Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 16(5): 4491–4505.
DOI: 10.3934/mbe.2019224
Received: 30 March 2019
Accepted: 15 May 2019
Published: 21 May 2019

Research article

A novel complex network based dynamic rule selection approach for open shop scheduling problem with release dates

Zilong Zhuang, Zhiyao Lu, Zizhao Huang, Chengliang Liu and Wei Qin*

School of Mechanical Engineering, Shanghai Jiao Tong University, 200240, Shanghai, China

*** Correspondence:** Email: wqin@sjtu.edu.cn; Tel: +8613472790428.

Abstract: In the open shop scheduling problem, resources and tasks are required to be allocated in an optimized manner, but when the arrival of tasks is dynamic, the problem becomes much more difficult. To solve large scale open shop scheduling problem with release dates, heuristic algorithms are more promising compared with metaheuristic algorithms. In this paper, a framework of general scheduling object is developed, under which open shop scheduling problem is described. Then, a complex scheduling network model for open shop scheduling problem is established, and the problem is transformed into reasonably arranging the node traversal order with the goal of traversing all nodes in the network as quickly as possible, on condition that each node has a traversal time and only disconnected nodes can be traversed simultaneously. Considering that network structural features and local time attributes of nodes can provide heuristic information, six single heuristic rules are raised and a novel complex network based dynamic rule selection approach is proposed to solve dynamic open shop problem by switching dynamically the scheduling rules based on real-time production status. Finally, two experiments are carried out and the experimental results show that the proposed heuristic rules have acceptable performance, and the proposed complex network based dynamic rule selection approach is feasible.

Keywords: complex network; open shop; heuristic rule; node traversal order; dynamic rule selection

1. Introduction

Scheduling is one of the most widely researched areas of operational research, and its primary objective is to optimize one or more performance indicators by allocating scarce resources to productive operations in a given period [1–3]. From the perspective of machine environment, there are five environments in literature, namely, single machine shop, parallel machine shop, flow shop, job shop and open shop. Different from other machine environments, open shop has no restriction on the processing route of each job during the production process. In this paper, the open shop scheduling problem is considered.

Since open shop scheduling problem (OSSP) was raised, it has received considerable attention over the last four decades and has been applied to different fields including agriculture, hospitals, transport, and manufacturing industry [4,5]. In classic OSSP, a set of n jobs are to be processed by a set of m machines, with all jobs and machines available at the beginning. Each job contains a set of m operations, and each operation should only be processed on one machine. At any time, a job can be processed by at most one machine, and any machine can process only one job. Besides, setup times are negligible and preemption of operations is not allowed. The problem is to find an optimal schedule for the operations that minimizes the makespan. Due to unconstrained operational processing order within a job, OSSP can be perceived as a generalization of the job shop scheduling problem (JSSP) [6]. Compared with JSSP, the solution space of OSSP (represented by the sequence of operations on machines and operations within jobs) is too large to find the optimal solution.

With the standard scheduling notation [7], the simplest OSSP can be described as $O_m||C_{max}$, where m is the number of machines. For problem $O_2||C_{max}$, a priority rule named *Longest Alternate Processing Time first* (LAPT) was developed to find the optimal scheduling in polynomial time [8]. Besides, an NP-hardness proof for $O_3||C_{max}$ was provided by [4]. As the job is available only after its arrival in practice, the study of open shop scheduling problem with release dates is closer to the practical production [9], but it also makes it harder to arrange resources and tasks properly. Lawler, Lenstra, Kan, et al. [10] pointed out that the problem $O_2|r_j|C_{max}$ is strongly NP-hard. For problem $O_3|r_j|C_{max}$, Chen, Huang and Tang [11] demonstrated that the worst-case performance ratio of greedy algorithm is 5/3. For small scale problems, Branch & Bound algorithm is the best choice [12,13]. For large scale problems, heuristic or metaheuristic algorithms may be the most effective way to obtain approximate optimal solutions. For example, hybrid genetic algorithms have been employed to solve large scale OSSP [14,15] while hybrid genetic algorithm and particle swarm optimization are introduced in integrated process planning and scheduling [16,17]. In addition, artificial bee colony algorithm is used in welding shop scheduling problem [18,19]. However, in terms of the scheduling problem with release dates, metaheuristic algorithms have great difficulty in modeling. Besides, metaheuristic algorithms may take much more time to obtain suboptimal solutions, making it difficult to meet the performance requirements of real-time scheduling decision. As the scale of the problem increases, the metaheuristic algorithms will fail fundamentally due to their inability to evaluate the performance of the solution in a timely and effective manner. In comparison, the heuristic algorithms are more promising as they achieve better trade-offs between computation time and solution quality. However, it remains a challenge to design appropriate heuristic algorithms for different problems.

In order to model and optimize large-scale complex dynamic systems in the real world, since the 1990s, complex network theory has been developed in the field of statistical physics [20,21]. Complex network consists of numerous nodes and edges, which respectively represent elements of the system and the correlations between the elements, providing a new way to deal with large-scale dynamic scheduling problem. The complex network theory holds that the complexity of a network is mainly caused by the complex association between a large number of nodes, rather than by the complex dynamic behavior of the individual nodes. Similarly, complex relationships among jobs, resources and operations determine the performance of the entire scheduling system. Therefore, complex network model may be an effective tool for solving large-scale dynamic open shop scheduling problem.

