Investigation of reconfiguration effect on makespan with social network analysis for flexible job shop scheduling problem

M. B. S Sreekara Reddy[a], C.h Ratnam[a], Rajeev Agrawal[b], M. L. R. Varela[c], Ila Sharma[d] , and V. K. Manupati[d*]

[a]Department of Mechanical Engineering, AUCE, Visakha Patnam, AP, India.

[b]Department of Production Engineering, Birla Institute of Technology, Ranchi, India.

[c]Department of Production and Systems, School of Engineering, University of Minho

[d]School of Mechanical Engineering, VIT University, Vellore, Tamil Nadu, INDIA.

Mobile: +91 9775627564

Email: manupativijay@gmail.com,

mbssreddy@kluniversity.in,f.chan@polyu.edu.hk,chratnam@gmail.com

* Corresponding author: manupativijay@gmail.com

Abstract

This paper presents a novel social network analysis based method (SNAM) to evaluate the reconfiguration effect i.e., identification of key machines and their influence on the system performance in the context of Flexible job shop scheduling problem (FJSSP). This research formulates a mathematical model along with the constraints by incorporating the total completion time of jobs as an objective function. The proposed SNAM has been applied to generate the collaboration networks by transforming the input data and presenting them in the form of an affiliation matrix to the network analysis software. Thereafter, to analyse the collaboration networks various SNA measures that have been calculated and different functional properties are evaluated. Finally, to investigate the reconfiguration effect on makespan integration of process planning and scheduling (IPPS) has been implemented with adopted effective game theory based hybrid deoxyribonucleic acid (DNA) algorithm. The validation of the proposed approach and its effectiveness is conducted through comparisons with benchmark instances and results confirm the efficiency of the proposed approach.

Keywords: Flexible job shop scheduling problem; Social network analysis method; Integration of process planning and scheduling; Deoxyribonucleic acid.

1. Introduction

The current manufacturing sector needs to change its trend and scale new heights by inculcating the recent advancements in technology that has created the need for customized products amongst customers leading to high degree of competition between industries. To cater the mentioned requirements, particularly in a shop floor scenario, it is crucial to choose efficacious manufacturing strategies that can schedule effectively and efficiently to enhance the firm's productivity. In recent years, researchers are trying to achieve an optimum as well as a feasible schedule, which can able to perform an operation on more than one machine out of set of machines.

In our paper, we are focussing on Flexible Job Shop Scheduling Problem (FJSSP), which is an extension of the classical JSSP. The FJSSP, as the name suggests, is able to adopt a more flexible setup where it is feasible to run an operation on more than one machine. Hence, in FJSSP, it is very difficult to identify the allocation of operations on a particular machine out of a set of competent machines. Additionally, monitoring the sequence of operations in order to minimize the makespan is a cumbersome task which makes it more arduous than JSSP. From the literature, job shop scheduling problem has been validated as a NP-hard problem Garey and Sethi, (1976) and it can be concluded that FJSSP is a NP-hard combinatorial optimization problem.

In general, FJSSPs have been solved by two different approaches: Hierarchical and Integrated approaches Zhang et al., (2011). In Hierarchical or the traditional approach, process planning and scheduling has been regulated in a successive manner. In other words, to solve the problem with hierarchical approach, divide the problem into sub-problems and then solve it individually Brandimarte (1993). The greatest risk associated with this approach is the chance of losing out on high quality favourable solutions and settling with a compromising solution for both objectives. Saygin and Kilc (1999) mentioned various difficulties with traditional/ hierarchical approach and the related obstacles to be settled in order to improve the productivity of the system. Numerous approaches for solving the FJSSP were proposed by Hurink et al., (1994); and Najid et al., (2002). Yang (2001) introduced the multistage genetic algorithm to solve the FJSSP for improving the performance measures of the system. Tay and Ho (2008) combined different dispatching rules as composite rules for solving the multi-objective FJSSP by improving the objective functions such as minimization of makespan, mean tardiness, and mean flow time. With sensitivity analysis the validation of the presented rules has been established and the robustness of the proposed approach is examined. Wang et al., (2011) presented the bi-population based estimation of distribution algorithm to solve the FJSSP.

Using the Taguchi tools, the parameters have been examined and then the favourable ones are identified. Using these identified parameters the algorithm performance has been examined and the best performance is realized.

Coello (1999) presents a critical review on the evolutionary based multi-objective optimization techniques and described various advantages and disadvantages with some of the know applications. Fonseca and Fleming (1995) reviewed multi-objective optimization evolutionary algorithm approaches ranging from the conventional analytical objectives into a single function to a number of population-based approaches. Tayebi Araghi et al., (2014) proposed the genetic variable neighbourhood search along with affinity algorithm to optimize maximum completion of jobs in the context of FJSSP. Along with the Taguchi based approach the algorithm parameters are tuned to their best values and in different experimental setups the experimentation has been conducted. By statistical analysis the effectiveness of the algorithm has been demonstrated and proven. Zhao et al., (2014) introduced an improved shuffled complex algorithm with opposition based learning for a permutation flow shop scheduling problem to optimize the key performance measure i.e., makespan. Zhao et al., (2015) proposed chemotaxis enhanced bacteria foraging algorithm to solve the tumble failure problem in tumble step and then with different job shop scheduling instances the performance of the algorithm has been tested and proved its efficiency. Türkyılmaz and Bulkan (2015) developed a hybridization algorithm by integrating genetic algorithm with variable neighbourhood search algorithm at the sophisticated selection phase of the genetic algorithm to solve the flexible job shop scheduling problem by minimizing the total tardiness of the system.

Motivation drawn from the fact that over the past few years, research and study Wu and Xia (2005); Zhang et al., (2010); Bagheri and Zandieh (2011); Ulungu et al., (1999) have been done by many researchers and they have focused on different hybrid evolutionary algorithms to solve the multi-objective FJSSP. Although many approaches and models have been constructed in the recent past, we have identified that there persists a need to address certain issues of the newly emerged manufacturing systems such as FJSSP. The network structure has negligible effect on the performance of the manufacturing system. A network diagram is a visual representation of the interaction of the various elements from which we derive flexible characteristics of a job amongst the process plans. This research primarily focuses on responding the following questions:

(1) How are the network diagrams from SNAM relevant to our work and how the relationships in the network involved in influence the system performance?

(2) What are the effects of the proposed social network analysis method on the considered FJSSP, and how these effects influence the considered objectives i.e., makespan and the computational time?

(3) How do different centrality measures affect the system configuration and how do these measures help to identify the key machines/ hubs which are randomly used for processing the jobs on machines till date?

SNAM allows calculating measures and drawing graphs that describe and illustrate the individual and collective structure of a network (Fidalgo 2012). The main measures calculated in SNA are centrality measures. Centrality measures identify the most prominent actors, i.e. those extensively involved in relationships with other network members (Freeman 1979). The most commonly used centrality measures are: (i) degree centrality; (ii) betweenness centrality; and (iii) closeness centrality; Degree centrality corresponds to the number of actors with whom a particular actor is directly related (Borgatti 1995). Betweenness centrality represents the number of times an actor connects pairs of other actors (Bonacich 2007). With closeness centrality, it is possible to know how closely actors are connected to the entire social network (Opsahl et al. 2010). Despite the tremendous capabilities of SNAM on addressing a wide range of problems varying from natural phenomena (Neukum and Ivanov, 1994, Lu and Hamilton 1991, and Crovella and Bestavros, 1996), military (Roberts and Turcotte, 1998), World Wide Web (Adamic and Huberman, 2000), etc. very limited work has been done in the domain of social network analysis of manufacturing problems in contrast to networked manufacturing problems.

