An Algorithm for Solution of Scheduling Problem for Job Shop with Group Machining

D. I. Korovin[I], E. V. Romanova[I], S. R. Muminova[I], A. V. Osipov[I,III], E. S. Pleshakova[I,III], L. T. Chernyshov[I,II], S. T. Gataullin[III]

[I] Financial University under the Government of the Russian Federation, Moscow, Russia
[II] Moscow Aviation Institute (National Research University), Moscow, Russia
[III] Moscow Technical University of Communications and Informatics, Moscow, Russia

Abstract: The paper presents the new algorithm for solving one problem from the scheduling theory. The method is based on the principle of graph coloring and allows simultaneous processing of several details in one workplace. The problems of scheduling theory are briefly analyzed and the place of the given problem is determined within the general classification of problems. The scheduling algorithm and the program on the basis of it have been developed to solve this problem for various optimality criteria. Two versions of the program have been implemented. The first one follows directly the data structures and the sequence of actions of the graph coloring method. In the second version, the structures of the linear representation of the graph are used, as well as multi-step operations are introduced, which made it possible to increase the efficiency of the scheduling algorithm. The time characteristics of the program execution on a different number of details for two versions of the program are given. The prospects for the development of the program and the scope of its application are discussed and could be rather wide, from agribusiness, such as optimizing the production of meat products, to manufacturing enterprises with a significant range of product line.

Keywords: graph theory; scheduling theory; graph coloring; scheduling algorithm.

DOI 10.14357/20718632230112

1. Introduction

Automation of manufacturing processes significantly impacts on business success potential as it favors decrease of costs and optimization of resource consumption. Mathematical modeling is a powerful tool in robotics [1-4], education [5] and financial technologies [6, 7], transportation [8] and even in sport [9]. Scheduling theory as a part of mathematical modeling has become an area of numerous researches aimed at increasing efficiency of industrial plants producing some articles or orders. The problems of planning and organization of “smart” production employing technologies of artificial intelligence are described in the latest studies [10-13].

The authors of the given paper analyze a special case of job shop production: there are some workplaces at which several details can be processed simultaneously. It is so-called a job shop with *group machining*. Furnaces are the example of the equipment used in group machining. The furnaces are characterized by different parameters (temperature, heating time, etc.) and can be used for different types of details. Compared to common workplaces, it is impractical to use one furnace for one detail, therefore a set of details to be charged in it. Since different details follow different processing sequences, a sequential queue of details is formed for each furnace. As soon as the necessary quantity of details is accumulated in the queue, the furnace is launched.

Such type of planning problems is known in other fields. For example, in the course of time scheduling at universities, lecture classrooms can be regarded as machines of group processing for several student groups. The difference is that there is a strict order of disciplines and classrooms to be passed by the student groups. If we consider the problem of distribution of virtual machines and operation flows in cloud services, computational nodes performing several processes can be analyzed. But in our case the principal factor is the physical nature of the machines, comparing to virtual ones.

To resolve scheduling problem of job shop production many approaches are used, some of them are described below in section 1.1. The authors did not find the examples of approaches designed for group machining. At first, to solve the considered problem, the authors applied colored graph method as it was done in similar works [14, 15]. Its modification allowed assigning machines of group processing. However, the efficiency of the program executing this method was not proved. Thus, the aim of this study was to enhance the scheduling algorithm.

1.1. The overview of the problems

Generally, the scheduling theory [16] formulates a planning problem in the following way: a set of jobs (requirements) J = {J_1, …, Jn} with specific characteristics (requirement processing time (simple case), requirement processing cost, requirement arrival time, schedule requirement processing termination date) is given. A set of equipment (machines) M = {M_1, …, Mm} on which the requirements are to be processed in a specific order is given. It is necessary to make a schedule which minimizes job processing, job costs, etc. The schedule determines which machine processes a requirement at the given time.