In recent years, complex network theory has already found its way into the scheduling problems. To the best of our knowledge, the first complex network model for open shop scheduling problem was built by [22,23]. The open shop scheduling problem was transformed into reasonably arranging the node traversal order, with the goal of traversing all nodes in the network as quickly as possible. Moreover, three scheduling rules based on complex network characteristics, namely Degree rule, Cluster rule and Redundancy rule were proposed. This research is very instructive, but has obvious limitations. Specifically, these proposed rules only apply to non-uniform network topologies, which are obtained by setting jobs with different numbers of operations. Therefore, for classical production scheduling problem, the performance of these algorithms will be greatly compromised due to their inability to identify valid initial scheduling nodes. Besides, a multi-task directed-weighted network was established by setting production factors involved in the job shop system as nodes, possible process routes and logistics paths between nodes as edges [24]. In conclusion, the production scheduling research based on complex network is still in its infancy. In this paper, both the network structural features and time attributes are considered to construct effective heuristic rules.

The rest of the paper is organized as follows: section 2 describes the framework of general scheduling object; Section 3 develops a complex scheduling network model for OSSP; Section 4 proposes several complex network based heuristic rules, and a novel complex network based dynamic rule selection approach. In section 5, two groups of experiments are conducted. Finally, conclusions and future works are summarized.

2. Framework of general scheduling object

The general scheduling object involves many elements that have specific meanings in different application scenarios. In many cases, the relationship between these elements is quite complicated. In order to solve the scheduling problem in as many application scenarios as possible, a framework of general scheduling object is established based on the induction of the basic elements of the object, which can cover various types of scheduling problems, including OSSP.

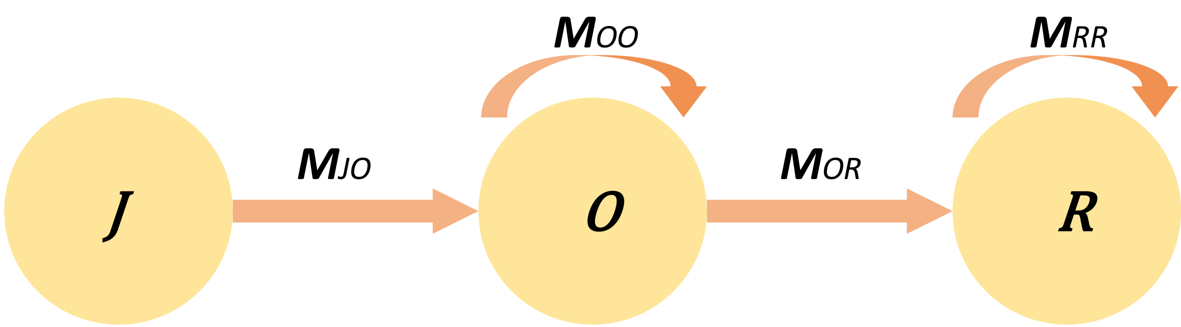

Figure 1. Structure diagram of the general scheduling object.

As shown in Figure 1, a general scheduling object (GSO) can be defined by a formula:

$$GSO = [R, J, O, M_{JO}, M_{OR}, M_{OO}, M_{RR}] \tag{1}$$

where R, J, O, M_{JO}, M_{OR}, M_{OO} and M_{RR} denotes resource set, job set, operation set, job-operation mapping, operation-resource mapping, operation-operation mapping and resource-resource mapping, respectively.

Resource set R can be simply defined as (2), which is made up of m resources.

$$R = \{r_1, r_2, ..., r_m\} \tag{2}$$

Each element in the job set J represents a job, which can contain some attributes, such as priority level, release date and due date. J with size n can be represented as (3). The release date RD of tasks can be represented as (4), and the due date DD of tasks can be described as (5).

$$J = \{J_1, J_2, ..., J_n\} \tag{3}$$

$$RD = \{rd_1, rd_2, ..., rd_n\} \tag{4}$$

$$DD = \{dd_1, dd_2, ..., dd_n\} \tag{5}$$

Job-operation mapping M_{JO} can be expressed as (6). Each job J_i is composed of n_i operations to be processed, and O_i^k denotes the kth operation of job J_i (k=1, 2,…, n_i), which has the processing time t_i^k . For flow shop and job shop, each job J_i is an ordered set.

$$J_i = \{O_i^1, O_i^2, ..., O_i^{n_i}\} \tag{6}$$

Operation-resource mapping M_{OR} can be presented as (7), of which the number of resources is $size_i^k$ given that the processing of operation O_i^k usually requires the cooperation of several resources.

$$OR(O_i^k) = fix_i^k \tag{7}$$

In order to approach the real-world problem, the setup time between any two operations should not be ignored, and the length of the setup time on the machine depends on the similarity between the two consecutive operations. The higher the similarity, the shorter the setup time. Operation-operation mapping M_{OO} can be expressed as (8), where S denotes the total number of operations, and can be calculated by (9). Any element $OO_{i,j}$ represents the switching time from ith to jth operation.