In this paper, with SNAM approach the flexibility of self-contained structure in collaboration networks, the cohesive sub-networks (for identification of the autonomous work systems) and its descriptive statistics are collectively used to analyse the performance of the system (Wasserman and Faust, 1994) is obtained. Moreover, in FJSSP, the size, scope and complexity of the network is not defined whereas in SNAM, a clearer representation of the functional properties, such as centrality measures, network complexity and the network size is probable (Newman, 2005). The SNAM theory considers three basic elements (see e.g. (Barabási 2002; Hawe et al. 2004)): (i) Actors - network members that can be distinct individuals or collective units; (ii) Ties - link actors within a network; and (iii) Graphs - visual representation of the networks where nodes acts as the actors and the ties as lines. Additionally, there is sparse information of the communication flow within the network structure and the

descriptive statistics that can be used to extract some information about the speed or nature of the structure. In this research work, SNAM played a crucial role in identifying the key machines by making the entire process facile unlike the process of simulating the reconfiguration effect that involves step by step elimination of every machine from the system, change in various parameters, reprogramming of the software at every step of reconfiguration effect and interpretation of the result after the simulation is complete. To our knowledge, real-time event, such as machine breakdown, in FJSSP is difficult to express and no profound study has been invested in it. Thus, we have translated this research work to examine the system's behaviour when real event occurs and how much the system performance is deviated with classical FJSSP is detailed as follows:

The remainder of this paper is organized as follows. In section 2, we give a detailed problem description with the basic assumptions and developed a mathematical model along with its constraints. In section 3, we presented a framework and the logical steps of the execution of a case with proposed SNM. The detailed SNAM to find the functional properties of the network has been discussed in section 4. The experimentation with an illustrative example having different complex scenarios is illustrated and their results are presented in Section 5. The paper concludes with section 6 which suggests the directions of the future work.

2. Problem description

In this paper, flexible job shop scheduling problem (FJSSP) has been addressed and it is defined as a set of products P with k operations where k varies between $(1 \geq k \geq n)$. Each operation k_{jm} of products P is to be processed on one machine m from the set of eligible machines M_{kj}. We assume the processing time of the operations is known and all machines are continuously available at time zero i.e. before the scheduling of operations. Each machine can only process one operation at a time, and the consecutive products can wait at buffers until its preceding job should finish its process. Here, we have considered the buffer sizes are unlimited. To evaluate the performance of the system, most frequently considered performance measure such as minimization of makespan for total completion time of the processes is considered. A mathematical model has been developed with considers the objective function to conduct SNAM for analysing the reconfiguration effect on the above mentioned flexible job shop scheduling problem.

2.1. Assumptions

(a)   The considered machines before scheduling must be available at time zero.

(b)   All products can be started at time zero.

(c)   At a time it is possible to process only one operation on one machine.

(d)   Processing of operations on the machines should not be interrupted.

(e)   The sequence of operations of each job for further processing has to be pre-defined.

(f)   Release times and due dates are not specified.

(g)   Job transportation time among machines is not considered.

For simplicity, the symbols and notations are defined and explained as follows:

Z The number of operations for each job as a set. Where $Z(P) = k$ represents product P has k number of operations.

$C_{z(j)pk}$ Total completion time of Kth operation of product P on machine k

T_{rpk} Processing time of rth operation of product P on machine k

T_k Total time machine k has taken to process

Objective Functions:

$$F1: \; Min \max \left\{ C_{z(p)pk} \right\}; \; Z(p) \in Z; \; P \in N; \; k \in M \tag{1}$$

$$Subject \; \text{to:}$$

$$c_{lpk} - t_{lpk} \geq c_{(l-1)pk} \quad L = 1, 2, 3, ..., z(p); \tag{2}$$

$$P = 1, 2, 3, ..., n; \; k = 1, 2, ..., m$$

$$c_{lpk} - t_{lpk} \geq c_{lip}; \quad l = 1, 2, ..., Z(p); \tag{3}$$

$$i = 1, 2, ..., n; \; P = 1, 2, ..., m$$

$$c_{lpk} \geq 0; \quad l = 1, 2, ..., Z(p); \; P = 1, 2, ..., n; \; k = 1, 2, ..., m \tag{4}$$

From the above defined objectives, i.e. minimization of makespan ($F1$) given by Eq (1), the constraints (3) and (4) indicate processing constraints, which the preceding constraints among operations of the same job should follow so that each machine is available to other operations if the concerned operations are complete.

3. The framework of proposed SNAM approach

This section develops the framework of the proposed SNAM by identifying the key machines and its effect on the mentioned manufacturing system in a FJSSP. In order to meet the

foretasted requirements, it is imperative to develop an effective approach that can respond according to current flexible systems demand.

```
┌──────────────┐
│    Start     │
└──────┬───────┘
       ▼
┌──────────────────────────────────────────────────────┐
│ Initialization Input Data: Gather the information of   │
│ Manufacturing Execution Data.  Work Orders, Jobs,      │
│ Operations, and Machines/resources.                    │
└──────┬─────────────────────────────────────────────────┘
       ▼
┌──────────────────────────────────────────────────────┐
│ Network modeling Representation of Complex              │
│ Manufacturing Network with UCINET and Netdraw for      │
│ the manufacturing data.                                │
└──────┬─────────────────────────────────────────────────┘
       ▼
┌──────────────────────────────────────────────────────┐
│ Social network analysis Conduct tests to analyze the   │
│ performance of the obtained collaborative social       │
│ network. Identifying hubs (Key machines). Find the     │
│ size and connections strength of the network. Conduct  │
│ descriptive statistics                                 │
└──────┬─────────────────────────────────────────────────┘
       ▼
┌──────────────────────────────────────────────────────┐
│ Evaluation of the system with identified key machines  │
│ by using adopted game theory based evolutionary        │
│ algorithm approach                                     │
└──────┬─────────────────────────────────────────────────┘
       ▼
┌──────────────────────────────────────────────────────┐
│ Comparing results of adopted method with proposed      │
│ method for checking the efficiency of the matrix.      │
└──────┬─────────────────────────────────────────────────┘
       ▼
┌──────────────┐
│    Stop      │
└──────────────┘
```

Figure 1. Framework of the proposed SNAM.

In this section, we propose step by step methodology of SNAM and its implementation on FJSSP with a framework illustrated in Figure 1. The framework has been divided into three sections: (1) Model the problem as a collaborative network (2) Apply the social network analysis method to identify the key machines (3) Investigate the system performance and the effect of the identified key machines (reconfiguration) with an evolutionary algorithm based approach. Details of the collected data and the description of the steps in a methodical order are elaborated in the following sections.

4. Solution Methodology

In this section, the detailed description of how the manufacturing execution data can be extracted and viewed as a network is shown. The proposed methodology provides an opportunity for the manufacturing system environment to analyse the uncertain conditions before their occurrence. For simplicity, the SNAM is divided into two steps: (a) Modelling the problem as network, and (b) Analysing the obtained network for its consistency that is mentioned in the following sections:

4.1 Network Modelling

This section presents how the adopted manufacturing system execution data can be represented in the form of collaborative networks and how these networks connect its resources with a set of nodes is discussed (Newman, 2004).