The mode of equipment operation defines four main classes of problems: 1) Open shop – the sequence of processing on the equipment is arbitrary, 2) Job shop – for each requirement a processing itinerary is given, 3) Flow shop – all machines are put in a specific order and each job passes all machines in this order, 4) A job with schedule time – for each job an arrival time, a processing time and job processing termination schedule are given. Additionally, the problems are classified by the opportunity to break job processing. In each class of problems different optimization criteria are considered. The problem considered in the paper refers to job shop without breaking.

The solution of «Job shop» problem, even with one machine without breaking, is a permutation of jobs and for arbitrary objective function these problems are NP-complete. This result was obtained in one of the early publications in 1976 [17-20] in which number of machines was three.

Full classification of statements of the problems and publications concerning scheduling theory is available on website [21]. Classification considers the following types of problems: simultaneously operating identical machines (type P); machines operating with specific rates (type Q); machines with fixed processing time for some jobs (type R); machines for which job processing sequence is arbitrary (type O); flow shop in which each job is performed on fixed sequence of machines (type F) and, finally, a workshop in which requirements consist of a sequence of operations performed with different machines (type J). As a rule, the scope of planning problem is to minimize total processing time. For each type of problems cases with different numbers of machines are regarded independently. The problems can differ by addition of varying constraints: priority ratio, grouping of requirements into packages, processing time limits, etc.

The problem considered in the paper refers to type J. At this, there is an additionally imposed condition: possible presence of machines which simultaneously process several jobs (requirements). Number of such requirements can be set by value range. The classification in [22] does not contain such condition.

1.2. Formulation of the problem

We considered the following problem. It is necessary to design an scheduling algorithm to supply details in experimental job shop of thermal-mechanical processing, in which there are N1 workplaces with N2 operation types (each operation is either a machine tool, or a furnace, or a control operation position). Each operation is characterized by quantitative indicators of simultaneous detail processing. This is critical for the furnaces which can hold varying amount of details of one type $a_{\text{(furnace-operation number)(type of detail)}}$, for many operations

$a_{ij} = 1.$

An order is a set V= (V_1, V_2, …, Vn) of n details to be produced according to technological plans. We present them as a table with these attributes:

(operation number; standard execution time; assigned equipment group (number): limiting boxing-in number $a_{(furnace\text{-}operation\ number)(\ type\ of\ detail)}$).

Sequence of operations for one detail is its technological plan. The details of same type have the same technological plan. If a machine processes a detail of one type and after that it processes a detail of another type, reconfiguration (resetting) of the machine can be needed. Time required to reset the operation mode of machine s from detail i to detail j is marked as t_{sij}.

Nontrivial problem, even when values Vi, N1 and N2 are not so significant, is to design at least one noncontradictory schedule. We propose the algorithm of "smart" search of designed noncontradictory schedules, as a result of which the variants with optimal indicators are selected:

F1 – total time of order execution,

F2 – idle-time of equipment,

F3 – number of equipment reconfigurations.

We assume that all operations are performed in one-time step, and if a technological process requires much time, we divide it into several one-time step operations. The task is to develop an algorithm to create all possible workplace schedules and to choose the optimal one.

2. Materials and Methods

2.1 Mathematical Methods to be Used in Algorithm

We will apply some methods used in graph theory [23]. A graph is called vertex K-colorable graph, if its vertices can be colored in K colors in a way that no two neighboring vertices are of the same color. Minimum number K, for which graph G is a vertex K-colorable graph, is called chromatic number of the graph. The graph can be colored according to this algorithm:

1. Find number of bounds of all vertices in the map (vertex degree).
2. Consider the vertices in descending order of number of bounds.
3. Apply color 1. Consider vertices sequentially. If a selected vertex is not colored and there is no bound with the vertices with the color 1, then it will be applied color 1.
4. If all vertices have been passed but not colored, repeat step 3 using another color unless all vertices get colored.

This algorithm can be used to make a schedule. The vertices with bounds are the activities which cannot be performed simultaneously since the same tasks or jobs are used for the same object.