$$M_{OO} = [OO_{i,j}]_{S*S} \tag{8}$$

$$S = \sum_{i=1}^{n} n_i \tag{9}$$

The occurrence of resources in parallel is common in the actual systems. Therefore, it is important to figure out if the resources can perform the same functions. Resource-resource mapping M_{RR} can be formulated by a matrix, in which any element $RR_{i,j}$ represents the substitution efficiency of the corresponding two resources, specifically, $RR_{i,j}$ equals the ratio of the time spent on the *i*th resource to the *j*th resource.

$$M_{RR} = [RR_{i,j}]_{m*m} \tag{10}$$

Most scheduling problems can be described under the framework of GSO, including OSSP. $O_m||C_{max}$ can be obtained from the GSO under these constraints: (1) the resource set R is made up of *m* unrelated resources, that is, no resource can be replaced by any other resource, thus, M_{RR} is an unit matrix; (2) the job set *J* is made up of n jobs, and the release date RD of any task is set to zero, regardless of the due date DD; (3) each job J_i is composed of *m* operations to be processed, and its *k*th operation O_i^k needs to be processed for t_i^k unit time on the *k*th resource. Thus, $fix_i^k = \{k\}$ and $size_i^k = 1$ for all *i* and *k*; (4) M_{OR} is consistent with the above expression; (5) all setup times are assumed to be zero, so M_{OO} is zero matrix. When the release date of the *i*th job rd_i is set as r_i, the problem is converted into $O_m|r_j|C_{max}$.

3. Complex scheduling network model for OSSP

Under the framework of GSO, most scheduling problems can be easily transformed into complex scheduling network models. The complex scheduling network G can be represented by a triad *G*= (*V*, *L*, *UR*), where *V* denotes the set of nodes, *L* denotes the set of edges, and *UR* denotes the set of network update rules.

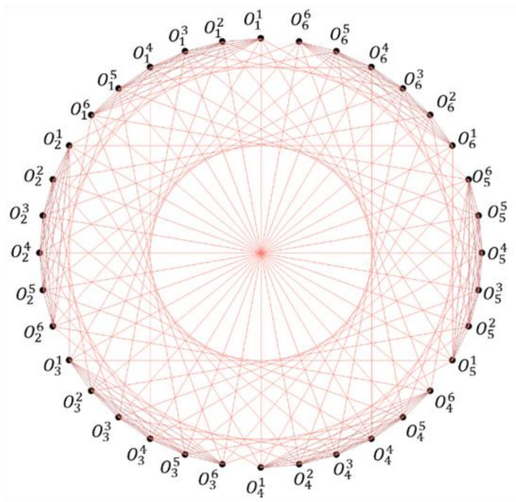

Figure 2. A complex scheduling network model for OSSP with 6 jobs and 6 machines.

As shown in Figure 2, in the complex scheduling network model for OSSP, a node O_i^k represents the kth operation of the ith job, assigned two variables respectively representing the required processing time t_i^k, and the release date rd_i. Every edge represents mutually exclusive timing constraint between two operations. Specifically, since any two operations of the same job can't be processed simultaneously, an edge will link any pair of nodes of the same job (O_i^k and O_i^l). Similarly, since a resource can only be used for processing one operation at a time, an edge will link any pair of nodes that require the same resource (O_i^k and O_j^k). Then the problem can be formulated as: how to arrange the node traversal order so that all the nodes in the network can be traversed as quickly as possible, on condition that each node has a traversal time, and only the disconnected nodes can be traversed simultaneously.

As is known to us all, the network update rules *UR* can be divided into two parts. One part is growth rule (*GR*), and the other part is cutting rule (*CR*). *GR* controls the increase of network nodes and edges caused by the dynamic arrival of jobs, and *CR* determines the disappearance of network nodes and edges with the completion of operation. For $O_m|r_j|C_{max}$, *GR* is predictable, but not fundamentally controllable. However, *CR* can be controlled to some extent by designing different scheduling algorithms.

4. Complex network based dynamic rule selection approach

Before presenting the complex network based dynamic rule selection approach, it is necessary to design several heuristic rules based on complex networks. The complex scheduling network model expresses the constraints between operations. Analysis of the topological features of complex networks helps simplify complex scheduling problems and inspire the design of heuristic rules based on complex networks. Given the fact that complex scheduling problems with different timing scheduling objectives can be congruously mapped to the node traversal problem on the complex scheduling network model, the general idea for generating complex network based heuristic rule is identical. The systematic heuristic rule generation approach based on network topological features can be divided into four steps:

(1) Establish complex scheduling network model for given scheduling object;
(2) Extract global features related to scheduling optimization objectives;
(3) Structure local features related to global features;
(4) Design heuristic rules based on the local features.

Intuitively, the more mutually exclusive timing constraints between the nodes in a network, the stronger the coupling between nodes, and consequently, the lower the traversal efficiency, and vice versa. Therefore, one feasible approach is to prioritize nodes that can significantly reduce system coupling after removal. Given that network average degree and network average efficiency can distinctly reflect this coupling, local topological features such as degree and clustering coefficient can be used as heuristic information. Therefore, the *Largest Degree first* (LD) rule and the *Smallest Cluster Coefficient first* (SCC) rule can be employed. In addition, the time attributes of nodes can also provide heuristic information, especially for non-uniform complex scheduling networks. Hence, the *Longest Processing Time first* (LPT) rule and the *Shortest Processing Time first* (SPT) rule are worthy of consideration. To combine the heuristic information from network topological features and time attributes of nodes, the *Longest Total Remaining Processing on Adjacent Operations first* (LTRPAO) rule and the *Longest Total Remaining Processing on Other Machines first* (LTRPOM) rule are developed. The effectiveness and performance of these six heuristic rules will be confirmed and compared in the experiments of Section 5.