Table 1. Data of 10 job 6 machine case

$P_{i,k,j}$		M_1	M_2	M_3	M_4	M_5	M_6
J_1	$O_{1,1}$	5	∞	4	∞	∞	∞
	$O_{1,2}$	∞	1	5	∞	3	∞
	$O_{1,3}$	∞	∞	4	∞	∞	2
	$O_{1,4}$	1	6	∞	∞	∞	5
	$O_{1,5}$	∞	∞	1	∞	∞	∞
	$O_{1,6}$	∞	∞	6	3	∞	6
J_2	$O_{2,1}$	∞	6	∞	∞	∞	∞
	$O_{2,2}$	∞	∞	1	∞	∞	∞
	$O_{2,3}$	2	∞	∞	∞	∞	∞
	$O_{2,4}$	∞	6	∞	6	∞	∞
	$O_{2,5}$	1	∞	∞	∞	∞	5
J_3	$O_{3,1}$	∞	6	∞	∞	∞	∞
	$O_{3,2}$	∞	∞	4	∞	∞	2
	$O_{3,3}$	1	6	∞	∞	∞	5
	$O_{3,4}$	∞	6	4	∞	∞	6
	$O_{3,5}$	1	∞	∞	∞	5	∞
J_4	$O_{4,1}$	1	6	∞	∞	∞	5
	$O_{4,2}$	∞	6	∞	∞	∞	∞
	$O_{4,3}$	∞	∞	1	∞	∞	∞
	$O_{4,4}$	∞	1	5	∞	3	∞
	$O_{4,5}$	∞	∞	4	∞	∞	2
J_5	$O_{5,1}$	∞	1	5	∞	3	∞
	$O_{5,2}$	1	6	∞	∞	∞	5
	$O_{5,3}$	∞	6	∞	∞	∞	∞
	$O_{5,4}$	5	∞	4	∞	∞	∞

	$O_{5,5}$	∞	6	∞	6	∞	∞
	$O_{5,6}$	∞	6	4	∞	∞	6
J_6	$O_{6,1}$	∞	∞	4	∞	∞	2
	$O_{6,2}$	2	∞	∞	∞	∞	∞
	$O_{6,3}$	∞	6	4	∞	∞	6
	$O_{6,4}$	∞	6	∞	∞	∞	∞
	$O_{6,5}$	1	6	∞	∞	5	∞
	$O_{6,6}$	3	∞	∞	2	∞	∞
J_7	$O_{7,1}$	∞	∞	∞	∞	∞	1
	$O_{7,2}$	3	∞	∞	2	∞	∞
	$O_{7,3}$	∞	6	4	∞	∞	6
	$O_{7,4}$	6	6	∞	∞	1	∞
	$O_{7,5}$	∞	∞	1	∞	∞	∞
J_8	$O_{8,1}$	∞	∞	4	∞	∞	2
	$O_{8,2}$	∞	6	4	∞	∞	6
	$O_{8,3}$	1	6	∞	∞	∞	5
	$O_{8,4}$	∞	6	∞	∞	∞	∞
	$O_{8,5}$	∞	6	∞	6	∞	∞
J_9	$O_{9,1}$	∞	∞	∞	∞	∞	1
	$O_{9,2}$	1	∞	∞	∞	5	∞
	$O_{9,3}$	∞	∞	6	3	∞	6
	$O_{9,4}$	2	∞	∞	∞	∞	∞
	$O_{9,5}$	∞	6	4	∞	∞	6
	$O_{9,6}$	∞	6	∞	6	∞	∞
J_{10}	$O_{10,1}$	∞	∞	4	∞	∞	2
	$O_{10,2}$	∞	6	4	∞	∞	6
	$O_{10,3}$	∞	1	5	∞	3	∞
	$O_{10,4}$	∞	∞	∞	∞	∞	1
	$O_{10,5}$	∞	6	∞	6	∞	∞
	$O_{10,6}$	3	∞	∞	2	∞	∞

The obtained network diagram represents nodes as entities with jobs, machines and operations. Whereas, the ties namely the edges represent the direction of interaction between the entities and the number of edges represents the degree of interaction between the entities. The data has been collected from the literature (Brandimartie, 1993) and a detailed explanation of one instance out of nine different instances has been shown in Table 1. Thereafter, with the identified FJSSP attributes (machines, jobs, and operations), information in the form of an affiliation matrix has been filled and it is shown in Figure 2.

Figure 2 shows the affiliation matrix with rows and columns whose FJSSP attributes represent the interrelationship among attributes. Where, *1* in the matrix represents a relation between the

attributes, or else it is represented as *0*. In this paper, the interactions with various attributes resemble the actual material flows between different resources. However, the data in Table 1, represents a machine performing an operation against the specific job. The data represents the machine identification numbers, which can perform the specific operation. The numbers represent the time required to finish the specific operation against the listed machines. For example, *J1* which represent Job1 will be undergoing three operations namely *O1, O2 and O3*. Narrowing it down to *O1*, this operation is performed on machines *M1-M10*. Against each machine for *O1* on *J1*, we have the processing time given, for example the processing time on *M1* is *1* unit of time. A similar kind of representation has been followed for the nine different benchmarked instances.

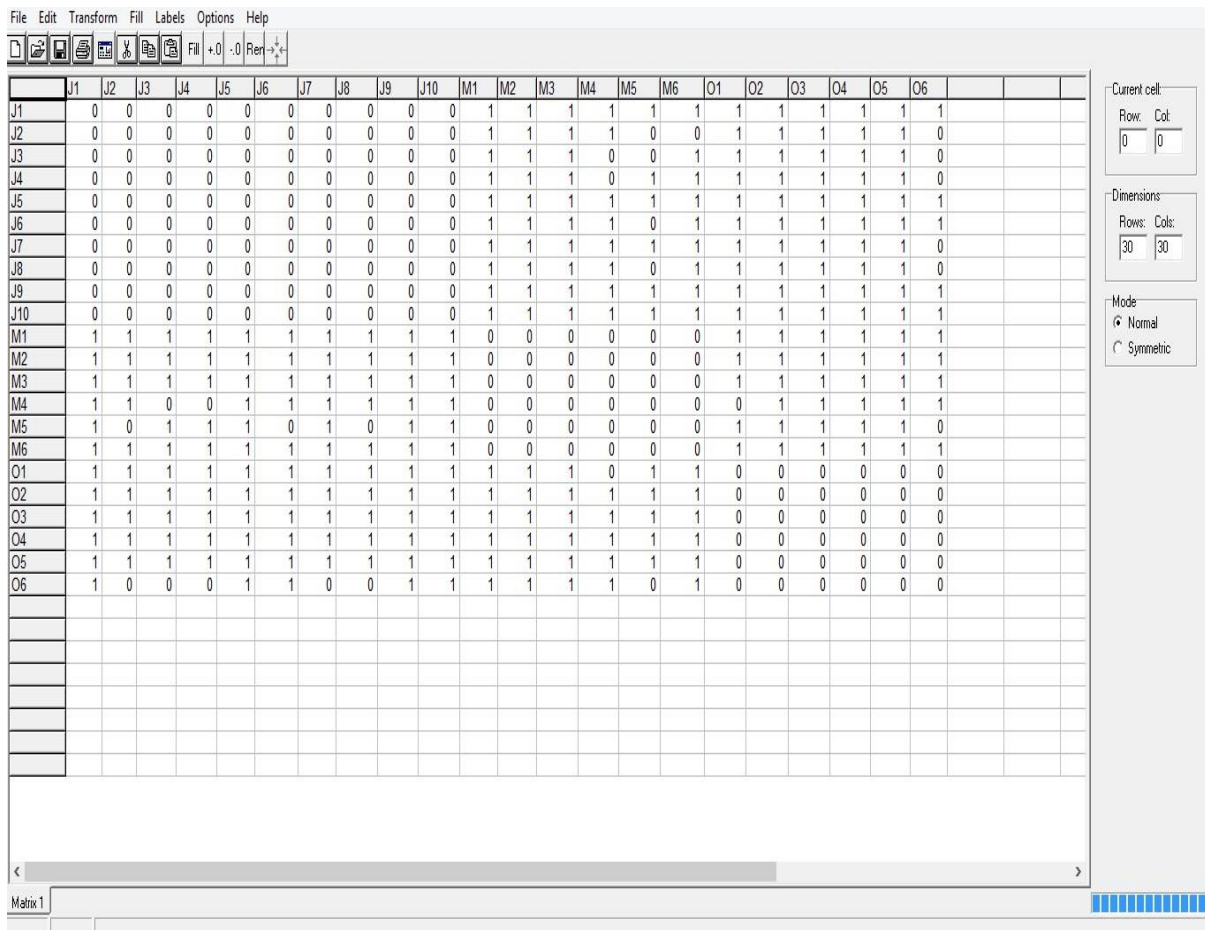|	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	M1	M2	M3	M4	M5	M6	O1	O2	O3	O4	O5	O6
J1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
J2	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	0
J3	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	1	1	1	0
J4	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	0
J5	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
J6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	1
J7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	0
J8	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	0
J9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
J10	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
M1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
M2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
M3	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
M4	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1
M5	1	0	1	1	1	0	1	0	1	1	0	0	0	0	0	0	1	1	1	1	1	0
M6	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
O1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	0
O2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
O3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
O4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
O5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
O6	1	0	0	0	1	1	0	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0