In our case, an additional condition emerges: some activities cannot be done if other jobs have not been performed with a batch of details which is followed by the given batch of details. For this case we introduce a term "directed link": two angular points A and B are connected with directed link A□B, if an event B cannot happen before A. It is shown by the arrowed line and the initial graph turns into oriented graph.

Algorithm of graph coloring is carried out in the following way. One need to remember that the vertices are the operations of the same color that should be performed at the same time.

1. Select a vertex without an arrowed line (that means that its in-degree is null) and apply color 1 to it.
2. Go from vertex to vertex in a predefined order.
3. Start applying the given color. If the considered vertex is not colored and is not connected to the vertices of the given color and its in-degree is null, then it will be applied the given color. Delete (if available) all arrowed lines, coming from this vertex.
4. If all vertices are shown but not colored, repeat step 3 using another color unless all vertices get colored.

The order of detail processing operation must be regulated by the following rule: each operation must not be followed by any other operation performed earlier according to technological plans.

Thus, having colored the graph, we can interpret the algorithm. Since each vertex is a necessary operation for a certain detail, performed on the machine, a set of vertices of the same color is a set of operations performed simultaneously. Formed sequence of colors clearly determines the order of execution for operations with the given details on given machines.

Known algorithms contain a problem of simultaneous detail processing. Our model resolved this problem in the following way.

The imposed problem provides that the same details are simultaneously processed when being kilned in the furnaces. Number of details which must be simultaneously processed (boxing-in) is predetermined.

In our algorithm such furnace is a sequence of vertices sequentially bounded into network by arches. Number of vertices in the network corresponds to processing time steps for details of the given type. The first vertex is characterized with n inbound edges, where n-boxing-in, the last network's vertex – n-outgoing edges – already processed details.

This mechanism allows launching furnace only when the last boxing-in element is loaded and force transition to further processing of all n details as the vertex will not be deleted unless the last outgoing edge is "erased".

2.2. A Scheduling Algorithm

To solve a scheduling problem for a program employing described algorithm has been written in C# in MS Visual Studio. The first release of the code – we call it an initial algorithm - contained data structure and sequence of activities that accurately corresponded to the algorithm. Thus, the graph of one-time step operations was presented as a vertex incidence matrix, search of the vertices was performed by exhaustive search and the loop nesting was four in the program. Tests on different datasets resulted in improvements which shortened time processing. Firstly, one-time step operations were replaced with multi-step ones. Secondly, coloring included not only step number but also machine number among machines of the same type. Additional data structures were implemented, that made possible to decrease the loop nesting from four to two when calculating one schedule.

Advanced performance algorithm which we will call it "smart" or "enhanced" algorithm, is described further. The examples are input and output data of the program.

2.3. Statement of the Problem

There is a set of types of machines $M = \{M_1, \ldots, M_m\}$ used in detail processing. Job shop consists of machines of different types and the total amount is $Ns = ks_1 + \ldots + ks_m$, where ks_i – number of machines of type M_i. Each M_i type machine can process N_i details. For the machines that process more than one detail (furnaces) processing time T_i in time steps (time step is a fixed period of time) is specified. These machines process many details of different types $D = \{D_1, \ldots, D_n\}$. Total number of details is $N_d = kd_1 + \ldots + kd_n$, where kd_j is a number of details of type D_j.

For each detail type D_j a sequence of operations (processing plan) performed on the machines of the specified type is given: $P_j = ((M_{j1}, dt_{j1}), \ldots, (M_{jq}, dt_{jq}))$, where M_{ji} is a type of the machine tool, dtji is a number of time steps needed for detail processing. For the operation executed on furnace the number of steps dt_{ji} must be the same as duration T_{ji} of the machine.

The problem to plan processing of all details is to schedule performance of each machine tool. In each step from 1 to t_{max} (total duration of processing of all details) each machine either does not operate or processes some detail. Thus, for each machine mi it is necessary to design an order $Plan_i = [(d_{i1}, t_{i1}), (d_{i2}, t_{i2}), \ldots]$, where d_{ij} is a detail processed in the step t_{ij}, at this $t_{i1} < t_{i2} < \ldots t_{max}$ and motion path (transfer of the detail from one machine to another one) of each detail corresponds to processing plan for its type and in each step the number of employed machines of M_i type is not more than ks_i.