Previous studies have shown that any elevated performance of an algorithm over one class of problems is offset by its performance over another class [25], which means no single dispatching rule performs dominantly better than any other in all scheduling environments [26]. Thus, designing or improving heuristic rules may not be the best research direction for solving NP-hard scheduling problems. With the arrival of new jobs and the completion of the operation, the production status changes over time, therefore, it is necessary to select the appropriate scheduling rules dynamically based on real-time production status, involving the effectiveness evaluation and selection of scheduling rules. The principle of dynamic rule selection based on complex networks can be divided into the following three steps:

(1) Calculate the attribute values of each node under different rules for each scenario.
(2) Design an effectiveness evaluation scheme based on the distribution of attribute values of nodes under different rules.
(3) Select the rule with the highest effectiveness as the rule in the current scenario.

Since the local features of nodes reflect the status of nodes in the network, the differences of local features provide heuristic information [27,28]. The larger the differences, the better effect the heuristic rule may achieve. Therefore, a feasible scheduling rule effectiveness evaluation scheme is to evaluate the difference in node attribute values under different rules. Considering that the node attribute values of different rules have different magnitudes and only one node needs to be selected at a time according to one scheduling rule, the ratio of the attribute value of the optimal node to the attribute value of the suboptimal node under different rules can be used as the validity index of the rule. The proof experiment of the proposed complex network based dynamic rule selection approach is also presented in Section 5.

5. Experiment results and discussion

To investigate the effectiveness of the proposed heuristic rules based on complex network, and to check the feasibility of the proposed complex network based dynamic rule selection approach, two experiments are carried out, respectively. All the experiments are executed on R2017a Matlab, 8 GB RAM and i7 processor.

5.1. Case study I: Heuristic rules for $O_m||C_{max}$

In the first experiment, the open shop benchmarks from [29] are used to fully test the proposed heuristic rules based on complex networks, compared with the currently known optimal solution, and the chosen benchmarks are identical with the experiment in [30]. In order to verify whether the topology uniformity of the complex scheduling network has a significant impact on the performance of the rules, a carefully crafted set of test problems is generated for lack of the benchmark instances in the literature. To obtain the non-uniform open shop benchmarks with m jobs and m machines, the number of operations that are randomly removed from each job of the above benchmarks is subject to a discrete power-law distribution [31,32], with the range from 1 to m-1.

Table 1. Results of heuristic rules for $O_m||C_{max}$.

Benchmark	BKS	LD	SCC	LPT	SPT	LTRPAO	LTRPOM
10×10^-1	637	702	702	**674**	683	**674**	682
10×10^-2	588	629	629	658	**624**	639	626
10×10^-3	598	711	711	**698**	706	708	749
10×10^-4	577	690	690	**668**	680	689	710
10×10^-5	640	**685**	**685**	723	707	707	746
10×10^-6	538	668	668	666	**660**	665	678
15×15^-1	937	958	958	946	995	**944**	948
15×15^-2	918	998	998	1011	1013	**981**	1019
15×15^-3	871	1013	1013	933	933	**905**	925
15×15^-4	934	1064	1064	**1055**	**1055**	1063	1067
15×15^-5	946	978	978	**973**	1024	986	1012
15×15^-6	933	1043	1043	983	987	**972**	1009
20×20^-1	1155	1290	1290	**1230**	1234	1314	1252
20×20^-2	1241	1277	1277	**1267**	1316	1277	1287
20×20^-3	1257	1391	1391	1375	**1366**	1386	1398
20×20^-4	1248	1289	1289	1282	1340	**1280**	1337
20×20^-5	1256	**1276**	**1276**	1279	1286	1305	1293
20×20^-6	1204	1260	1260	1247	1254	**1246**	1251
10×10^-1’	-	632	**631**	634	671	650	664
10×10^-2’	-	**527**	574	562	579	533	543
10×10^-3’	-	650	669	**646**	668	650	655
10×10^-4’	-	**584**	605	622	591	615	609
10×10^-5’	-	**639**	665	692	651	670	659
10×10^-6’	-	661	**660**	**660**	**660**	693	662
15×15^-1’	-	923	**883**	909	911	908	908
15×15^-2’	-	996	970	**965**	975	992	976
15×15^-3’	-	898	855	870	**854**	861	869
15×15^-4’	-	1082	**1047**	1053	1059	1048	1073
15×15^-5’	-	938	**895**	927	901	928	969
15×15^-6’	-	925	972	**909**	942	916	917
20×20^-1’	-	1136	1144	1170	1135	**1131**	1155
20×20^-2’	-	1282	1303	1284	**1262**	1280	1279
20×20^-3’	-	1320	1306	1297	1303	**1296**	1321
20×20^-4’	-	1268	1260	1267	1309	**1254**	1280
20×20^-5’	-	1219	**1187**	1266	1233	1198	1214
20×20^-6’	-	1192	1233	1192	1202	1219	**1190**
Total time(s)	-	120.9	122.6	145.8	145.2	172.6	174.4

After mapping open shop scheduling problem to complex scheduling network model,

$O_m||C_{max}$ can be transformed into reasonably arranging the node traversal order with the goal of traversing all nodes in the network as quickly as possible. All the proposed heuristic rules (LD, SCC, LPT, SPT, LTRPOM and LTRPAO) are used separately for solving $O_m||C_{max}$. The results are listed in the Table 1, where the "BKS" is the best-known solution, "$m \times m^{-k}$" means the kth instance for the OSSP with m jobs and m machines and "$m \times m^{-k}$" is the instance generated by removing a part of the operations on the basis of "$m \times m^{-k}$".