Figure 2. Affliation matrix for 10x6.

Thereafter, the matrix is analysed using the modelling algorithm in a network analysis software and an analysis has been conducted with Netdraw software which is inbuilt in Ucinet for obtaining the collaborative network. This collaboration network is more interesting and informative than the simple network in terms of its characteristics, size, etc. The arrows in the

network show interactions among attributes that is not available in the traditional representation.

The above mentioned procedure is repeated for the remaining instances to obtain their collaboration networks. Figure 2 represents an excerpt from the affiliation matrix for the data 10 jobs 6 machine case, after uploading this matrix in UCINET the obtained collaboration network is shown in Figure 3.
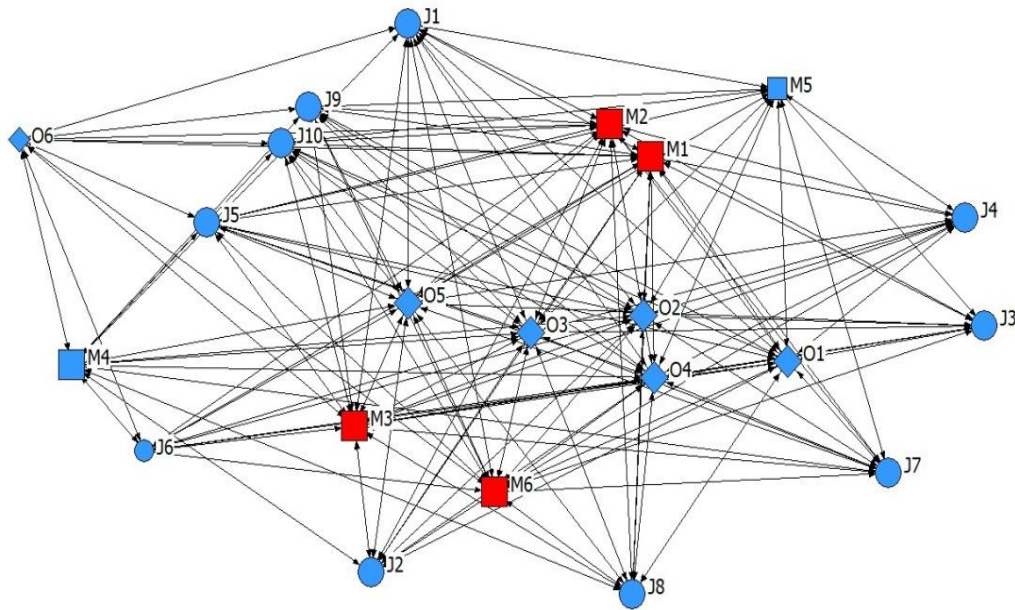

Figure 3. Collaborative network for the data set 10 by 6.

The nodes in the network shown in Figure 3 represent the direct connections that exist between attributes during the project. To distinguish each attribute we have highlighted different attributes with different colours and labelled them with their respective notation. For example, in Figure 3 the nodes with circles indicate different operations, and the square symbol indicates different machines participating for performing the task respectively. Whilst for different job nodes, we have shown with different colour and name on the node. Before analysing the network, it is necessary to conduct some preliminary analysis to describe the overall nature of the network. The detail description of the preliminary and their detailed statistical analysis of the resultant networks are specified in the following section.

4.2 Network analysis

The aim for conducting a network analysis is to observe the information of the obtained collaboration networks for potential synergies and the most important centrality measures such as degree, betweenness, and closeness centrality of different networks have been examined and their descriptive statistics have been studied in detail.

4.2.1 Network Centrality measures

A centrality is used to understand how central a node is in the network. In this research, we have considered the three most popular centrality measures such as degree, betweenness, and closeness centralities for the analysis of the network. The degree centrality measures the influence of the node on its closest neighbour with complexity of $O(n)$ to linearly scale the nodes in the network, where *n* is the number of nodes. The second centrality measure, betweenness, has the higher level of control on the information floating between different nodes in the network. Closeness measures how closely the nodes are connected to each other. One can find the detailed information of various centrality measures and their complexity (Kvernvik and Hildorsson, 2009).

In this paper, we have submitted the input data in Ucinet and then obtained the results of degree and betweenness centrality of each attribute. By observing the Table 2, which presents the results of three centralities for all the instances, we can extract a number of conclusions of the obtained collaboration networks. Based on properties of the network and their descriptive statistics i.e., maximum, minimum, number of observations and network centralization is shown in Table 3, it can be concluded that the graph is fully connected. The machines with higher degree centrality represent strongly connected, whereas the machines with lower degree centrality exhibit very less connections. Thus, we have identified the machines which are having higher degree centrality, can act as hubs and serve as key elements or central elements to the network.