The sets of machines and the sets of details to be produced will be named as job shop configuration. To make description of configurations convenient for program, types of machines and details can be assigned names.

Example S1 of input data of program – description of job shop configuration:

S1: m1, P(2, 3), m2, m3(2)
d1,1 (m1, P, m2/2)
d2,1 (m2, m3, P, m2)
d3,2 (m3, m1, P, m1, m2)
d4,2 (m1, P, m3, m1)

The first line contains a list of types of machines and quantity of them. There is one machine of types m1, P, m2, and there are two machines of m3 type. The machine P (furnace) processes 2 details simultaneously in 3 time steps. Further, types of details, their quantity and processing plans are described. There is one detail of types d1 and d2, and there are two details of types d3 and d4. All operations on the machines which process one detail are performed in one-time step, excluding m2/2, which is performed in 2 time steps.

```
        1    2    3    4    5    6    7    8    9   10   11   12   13   14
m1/1 d1/1 d3/1 d3/2 d4/1 d4/2                d3/1 d3/2           d4/1 d4/2
P/1            d1/1-d1/1-d1/1 d3/1-d3/1-d3/1 d4/1-d4/1-d4/1
P/1            d2/1-d2/1-d2/1 d3/2-d3/2-d3/2 d4/2-d4/2-d4/2
m2/1 d2/1                     d1/1-d1/1 d2/1      d3/1 d3/2
m3/1 d3/1 d2/1                                              d4/1
m3/2 d3/2                                                   d4/2
```

Fig. 1. Result of program execution: schedules of machines

The result of program execution is presented in the table (Fig.1), in which lines contain schedules of machines and the column heads are step numbers.

For example, schedule of the machine $m1_1$ will be: $Plan_1$ = [($d1_1$,1), ($d2_1$,2), ($d3_2$,3), ($d4_1$,4), ($d4_2$,5), ($d3_1$,9), ($d2_2$,10), ($d4_1$,13), ($d4_2$,14)]. It is also possible to track detail processing in the table. For example, for $d1_1$ operation sequence (marked **in bold** in the table) is [($m1_1$, 1, 1), (P, 3, 2), ($m2_1$, 6, 2)]. Each operation is assigned with a machine, number of time step and duration of processing. Sequence of operation corresponds to processing plan (m1, P, m2/2) for a detail of type d1.

2.4. Representation of Data

Each detail is presented by the list sD = (td, nd, ti, p), where td – type of the detail, nd – detail number, ti – step number, p – index in operation sequence, from which operation sequence starts. Step number ti can take values from 0 to dt and it determines how many time steps the machine processed. Operation sequence for processing of details of certain type is specified by the set of the lists Op = [(st, kst, col, dt), ...], where st – type of the machine, kst – machine number among machine of type st, col – step number in common processing, dt – number of steps to be performed by the machine. Order of the operations in this sequence corresponds to that in ordered set of detail types, i.e. the first position belongs to the processing plans for details of type D_1, then the processing plans for details of type D_2 follow, etc. Operation sequence for each detail mi with the set sD_i is a subsequence in Op, with indices from $sD_{i.p}$ to $sD_{i+1 \cdot p-1}$.

