ji, Z., Solving flexible job shop scheduling problem using a discrete particle swarm optimization with iterated local search, *Communications in Computer and Information Science*, vol. 643, pp. 603–612, 2016.

Li, J. Qing, Deng, J. wen, Li, C. you, Han, Y. yan, Tian, J., Zhang, B. and Wang, C. gang, An improved Jaya algorithm for solving the flexible job shop scheduling problem with transportation and setup times, *Knowledge-Based Systems*, Elsevier B.V., vol. 200, p. 106032, 2020.

Li, J.Q., Pan, Q.K. and Tasgetiren, M.F., A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities, *Applied Mathematical Modelling*, Elsevier Inc., vol. 38, no. 3, pp. 1111–1132, 2013.

Li, M. and Lei, D., Engineering Applications of Artificial Intelligence An imperialist competitive algorithm with feedback for energy-efficient flexible job shop scheduling with transportation and sequence-dependent setup times, vol. 103, no. October 2020, 2021a.

Li, M. and Lei, D., Engineering Applications of Artificial Intelligence An imperialist competitive algorithm with feedback for energy-efficient flexible job shop scheduling with transportation and sequence-dependent setup times, *Engineering Applications of Artificial Intelligence*, Elsevier Ltd, vol. 103, no. May 2020, p. 104307, 2021b.

Li, X. and Gao, L., An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem, *International Journal of Production Economics*, Elsevier, vol. 174, pp. 93–110, 2016.

Luo, M. and Lin, J., Solving Flexible Job-shop Problem with Sequence-dependent Setup Times by Using Selection Hyper-heuristics, pp. 428–433, 2020.

Mousakhani, M., Sequence-dependent setup time flexible job shop scheduling problem to minimise total tardiness, vol. 7543, 2013, available at:https://doi.org/10.1080/00207543.2012.746480.

Nouri, H.E., Solving the flexible job shop problem by hybrid metaheuristics-based multiagent model, pp. 1–14, 2018.

Nouri, H.E., Belkahla Driss, O. and Ghédira, K., Simultaneous scheduling of machines and transport robots in flexible job shop environment using hybrid metaheuristics based on clustered holonic multiagent model, *Computers and Industrial Engineering*, Elsevier Ltd, vol. 102, pp. 488–501, 2016.

Saad, A., Kawamura, K. and Biswas, G., Performance Evaluation of Contract Net-Based Heterarchical Scheduling for Flexible Manufacturing Systems, *Intelligent Automation and Soft Computing*, vol. 3, no. 3, pp. 229–247, 1997.

Wang, S. and Yu, J., An effective heuristic for a flexible job-shop scheduling problem with maintenance activities, *Computers and Industrial Engineering*, Elsevier Ltd, vol. 59, no. 3, pp. 436–447, 2010.

Wu, Z. and Weng, M.X., Multiagent scheduling method with earliness and tardiness objectives in flexible job shops, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 2, pp. 293–301, 2005.

Xia, W. and Wu, Z., An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, *Computers and Industrial Engineering*, vol. 48, no. 2, pp. 409–425, 2005.

Zhang, G., Hu, Y., Sun, J. and Zhang, W., An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints, *Swarm and Evolutionary Computation*, Elsevier Ltd, vol. 54, no. February, p. 100664, 2020.

Zhang, J., Ding, G., Zou, Y., Qin, S. and Fu, J., Review of job shop scheduling research and its new perspectives under Industry 4.0, *Journal of Intelligent Manufacturing*, Springer US, vol. 30, no. 4, pp. 1809–1830, 2019.

Zhang, Y., Zhu, H. and Tang, D., An improved hybrid particle swarm optimization for multi-objective flexible job-shop scheduling problem, *Kybernetes*, vol. 49, no. 12, pp. 2873–2892, 2020.

Biographies

Manojkumar Pal is currently a PhD student at MNIT JAIPUR. He received a master's degree in Industrial engineering from MNIT JAIPUR in 2013. His research interests are Smart Manufacturing, Industry 4.0, Multi-agent systems, Scheduling, Optimization, Artificial Intelligence, and Machine learning.

Dr Murari Lal Mittalis a Professor in the Department of Mechanical Engineering, Malaviya National Institute of Technology Jaipur (India). His research areas include Production and Operations Management, Project Management and Scheduling, Multiagent Systems, and Industry 4.0. He has authored papers in various reputed journals such as Engineering Application of Artificial Intelligence, Computers and Industrial Engineering, International Journal of Production Economics, International Journal of Production Research, Production Planning and Control, International Journal of Advanced Manufacturing Technology, International Journal of Lean Six Sigma, Measuring Business Excellence, International Journal of Operational Research, International Journal of Advanced Operations Management.

Dr Gunjan Soni did his B.E. from the University of Rajasthan, M.Tech from IIT-Delhi and PhD. from Birla Institute of Technology and Science, Pilani, in 2012. He works as an Assistant Professor at Malaviya National Institute of Technology, Jaipur, Rajasthan, India. He has over 15 years of teaching experience at undergraduate and graduate levels. His areas of research interest are Predictive maintenance and Digital technology applications in supply chain management. He has published more than 80 papers in peer-reviewed journals, including IEEE Transactions on Engineering Management, Production Planning and Control, Annals of Operations Research, Computers and Industrial Engineering etc.

Satyendra Singh Chouhan received a PhD in computer science and engineering from the Indian Institute of Technology (IIT), Roorkee, India. Currently, he works as an Assistant Professor with Malaviya National Institute of Technology (MNIT), Jaipur, India. He has authored more than 12 research papers in international conferences/journals and book chapters. His research interests are AI planning, machine learning, and deep learning for natural language processing.