Table 2. Centrality measures of different instances

Data	Machines	Degree centrality	Betweeness centrality	Closeness centrality
10 by 6	**Machine 1**	**16**	**13.516**	**80.769**
	Machine 2	**16**	**13.516**	**80.769**
	Machine 3	**16**	**13.516**	**80.769**
	Machine 4	13	8.132	72.414
	Machine 5	12	4.611	67.742
	Machine 6	**16**	1**2.238**	**77.778**
10 by 6	Machine 1	21	3.750	21
	Machine 2	21	3.750	21
	Machine 3	21	3.750	21
	Machine 4	21	3.750	21
	Machine 5	21	3.750	21
	Machine 6	21	3.750	21
15 by 8	Machine 1	25	7.790	82.051
	Machine 2	25	7.790	82.051
	Machine 3	25	7.790	82.051

	Machine 4	25	7.790	82.051
	Machine 5	25	7.790	82.051
	Machine 6	25	7.790	82.051
	Machine 7	25	7.790	82.051
	Machine 8	25	7.790	82.051
15by 8	**Machine 1**	**24**	**24.093**	**27.5**
	Machine 2	14	5.439	22.5
	Machine 3	22	18.278	26.5
	Machine 4	**24**	**24.093**	**27.5**
	Machine 5	15	7.159	23
	Machine 6	20	14.134	25.5
	Machine 7	**23**	**21.485**	**27**
	Machine 8	12	2.685	21.5
15 by 4	Machine 1	23	40.489	87.000
	Machine 2	**24**	**45.822**	**90.000**
	Machine 3	**24**	**45.822**	**90.000**
	Machine 4	**24**	**45.822**	**90.000**
20 by 5	Machine 1	25	19.400	27.000
	Machine 2	25	19.400	27.000
	Machine 3	25	19.400	27.000
	Machine 4	25	19.400	27.000
	Machine 5	25	19.400	27.000
20 by 10	**Machine 1**	**34**	**11.087**	**38.500**
	Machine 2	**34**	**11.087**	**38.500**
	Machine 3	**33**	**10.565**	**38.00**
	Machine 4	**33**	**10.565**	**38.00**
	Machine 5	**34**	**11.087**	**38.500**
	Machine 6	32	9.739	37.500
	Machine 7	**33**	**10.565**	**38.000**
	Machine 8	**33**	**10.631**	**38.000**
	Machine 9	**33**	**10.565**	**38.000**
	Machine 10	**34**	**11.087**	**38.500**
20 by 10	**Machine 1**	**32**	**28.40**	**37.000**
	Machine 2	15	2.817	28.000
	Machine 3	**32**	**19.873**	**37.000**
	Machine 4	25	7.530	33.000
	Machine 5	30	12.453	36.000
	Machine 6	0	0.000	0.000
	Machine 7	30	12.704	36.000
	Machine 8	31	13.818	36.000
	Machine 9	25	8.618	33.000
	Machine 10	**34**	**53.607**	**38.000**
20 by 15	**Machine 1**	**34**	**10.899**	**39.000**
	Machine 2	**34**	**10.899**	**39.000**
	Machine 3	**34**	**10.899**	**39.000**
	Machine 4	**33**	**10.416**	**38.500**
	Machine 5	**33**	**10.416**	**38.500**
	Machine 6	**34**	10.899	**39.000**
	Machine 7	32	9.611	38.000
	Machine 8	32	9.611	38.000
	Machine 9	**34**	**10.899**	**39.000**
	Machine 10	**34**	**10.899**	**39.000**
	Machine 11	0	0.000	0.000

Machine 12	0	0.000	0.000
Machine 13	22	4.565	33.000
Machine 14	0	0.000	0.000
Machine 15	0	0.000	0.000

Table 3. Descriptive statistics obtained from Ucinet for 10 by 6 problem

		1	2	3	4
DESCRIPTIVE STATISTICS		OutDegree	InDegree	NrmOutDeg	NrmInDeg
1	Mean	13.143	13.143	65.714	65.714
2	Std Dev	2.167	2.167	10.833	10.833
3	Sum	276	276	1380	1380
4	Variance	4.694	4.694	117.347	117.347
5	SSQ	3726	3726	93150	93150
6	MCSSQ	98.571	98.571	2464.286	2464.286
7	Euc Norm	61.041	61.041	305.205	305.205
8	Minimum	6	6	30	30
9	Maximum	18	18	90	90
10	N of Obs	21	21	21	21

Table 4. Comparison of percentage loss/gain of performance measures with the bench mark s

Data	Machines	**Evaluation by SNAM**		**Bench-mark results**	
		Effect of the network reconfiguration by step wise removing of machines		**Effect of the network reconfiguration evaluated by centrality measures**	
		Makespan			
10 by 6	**Machine 1**	6%	**Very High**	**Very High**	100%
	Machine 2	4%	**Very High**	**Very High**	100%
	Machine 3	6%	**Very High**	**Very High**	**100%**
	Machine 4	0.5%	Low	Low	**100%**
	Machine 5	0%	Low	Low	100%
	Machine 6	3%	**Very High**	**Very High**	100%
10 by 6	Machine 1	3%			100%
	Machine 2	4%	Average	Average	100 %
	Machine 3	3.5%	Average	Average	100%
	Machine 4	3%	Average	Average	100%
	Machine 5	3.5%	Average	Average	100%
	Machine 6	3%	Average	Average	100%
			Average	Average	
15 by 8	Machine 1	2%	Average	Average	100%
	Machine 2	2.5%	Average	Average	100%
	Machine 3	2.75%	Average	Average	100%
	Machine 4	2.25%	Average	Average	100%
	Machine 5	2%	Average	Average	100%
	Machine 6	2.15%	Average	Average	100%
	Machine 7	2.5%	Average	Average	100%
	Machine 8	2.25%	Average	Average	100%

15 by 8	**Machine 1**	12%	**Very High**	**Very High**	**100%**
	Machine 2	1%	Average	Average	100%
	Machine 3	4%	Average	Average	100%
	Machine 4	10%	**Very High**	**Very High**	**100%**
	Machine 5	1.25%	Average	Average	100%
	Machine 6	5%	Average	Average	100%
	Machine 7	8%	**Very High**	**Very High**	**100%**
	Machine 8	0%	Average	Average	100%
15 by 4	Machine 1	1%	Average	Average	100%
	Machine 2	9%	**Very High**	**Very High**	**100%**
	Machine 3	8%	**Very High**	**Very High**	**100%**
	Machine 4	12%	**Very High**	**Very High**	**100%**
20 by 5	Machine 1	2%	Average	Average	100%
	Machine 2	3%	Average	Average	100%
	Machine 3	2.75%	Average	Average	100%
	Machine 4	2.80%	Average	Average	100%
	Machine 5		Average	Average	100%
20 by 10	**Machine 1**	11%	**Very High**	**Very High**	**100%**
	Machine 2	9%	**Very High**	**Very High**	**100%**
	Machine 3	3%	**Very High**	**Very High**	**100%**
	Machine 4	5%	**Very High**	**Very High**	**100%**
	Machine 5	10%	**Very High**	**Very High**	**100%**
	Machine 6	0%	Average	Average	100%
	Machine 7	3%	**Very High**	**Very High**	**100%**
	Machine 8	7%	**Very High**	**Very High**	**100%**
	Machine 9	4%	**Very High**	**Very High**	**100%**
	Machine 10	14%	**Very High**	**Very High**	**100%**
20 by 10	**Machine 1**	9%	**Very High**	**Very High**	**100%**
	Machine 2	0%	Average	Average	100%
	Machine 3	14%	**Very High**	**Very High**	**100%**
	Machine 4	0.5%	Average	Average	100%
	Machine 5	3%	Average	Average	100%
	Machine 6	0%	Average	Average	100%
	Machine 7	1%	Average	Average	100%
	Machine 8	2%	Average	Average	100%
	Machine 9	1%	Average	Average	100%
	Machine 10	18%	**Very High**	**Very High**	**100%**
20 by 15	**Machine 1**	13%	**Very High**	**Very High**	**100%**
	Machine 2	12%	**Very High**	**Very High**	**100%**
	Machine 3	13%	**Very High**	**Very High**	**100%**
	Machine 4	9%	**Very High**	**Very High**	**100%**
	Machine 5	7%	**Very High**	**Very High**	**100%**
	Machine 6	11%	**Very High**	**Very High**	**100%**
	Machine 7	1%	Average	Average	100%
	Machine 8	2%	Average	Average	100%
	Machine 9	8%	**Very High**	**Very High**	**100%**
	Machine 10	10%	**Very High**	**Very High**	**100%**
	Machine 11	0%	Average	Average	100%
	Machine 12	0%	Average	Average	100%
	Machine 13	3%	Average	Average	100%
	Machine 14	0%	Average	Average	100%
	Machine 15	0%	Average	Average	100%

The identified key machines for all different instances are shown in network diagram as square symbols with red ink and it is shown in Figure 4. These key machines or hubs have a wide spectrum of strength connections; even the shop floor manager can easily identify the hubs in the manufacturing system prior to actual scheduling by observing the collaborative networks. Accordingly, the load of the system can be better controlled thereby improving the effectiveness and efficiency of the scheduling process.