Example S2 of configuration with three machines and three details:

S2: M1, M2(2)

D1, 2 (M1, M2/2)

D2, 1 (M2, M1/3, M2)

Representation of data at the start of algorithm execution:

mD = [(D1,1,0,0),(D1,2,0,2),(D2,1,0,4)]

Op = [(M1,1,0,1), (M2,2,0,2), (M1,1,0,1), (M2,2,0,2), (M2,2,0,1), (M1,1,0,3), (M2,2,0,1)]

In fact, Op are the vertices of «not colored» graph. At first, the filed «kst» is placed in quantity of machine tools of type «st». After the schedule has been obtained, they became «colored», i.e. time step numbers are placed instead of nulls in the fields «col». The fields «kst» are placed into numerical order of the machine among the machines of the same type. In the given example the vertices will take the following values:

Op = ((M1,1,1,1), (M2,1,2,2), (M1,1,2,1), (M2,2,3,2), (M2,1,1,1), (M1,1,3,3), (M2,1,6,1))

On the basis of obtained coloring the schedule for each machine is designed:

```
M1/1   D1/1 D1/2 D2/1-D2/1-D2/1
M2/1   D2/1 D1/1-D1/1       D2/1
M2/2        D1/2-D1/2
```

The fields «ti» and «p» in the list of details change during algorithm execution. The field «ti» indicates the number of steps performed by the machine tool in multi-step operation, and «pi» indicates the list of the corresponding operation in sequence «Op».

2.5. The Algorithm of Planning

The algorithm of planning for «tcol» = 1, 2, … step-wise records values of time step and machine number into the fileds «col» and «kst» of the lists from the set «Op» accordingly, that defines which details and on which machines will be processed in this time step.

Flow diagram of the algorithm is presented on Fig. 2. The notation of flow diagram: Nd – total number of details, Dn – number of processed details, tcol – time step number (the given color) in common processing flow, i – sequence number of the detail, ti – step number in processing of detail i, kst – machine number in processing of the detail i, col – step number in processing flow for detail i.