From the above table, it is obvious that for uniform open shop scheduling problem, the consistent scheduling results are achieved by these two network topological features based heuristic rules: LD and SCC. The reasons are not hard to comprehend. (1) At the beginning, topological differences among nodes in the complex scheduling network for uniform $O_m||C_{max}$ do not work; (2) In the subsequent scheduling process, the local attributes of the nodes to be selected are sufficiently similar, and LD is approximately equivalent to SCC, subject to (11).

$$C_i = \frac{2 * E_i}{k_i * (k_i - 1)} \tag{11}$$

In addition, heuristic rules involving local time attributes of nodes (LPT, SPT, LTRPAO and LTRPOM) obviously provide more heuristic information for uniform open shop scheduling problem. The results show that 16/18 best scheduling results are achieved by the heuristic rules involving local time attributes of nodes. Among them, LPT and LTRPAO perform best, the next does SPT, and the worst does LTRPOM. Both LPT and LTRPAO obtain 7/18 best scheduling schemes.

For non-uniform open shop scheduling problem, the running results of LD and SCC are no longer consistent, since different initial processing nodes are selected. Once the initial traversal nodes are different, the network will evolve in a significantly different direction. However, SCC becomes the best scheduling rule, and achieves 6/18 best scheduling schemes. It can be concluded that the effectiveness of heuristic rules based on the network topological features is greatly influenced by the non-uniformity of the initial complex network topology. The higher the non-uniformity, the better the scheduling effect, and vice versa. The performance of other heuristic rules is basically consistent with that of the uniform open shop scheduling problem. Specifically, LPT performs the best, followed by LTRPAO and SPT, and LTRPOM exhibits worst performance.

As outlined in Table 1, in terms of the computational speed, heuristic rules based on topological features have a slight advantage over the heuristic rules involving local time attributes. But all of them can meet the real-time requirements. In all experiments, the closest result to the best-known solution is obtained by LTRPAO rule on benchmark 15×15^{-1}, and the Gantt chart is shown in Figure 3.

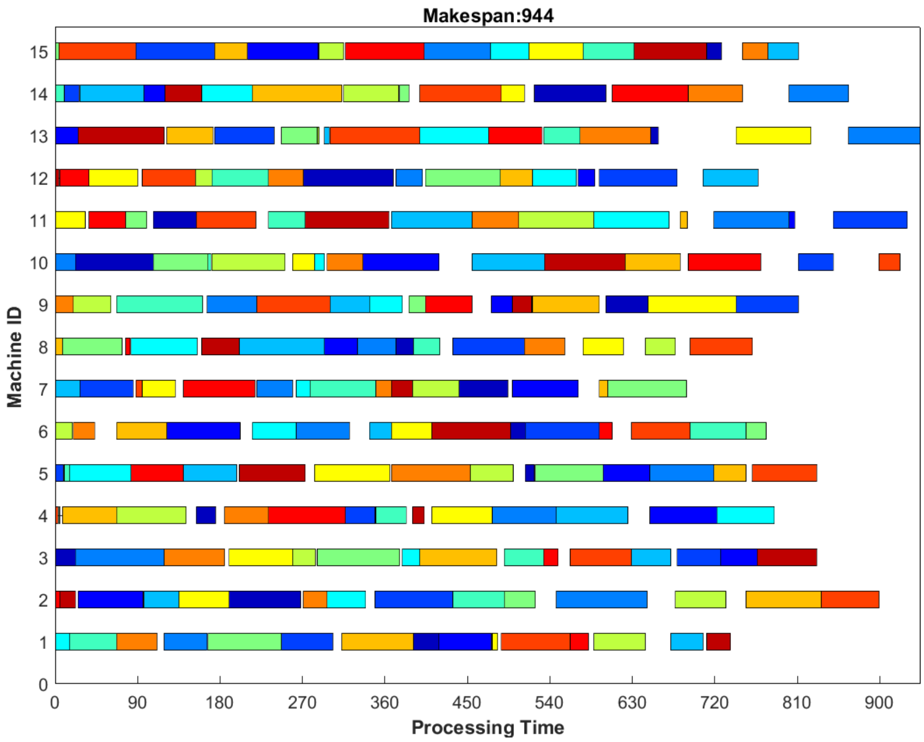

Figure 3. Gantt chart for benchmark 15 ×15^{-1} obtained by LTRPAO.