5. Experimentation

The proposed SNAM is implemented on different benchmark instances in FJSSP. From the above mentioned SNAM, we have identified different functional properties from the network. On the other hand, from the analysis section, attribute of every node is examined that plays a crucial role in identifying the key elements in the network. We identified from 10 by 6 problem data that machine 1, 2, 3, and 6 obtained higher degree, betweenness and closeness centrality, similarly the other problems data and their centrality measures are shown in Table 4. The identified machines with higher centralities act as hubs/ key machines in the network and their influence on the entire network is much higher. To prove this, from the 10 by 6 data we have removed machine *1* and then examined the performance measure i.e. makespan by executing with already developed game theoretic based hybrid dynamic-DNA (HD-DNA) algorithm Manupati et al. (2012). After removing the respective machine, the problem becomes a flexible process planning and scheduling problem where integration of process planning and scheduling (IPPS) approach is implemented to calculate the makespan. Figure 5 details the flowchart of IPPS with game theory based HD-DNA approach to evaluate the performance measures of the system. One can find the detailed description of IPPS and the implementation of game theory based evolutionary algorithms on the proposed problem in Manupati *et al.* (2012) and Zhou et al. (2010). The proposed algorithm has been coded in MATLAB and from the results we have observed sparse influence on the performance of the system due to its lower centralities. The detailed information of the performance of the machines and their effect on the network is illustrated in Table 4. Similarly, we have tested the performance of the system and the influence of the network, in a step by step removal of the machines from the data.

The above procedure is repeated for all the different benchmark instances mentioned in this research work. In Table 4, the first column represents different scenarios and their respective machines are represented in column 2, where column 3 indicates the percentage of

loss/ gain of the performance measures by removing respective machines from the data and the last column shows the effect on the performance manufacturing system.
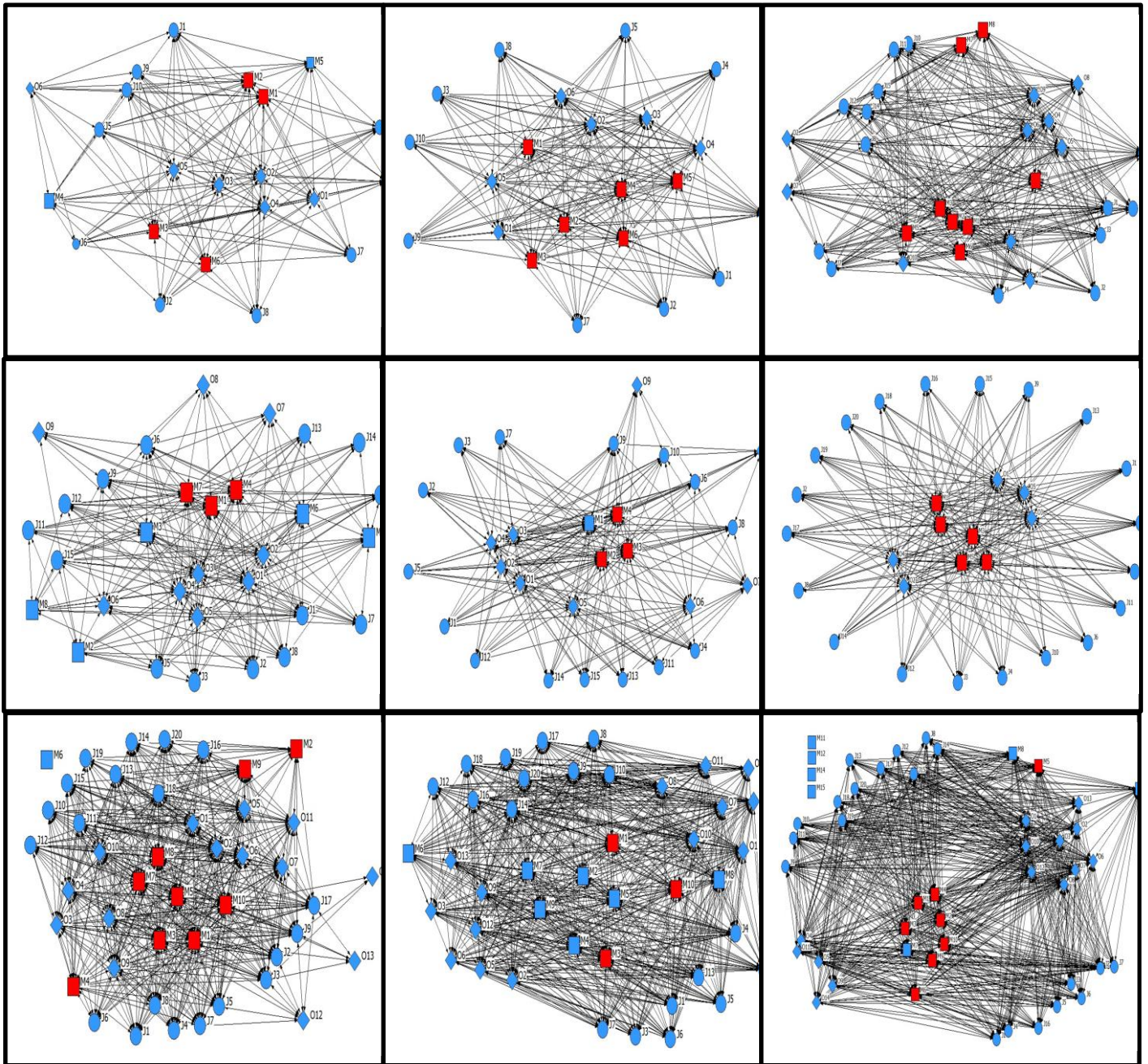
Figure 4. Collaborative networks of all the nine instances with their key machines.