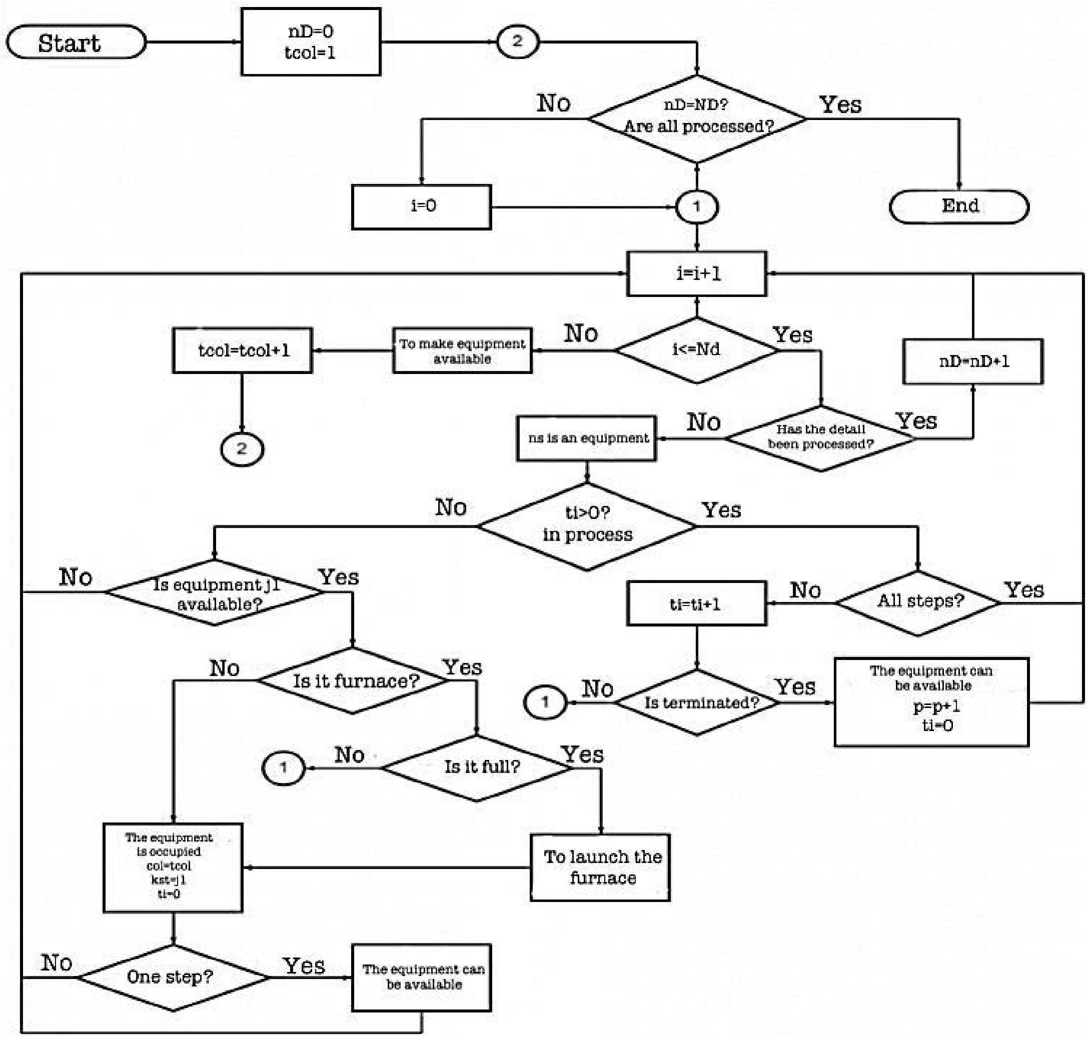

Fig. 2. Flow diagram of scheduling

Outer loop by time step tcol assigns colors to details (col = tcol) in the case when there is a machine which is available for them (kst = j1). Outer loop is terminated if all details have been processed (Dn = Nd). Internal loop by details checks if the detail is processed (ti = td), not processed (ti = 0) or is expecting to be put into the furnace (ti = -1). If the detail has not been processed yet, an available machine is searched. That is done due to the set which contains availability of machines. In the case of success, the index of machine j1 is recorded in the field ks of the machine tool i. Machine number ns is determined by the present operation.

If the machine is operating (ti > 0), there is a check whether it is the last time step. If it is the last one, a transfer to the next operation takes place (indicator p increases by 1), and the machine is marked as “available”. Marking is performed in the backing array. Real status of availability is assigned after all details have been checked, since this machine may be occupied by processing other detail in the given time step.

Furnace processing is shown as a procedure on the flow diagram. There are also additional structures that are used to monitor if the furnace is full. If the furnace is not full, the detail is in expectation mode (ti = -1), and when the number of details is sufficient, the details are marked correspondingly, and the furnace becomes available.

2.6. Search of optimal schedule

The given algorithm builds only one variant of the schedule. The result significantly depends on the order of description of machines as well as description of detail types. That can be seen even in very simple examples:

```
d2,1 (s1/2,s2)
d1,1 (s1/3,s2/2)
          1   2   3   4   5   6   7
s1/1    d2/1-d2/1 d1/1-d1/1-d1/1
s2/1            d2/1          d1/1-d1/1

d1,1 (s1/3,s2/2)
d2,1 (s1/2,s2)
          1   2   3   4   5   6
s1/1    d1/1-d1/1-d1/1 d2/1-d2/1
s2/1                d1/1-d1/1 d2/1
```

The program can perform exhaustive search of permutation of detail types and calculate some indicators. The given variant demonstrates how the program calculates total number of time steps (T) to process all details, number of time steps of idle time (P), number of reconfigurations (N) and average weighted indicator (T+P+N). Thus, in the example of configuration S2 we have T=6, P=7, N=2. In the course of search, a configuration for which one of these indicators is minimum is found.

In each step of outer loop of the algorithm selection of proper machines is performed in the order as they are described in configuration. That means the machines at the beginning of the list are of priority. It is obvious, that sometimes it is not the best option. Therefore, permutation of types of machines in the list should be provided. We call it «inner search», opposed to rearrangement of types of details, which is «outer search».

Even if the number of machines and details is rather small, the number of permutations can be extremely large. That is why there are two modes of program execution: 1) indication of initial permutation of maximum number of permutation; 2) random selection of permutation.