5.2. Case study II: Dynamic rule selection approach for $O_m|r_j|C_{max}$

The first experiment illustrates the effectiveness of proposed heuristic rules based on complex networks. To further study the feasibility and effect of the proposed dynamic rule selection approach for open shop scheduling problem with release dates, a carefully crafted set of test problems is generated for the lack of benchmark instances in the literature for $O_m|r_j|C_{max}$. In the benchmark instances, the processing time of each operation on the corresponding machine can vary within a range of 20–99, and the arrival time interval of two adjacent jobs can fluctuate within a range of 10–50. Based on the idea of dynamic rule selection approach, five new algorithms selected from different sets of heuristic rules are proposed, named DRSA1–DRSA5, respectively. The first one is selected from LPT, SPT, LTRPAO and LTRPOM, and the second to fifth algorithms are obtained by the first algorithm to remove separately LTRPOM, LTRPAO, SPT and LPT.

In Table 2, "m × m^{-k}" instance means the kth case for the OSSP with m jobs and m machines at the beginning, and then another m jobs will arrive at random. Besides, the lower boundary 'LB' can be calculated roughly by the following formula:

$$LB = \max(\max_{i \in Set1} \sum_{k=1}^{m} t_i^k, \max_{i \in Set2} \sum_{k=1}^{m} t_i^k + rt_i) \tag{12}$$

where Set1 denotes a set of jobs at the beginning, Set2 denotes a set of jobs arriving successively and m is the number of machines.

As seen in Table 2, for uniform open shop scheduling problem with release dates, LD and SCC can still achieve the same scheduling results. Besides, it is clear that LPT rule is still the best-performing scheduling rule, but LTRPOM rule defeats LTRPAO rule. Satisfyingly, DRSA1–DRSA5 achieve similar performance compared with other single heuristic rules on all the benchmarks, and obtains better schemes than any single heuristic rules on the 8/18 benchmarks. Specially, DRSA5 achieves better results than that of any single heuristic rule on six benchmarks (10×10^{-3}, 10×10^{-5}, 15×15^{-2}, 15×15^{-5}, 15×15^{-6} and 20×20^{-2}). And all the five dynamic rule selection approaches can obtain better results than that of any single heuristic rule on 15×15^{-5}. The Gantt chart of best approximate scheme achieved by DRSA5 on benchmark 15×15^{-5} is shown in Figure 4. Compared with Figure 3, it appears that the result in Figure 5 is much more compact at the beginning. The main reason is that there are many jobs that can be processed at the beginning, due to the dynamic arrival of jobs.

The experimental results adequately demonstrate the feasibility and necessity of the proposed complex network based dynamic rule selection approach.

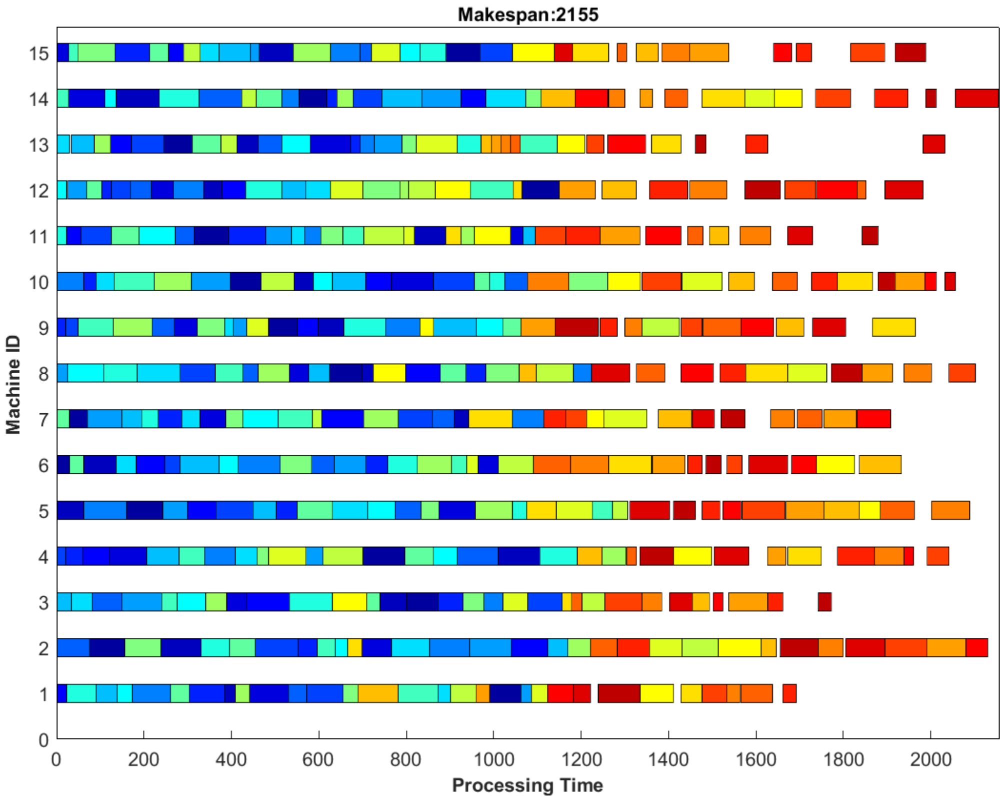

Figure 4. Gantt chart for benchmark ‘15×15^{-5}’ obtained by DRSA5.

Table 2. Scheduling results of proposed complex network based dynamic rule selection approach for $O_m|r_j|C_{max}$.