For validation, we have used results from Brandimartie (1993), as benchmark for finding the percentage loss of makespan. For example, for 10 by 6 problem instance the optimal makespan value of a benchmark problem is 70 and we obtained 78, which is the optimal makespan after removing the machine 1, thus the loss of percentage again after removing machine 1 is eight and it is shown in Table 4. The comparison between the benchmark results

and the key machines along with their performance measures obtained from the proposed method indicates similar results. With this, we prove the effectiveness of the proposed model.
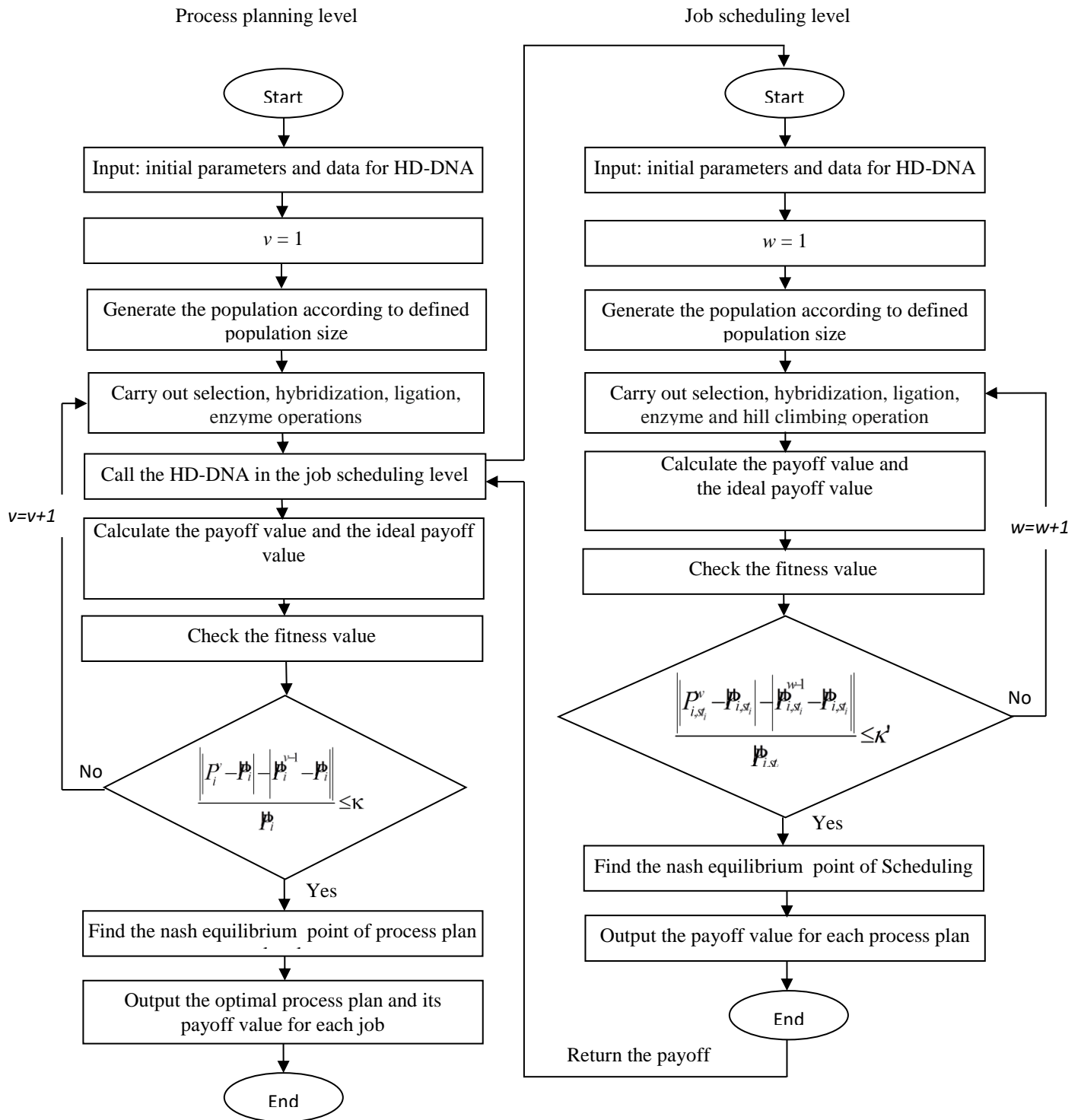
Figure 5. Flowchart for process planning and job scheduling of the game (Manupati et al. 2012).

It is clear from the results that the key machines are tremendously affecting the network and the behaviour of the system. In Figure 6 we have shown the different data sets and the effect on performance measures after removing the machines. The peak points in the Figures 6 a drop in the performance after removing the key machines. Finally, the analysis on different experimental settings has shown consistency in approach. Hence, we can say that with the social network analysis method it is highly possible to find the key elements effectively.
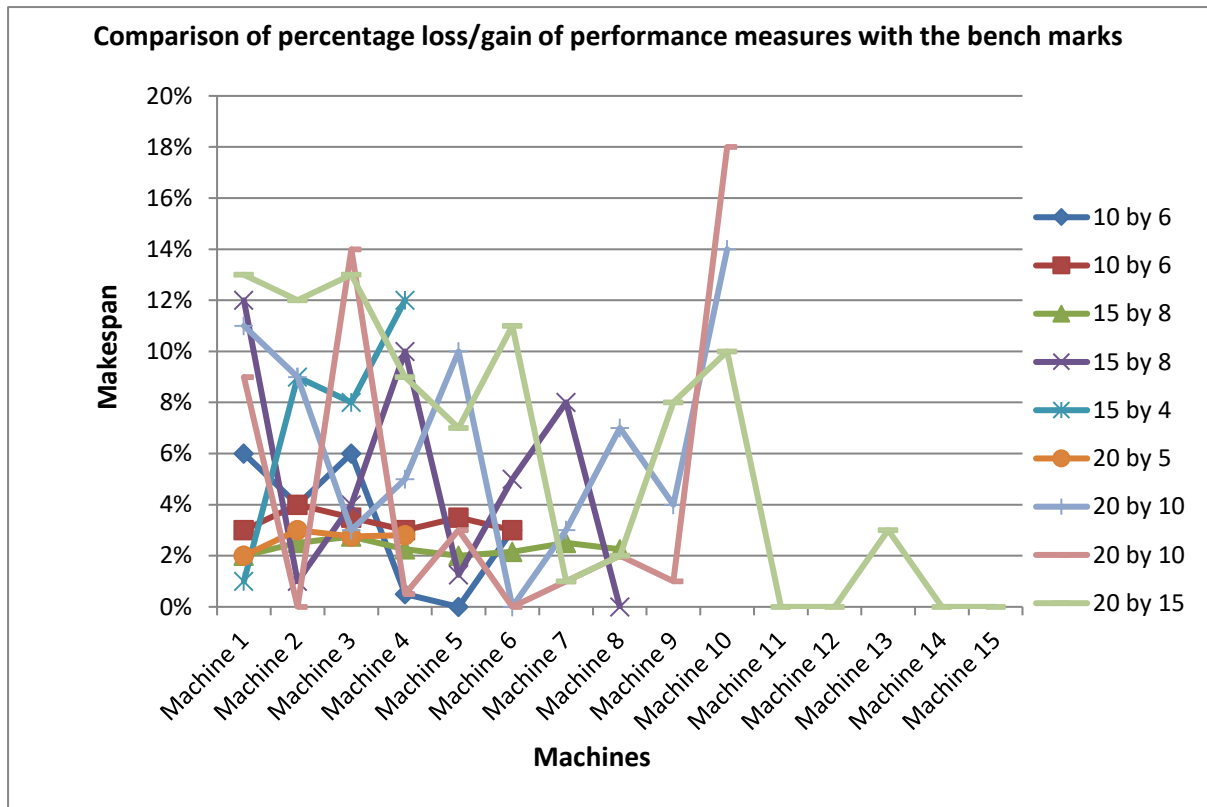
Figure 6. Percentage of effect on makespan by removing machines.