2.7. Track of time needed for reconfiguration

If the time needed for reconfiguration is less than one-time step, then obtained optimal schedules will be acceptable. Otherwise, that can have significant impact on the total processing time. Number of reconfiguration time steps can vary in dependence on the type of details. In the given algorithm that can be provided by adding some time steps (value ti) to time steps of detail processing, if the machines on the previous time step processed the detail of another type. For this one needs to add function Fn(ts, td1, td2) which returns number of time steps required for the machine tool of the type ts to shift from processing of details td1 to processing of details td2.

2.8. Track of time needed for reconfiguration

If the time needed for reconfiguration is less than one-time step, then obtained optimal schedules will be acceptable. Otherwise, that can have significant impact on the total processing time. Number of reconfiguration time steps can vary in dependence on the type of details. In the given algorithm that can be provided by adding some time steps (value ti) to time steps of detail processing, if the machines on the previous time step processed the detail of another type. For this one needs to add function Fn(ts, td1, td2) which returns number of time steps required for the machine tool of the type ts to shift from processing of details td1 to processing of details td2.

2.9. Furnace problem

The real time of furnace processing can vary in the range from minimum to maximum values. The charge – quantity of simultaneously processed details – can also vary. In principle, it is of small importance for searching optimal schedule. In the given option of the program both parameters are fixed. That results in a problem of furnace launching when the furnace has not been full charged yet, and there are already no unprocessed details. The launch of the furnace with the last portion could happen in any case of charge. To provide this condition, the algorithm should be enhanced.

Another problem arises if several furnaces are available and different details are to be processed in several furnaces in different time. We consider simple configuration:

S2: P1(2, 3), P2(2,4)
d1,1 (P1, P2)
d2,1 (P2, P1)

In the first step the detail d1 gets into line to the furnace P1 and the furnace will be waiting the second detail. Similarly, the detail d2 gets into line to the furnace P2, but this furnace cannot be launched as the first detail is in waiting mode. Due to the same reason the furnace P1 also cannot be launched. The common processing flow will be suspended: it is a dead lock situation. This problem is similar to that in operating system theory, thoroughly described in [24]. This theory uses the following terms: the details are processes and the machines are resources. One of the simple solutions is to allow processes to use resources strictly in one order. The only question that remains is whether this solution is acceptable for job shop configurations.

3. Evaluation of the Algorithm

The initial algorithm was based on the theory of colored graphs. The methods employing colored graphs can be applied for solving various practical tasks. But these methods were not efficient for our problem. To compare program performance with the initial and enhanced algorithms, a computational experiment was made on several configurations with different numbers of details. The data on machines and details describe the real job shop in simplified option.

The data for calculation are prepared in a separate program, in which input data are given in tables. For example, one needs to enter the code (type) of the detail, its name and quantity (Table 1). There are 66 types of details. It is also necessary to enter types and quantity of machines (they are coded with one or two-symbol codes) and processing plans.

As a result, the file with configuration is created:

S: G2,G3,P6,P4,P1,P2,PP,P0,R6,R0,R2,R8,R5,R2,K,N,PX,X8
1,2 (N,P4,P4,PX,PX,N,P4,R5,PX,PX,PX,N,P4,PX,PX,PX)
2,2 (N,P4,R5,PX,PX,PX,PX,N,P4,PX,PX,PX)
. . .

The first line contains codes of machines (18 of real 30). Then plan of processing for each type of details is given. The number of details of each type is 2. Since the number of operations in different plans is from 7 to 23, the total number of operations is 1112. The Table 2 contains the results of the computational experiment.

One can witness exponential growth in the initial algorithms, and the linear one in the enhanced algorithm. It is also evident the time consumption growth accompanied with increasing number of details. The number of permutations by K types of details is equal to K! (10! = 3628800, 20! ~= 2.4×10^{18}, 30! = 2.6×10^{32}, …). It is already at K>10 when the time of exhaustive search exceeds the capacities of up-to-date PC. One of the ways to empower the computational capabilities is to use cloud computing [25,26] The table 2 contains time parameters for K=2,4 and 8.