Benchmark	LB	LD	SCC	LPT	SPT	LTRPAO	LTRPOM	DRSA1	DRSA2	DRSA3	DRSA4	DRSA5
10×10^{-1}	984.3	**1382**	**1382**	1410	1409	1436	1397	1421	1434	1397	1405	1427
10×10^{-2}	933	1447	1447	1452	1489	1480	**1429**	1458	1453	1458	**1427**	1446
10×10^{-3}	949.6	1452	1452	1443	**1432**	1483	1461.4	1485	1442	**1421**	1453	**1427**
10×10^{-4}	1060	1588	1588	1569	1598	1570	**1516**	1548	1572	1548	1610	1598
10×10^{-5}	927	1418	1418	1436	1433	1418.6	**1404**	**1396.6**	1438.6	1408	**1396.6**	**1401**
10×10^{-6}	919.2	1429	1429	**1406**	1494.7	1421	1430	1454	1450	1425	1421	1430
15×15^{-1}	1471.9	2193	2193	2231	**2159**	2234	2217	2230	2196	2238	2204	2216
15×15^{-2}	1463.3	2225	2225	2206	**2188**	2256	2207	2255	2218	2255	2204	**2171**
15×15^{-3}	1342.5	**2030**	**2030**	2130	2101	2045	2100	2083	2065	2049	2076	2039
15×15^{-4}	1161.2	2083	2083	**2005**	2039	2026	2043	2064	2096	2039.1	2047	2029
15×15^{-5}	1379.1	**2194**	**2194**	2227	2204	2199	2199	**2162**	**2175**	**2162**	**2168.1**	**2155**
15×15^{-6}	1311.5	**2058**	**2058**	2085	2068	2117	2075	2088	**2053**	2088	2079	**2049**
20×20^{-1}	1912.6	2946	2946	**2892**	2927	2968	2948	2932	2944	2903	2904	2897
20×20^{-2}	1842.9	2890	2890	2879	**2877**	2925	2888	2910	**2867**	2904	2896	**2861**
20×20^{-3}	1797.4	2719	2719	**2690**	2723	2725	2737	2729	2714	2704	2725	2717
20×20^{-4}	1800.4	2764	2764	**2737**	2757	2764	2776	2762	2753.2	2771	2781	2787
20×20^{-5}	2001.9	2981	2981	2952	**2936**	2947	2961	2944	2990	**2933**	2949	2986
20×20^{-6}	1906.9	2906	2906	**2868**	2954	2957	2894	2898	2931	2909	3000	2973
Total time (s)	-	1819.2	1783.1	1893.8	1785.9	1901.3	1919.9	1808.1	1792.4	1796.4	1804.4	1804.4

6. Conclusion and future work

This study has dealt with open shop scheduling problems. Firstly, the framework of general scheduling object is built, under which most scheduling problems can be described, and open shop scheduling problem has no exception. Secondly, open shop scheduling problem is mapped to complex scheduling network model, in which one node denotes one operation of one job and one edge represents mutually exclusive timing constraint between two operations. By this means, OSSP can be transformed into reasonably arranging the node traversal order with the goal of traversing all nodes in the network as quickly as possible, on condition that each node has a traversal time, and only the disconnected nodes can be traversed simultaneously. Then from the perspective of decoupling complex scheduling networks, based on topological features, two heuristic rules (LD and SCC) are established. Given that local time attributes of nodes can also provide heuristic information, LPT and SPT are employed. Next, two heuristic rules (LTRPAO and LTRPOM) are developed to combine network topological features with local time attributes of nodes. Finally, an effective complex network based dynamic rule selection approach is proposed for open shop scheduling problem with release dates by switching the scheduling rules dynamically based on real-time production status.

For uniform $O_m||C_{max}$ and $O_m|r_j|C_{max}$, heuristic rules based on topological features obtain same scheduling results. Moreover, heuristic rules involving local time attributes of nodes (LPT, SPT, LTRPAO and LTRPOM) provide more heuristic information and achieve better scheduling results on the most benchmarks. For non-uniform $O_m||C_{max}$, LCC becomes the best scheduling rule and achieves 6/18 best scheduling results. Based on the idea of dynamic rule selection approach, DRSA1~DRSA5 are designed to change the scheduling rules dynamically based on real-time production status. They achieve similar performances compared with other single heuristic rules on all the benchmarks. Besides, they achieve better results than that of any single rule on the 8/18 benchmarks. The feasibility and necessity of the proposed complex networks based dynamic rule selection approach is confirmed. Future work is still needed to design more effective heuristic rules based on topological features or local time attributes of nodes for different scheduling objectives, and further study complex networks based dynamic rule selection mechanism for each scheduling decision scenario.

Acknowledgments

The authors would like to acknowledge financial supports of the National Science Foundation of China (No. 51775348, No. U1637211) and Shanghai Aerospace Science and Technology Innovation Fund (No. SAST2016048).

Conflict of Interest

The authors declare no conflict of interest.