6. Conclusion and Future work

In this paper, a social network analysis method (SNAM) has been proposed to evaluate the reconfiguration effect on makespan in a flexible job shop scheduling problem. A mathematical model with respect to the constraints has been developed by considering makespan as one of the performance measures of the system. More importantly, we have introduced a conceptual model with the help of a proposed framework that investigates the reconfigurability of the desired objective. The proposed model is executed by first converting the manufacturing execution data into an affiliation matrix, followed by entering the input data to the Ucinet software package to obtain the collaborative networks. Moreover, we have mapped the structure of the obtained network with the attributes of the manufacturing system to identify

the hubs (key elements or machines). To prove the effectiveness of the proposed method, we have illustrated an example having various complex scenarios whose experimentation has been conducted. We have used an evolutionary algorithm based approach for integration of process planning and scheduling, to optimize the parameters. Various comparisons have been made with the proposed approach as well existing approaches in the literature to prove the effectiveness of the model.

In order to find the reconfiguration effect of makespan, we have chosen the identified key machines to conduct the experimentation. To validate the role of hubs on the performance of the manufacturing system i.e., obtained from SNAM, the recently proposed effective hybrid evolutionary algorithm approach has been used for carrying the experiments. Results confirmed that the obtained hubs from SNAM are highly influencing the performance of the manufacturing system and its behaviour.

A more profound research effort needs to focus on the possibility of self-organization in the system along with different problem environments, the effectiveness and efficiency of the approach should be studied.

Acknowledgment

References

Adamic, L.A. and Huberman, B.A., (2000). Power-law distribution of the world wide web. *Science*, 287(5461), pp.2115-2115.

Bagheri, A; Zandieh, M; (2011). Bi-criteria flexible job shop scheduling with sequence-dependent setup times—variable neighbourhood search approach. *Journal of Manufacturing Systems* 30: pp.8–15.

Brandimarte, P; (1993). Routing and scheduling in a flexible job shop by Tabu search. *Annals of Operations Research.* 41, pp.157–83.

Borgatti, S.P., (1995). Centrality and AIDS. Connections, 18(1), pp.112-114.

Bonacich, P., (2007). Some unique properties of eigenvector centrality. *Social networks*, 29(4), pp.555-564.

Barabási, A.L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A. and Vicsek, T., (2002). Evolution of the social network of scientific collaborations. *Physica A: Statistical mechanics and its applications*, 311(3), pp.590-614.

Coello, C.A.C., (1999). A comprehensive survey of evolutionary-based multi-objective optimization techniques. *Knowledge and Information systems*. *1*(3), pp.269-308.

Crovella, M.E. and Bestavros, A., (1996). Self-similarity in World Wide Web traffic: Evidence and possible causes. ACM SIGMETRICS *Performance Evaluation Review*, 24(1), pp.160-169.

Fonseca, C.M. and Fleming, P.J., (1995). An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary computation*, *3*(1), pp.1-16.

Fidalgo, P. and Thormann, J., (2012). A social network analysis comparison of an experienced and a novice instructor in online teaching. *European Journal of Open, Distance and E-learning*, 15(1).

Garey, M.R; Johnson, D.S; Sethi, R; (1976). The complexity of flow shop and job shop scheduling. *Mathematics and Operations Research*. 1, pp.117–29.

Hurink, J; Jurisch, B; Thole, M; (1994). Tabu search for the job shop scheduling problem with multi-purpose machines. *Journal of OR Spectrum*. 15, pp.205–15.

Hawe, P., Webster, C. and Shiell, A., (2004). A glossary of terms for navigating the field of social network analysis. *Journal of epidemiology and community health*, 58(12), pp.971-975.

Lu, E.T. and Hamilton, R.J., (1991). Avalanches and the distribution of solar flares. *The astrophysical Journal*, 380, pp.L89-L92.

Manupati, V.K; Deo, S; Cheikhrouhou, N; and M.K.Tiwari; (2012). Optimal Process Plan Selection in Networked Based Manufacturing Using Game-theoretic Approach. *International Journal of Production Research*. 50 (18), pp.5239–5258.

Najid, N.M; Dauzere-Peres. S; Zaidat, A; (2002). A modified simulated annealing method for flexible job shop scheduling problem. *IEEE International conference on systems, man and cybernetics*. 5.

Newman, M.E., (2005). A measure of betweenness centrality based on random walks. *Social networks*, 27(1), pp.39-54.

Neukum, G. and Ivanov, B.A., (1994). Crater size distributions and impact probabilities on Earth from lunar, terrestrial-planet, and asteroid cratering data. *Hazards Due to Comets and Asteroids, Publisher: University of Arizona Press, Tucson, Arizona*, pp.359-416.

Opsahl, T., Agneessens, F. and Skvoretz, J., (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32(3), pp.245-251.

Roberts, D.C. and Turcotte, D.L., (1998). Fractality and self-organized criticality of wars. Fractals, 6(04), pp.351-357.

Saygin, C; and Kilic, S. E; (1999). Integrated flexible process plans with scheduling in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology.* 15 : pp.268-280.

Tay, J.X; and Ho. N.B; (2008). Evolving dispatching rules using genetic programming for solving multi-objective flexible job shop problems. *Journal of Computers and Industrial Engineering*. 54: pp.453–73.

Tayebi Araghi, M.E., Jolai, F. and Rabiee, M., (2014). Incorporating learning effect and deterioration for solving a SDST flexible job-shop scheduling problem with a hybrid meta- heuristic approach. *International Journal of Computer Integrated Manufacturing*, *27*(8), pp.733-746.

Türkyılmaz, A. and Bulkan, S., (2015). A hybrid algorithm for total tardiness minimisation in flexible job shop: genetic algorithm with parallel VNS execution. *International Journal of Production Research*, *53*(6), pp.1832-1848.

Ulungu, E.L; Teghem, J; Fortemps, P.H; and Tuyttens, D; (1999). MOSA method: a tool for solving MOCO problems. *Journal of Multi-Criteria Decision Analysis*. 8, pp.221–36.

Wang, L; Wang, S; Xu. Y; Zhou. G; Liu, M; (2011). A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem. *Journal of Computers and Industrial Engineering*, 64: pp.917-26.

Wasserman, S. and Faust, K., (1994). Social network analysis: Methods and applications. Cambridge university press.

Xia, W; Wu, Z; (2005). An effective hybrid optimization approach for multi-objective flexible job shop scheduling problems. *Journal of Computers and Industrial Engineering*. 48: pp.409–25.

Yang, J.B; (2001). GA-based discrete dynamic programming approach for scheduling in FMS environments. *IEEE Transactions on Systems, Man and Cybernetics, Part B*. 31: pp.824–35.

Zhang, G; Gao, L; Shi, Y; (2011). An effective genetic algorithm for the flexible job shop scheduling problem. *Expert System with Applications*. 38, pp.3563–73.

Zhang, G.H; Shao, X.Y; Li, P.G; and Gao, L; (2010). A genetic algorithm and tabu search for multi- objective flexible job shop scheduling problems. *IEEE International Conference on Computer Control and Industrial Engineering.* 1: pp.251–4.

Zhao, F; Jiang, X; Zhang, C; and Wang, J; (2015). A chemotaxis-enhanced bacterial foraging algorithm and its application in job shop scheduling problem. *International Journal of Computer Integrated Manufacturing*. 28, pp.1106-1121.

Zhao, F., Zhang, J., Wang, J. and Zhang, C., (2014). A shuffled complex evolution algorithm with opposition-based learning for a permutation flow shop scheduling problem. *International Journal of Computer Integrated Manufacturing*, DOI:10.1080/0951192X.2014.961965,116.