References

1. Y. T. Leung, Handbook of Scheduling: Algorithms, Models, and Performance Analysis. CRC Press, Inc. 2004.
2. M. Pinedo, Scheduling: Theory, Algorithms, and Systems, 4th ed., Boston, MA: Springer, New York, 2012.
3. D. Mourtzis, Internet based collaboration in the manufacturing supply chain, *Cirp J. Manufact. Sci. Technol.*, **4** (2011), 296–304.
4. T. F. Gonzalez and S. Sahni, Open shop scheduling to minimize finish time, *J. ACM*, **23** (1976), 665–679.
5. D. Bai, Z. Zhang, Q. Zhang, et al., Open shop scheduling problem to minimize total weighted completion time, *Eng. Optimiz.*, **49** (2017), 98–112.
6. J. Blazewicz, W. Domschke and E. Pesch, The job shop scheduling problem: Conventional and new solution techniques, *Eur. J. Oper. Res.*, **93** (1996), 1–33.
7. R. L. Graham, E. L. Lawler, J. K. Lenstra, et al., Optimization and approximation in deterministic sequencing and scheduling: a survey, *Annals discrete mathematics* (1979), 287–326.
8. M. Pinedo, Scheduling: theory, algorithms, and systems. Prentice Hall, USA, 2002.
9. D. Bai and L. Tang, Open shop scheduling problem to minimize makespan with release dates, *Appl. Math. Model.*, **37** (2013), 2008–2015.
10. E. L. Lawler, J. K. Lenstra, A. H. Kan, et al., Minimizing maximum lateness in a two-machine open shop, *Math. Oper. Res.*, **6** (1981), 153–158.
11. R. Chen, W. Huang and G. Tang, Dense open-shop schedules with release times, *Theor. Comput. Sci.*, **407** (2008), 389–399.
12. P. Brucker, J. L. Hurink, B. Jurisch, et al. A branch & bound algorithm for the open-shop problem, *Discret Appl. Math.*, (1997), 43–59.
13. U. Dorndorf, P. Erwin and T. Phanhuy, Solving the open shop scheduling problem, *J. Sched.*, **4** (2001), 157–174.
14. C. Liaw, A hybrid genetic algorithm for the open shop scheduling problem, *Eur. J. Oper. Res.*, **124** (2000), 28–42.
15. F. Ahmadizar and M. H. Farahani, A novel hybrid genetic algorithm for the open shop scheduling problem, *Int. J. Adv. Manuf. Technol.*, **62** (2012), 775–787.
16. X. Li, L. Gao, Q. Pan, et al., An Effective Hybrid Genetic Algorithm and Variable Neighborhood Search for Integrated Process Planning and Scheduling in a Packaging Machine Workshop, *IEEE Trans. Syst. Man Cybern.*, (2018), 1–13.
17. X. Li, L. Gao, W. Wang, et al., Particle swarm optimization hybridized with genetic algorithm for uncertain integrated process planning and scheduling with interval processing time, *Comput. Ind. Eng.*, (2019).
18. X. Li, S. Xiao, C. Wang, et al., Mathematical modeling and a discrete artificial bee colony algorithm for the welding shop scheduling problem, *Memet. Comput.*, (2019), 1–19.
19. X. Li, C. Lu, L. Gao, et al., An Effective Multiobjective Algorithm for Energy-Efficient Scheduling in a Real-Life Welding Shop, *IEEE Trans. Ind. Inform.*, **14** (2018), 5400–5409.
20. A. Barabasi and R. Albert, Emergence of Scaling in Random Networks, *Science*, **286** (1999), 509–512.

21. D. J. Watts and S. H. Strogatz, Collective dynamics of ‘small-world’ networks, *Nature*, **393** (1998), 440–442.
22. Q. Xuan and T. J. Wu, Network model and heuristic scheduling rule designing method for complex open shop problems, *J. Zhejiang University Eng. Sci. Edition*, **45** (2011), 961–968.
23. Q. Xuan and T. J. Wu, Open shop complex scheduling network model and characteristic analysis, *J. Zhejiang University. Eng. Sci.*, **45** (2011), 589–595.
24. X. Li, Y. Yuan, W. Sun, et al. Bottleneck identification in job-shop based on network structure characteristic, *Comput. Integr. Manuf.*, **4** (2016), 23.
25. D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE T. Evol. Comput.*, **1** (1997), 67–82.
26. A. S. Manne, On the job-shop scheduling problem, *Oper. Res.*, **8** (1960), 219–223.
27. L. Lu, D. Chen, X. Ren, et al. Vital nodes identification in complex networks, *Phys. Rep. Rev. Sec. Phys. Lett.*, **650** (2016), 1–63.
28. T. Bian and Y. Deng, Identifying influential nodes in complex networks: A node information dimension approach, *Chaos*, **28** (2018).
29. E. D. Taillard, Benchmarks for basic scheduling problems, *Eur. J. Oper. Res.*, **64** (1993), 278–285.
30. G. C. Ciro, F. Dugardin, F. Yalaoui, et al., Open shop scheduling problem with a multi-skills resource constraint: a genetic algorithm and an ant colony optimisation approach, *Int. J. Prod. Res.*, **54** (2016), 4854–4881.
31. A. Clauset, C. R. Shalizi and M. E. J. Newman, Power-law distributions in empirical data, *SIAM Rev.*, **51** (2009), 661–703.
32. Y. S. Virkar, Power-Law Distributions and Binned Empirical data, *Ann. Appl. Stat.*, **8** (2014), 89–119.