4. Discussion

One can enhance the algorithm in various ways:

1. To introduce aggregated operations, i.e. to regard a fixed sequence of operations applied for different types of details as one operation.

Table 1. Number and types of details

1	020607195000	fixture	148
2	020815098000	nozzle	40
3	021100029000	nut ring	30
	…		

Table 2. Performance of initial and enhanced algorithms

Number of detail	Size of matrix in initial algorithm	Calculation time for **one** schedule	Calculation time for **100** schedules in enhanced algorithm
2	2224 x 2224	33 sec	25 sec
4	4448 x 4448	2.5 min	54 sec
8	8896 x 8896	42.2 min	2 min 21 sec

2. To group details of the same type, i.e. to calculate processing of them as one “aggregated” detail.

3. To fix some subsequence in which permutations will not be made in the list of machines.

In general, many factors depend on the configuration of the selected job shop and, definitely, algorithm featuring should be done together with specialists who are aware of technological specifics.

5. Conclusions

The authors’ contribution to the theory of job-shop scheduling is the scheduling algorithm aimed at scheduling detail supply which partially solved the *original* problem: the problem with imposed condition of simultaneous detail processing in one workplace. The final solution of the problem is likely to be in the area of high-power computers. However, the described scheduling algorithm comprises a set of approaches which make it possible to significantly decrease consumption in computation capacities. The prospects for the development of the program and the scope of its application are discussed and could be rather wide, from agribusiness, such as optimizing the production of meat products, to manufacturing enterprises with a significant range of product line.

It is a complex problem concerning performance of the plans which allow simultaneous processing of several details in one workplace, application of several identical machines and general principles to resolve such problems do not exist.

Developed scheduling algorithm and program resolved the imposed problem of scheduling plan of job shop with the machines that can simultaneously process several details. Conducted numerical experiments proved possibility to implement processes with limited number of jobs into work flow organization.

Further research devoted to scheduling algorithm enhancing is assumed to determine concrete conditions of its application in technological processes.

References

1. Kochkarov, R. Multicriteria Optimization Problem on Prefractal Graph. Mathematics 2022, 10, 930. Https://doi.org/10.3390/math10060930.
2. Kochkarov, R. Research of NP-Complete Problems in the Class of Prefractal Graphs. Mathematics 2021, 9, 2764. Https://doi.org/10.3390/math9212764.
3. Dogadina, E.P.; Smirnov, M.V.; Osipov, A.V.; Suvorov, S.V. Evaluation of the forms of education of high school students using a hybrid model based on various optimization methods and a neural network. Informatics 2021, 8, 46.
4. Osipov, A.; Filimonov, A.; Suvorov S. Applying Machine Learning Techniques to Identify Damaged Potatoes. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). In Proceedings of the LNCS, 20th International Conference on Artificial Intelligence and Soft Computing, ICAISC 2021, Virtual, Online, 21-23 June 2021 DOI 10.1007/978-3-030-87986-0_17.
5. Pavlyutin, M.; Samoyavcheva, M.; Kochkarov, R.; Pleshakova, E.; Korchagin, S.; Gataullin, T.; Nikitin, P.; Hidirova, M. COVID-19 Spread Forecasting, Mathematical Methods vs. Machine Learning, Moscow Case. Mathematics 2022, 10, 195. Https://doi.org/10.3390/math10020195
6. Soloviev, V. Fintech Ecosystem in Russia. In Proceedings of the 2018 11th International Conference; Management of Large-Scale System Development, MLSD, Moscow, Russia, 1–3 October 2018. 10.13187/mlsd.2018.855180810.
7. Ivanyuk, V. Formulating the concept of an investment strategy adaptable to changes in the market situation. Economies 2021, 9, 95. 10.3390/economies9030095
8. Andriyanov, N.; Khasanshin, I.; Utkin, D.; Gataullin, T.; Ignar, S.; Shumaev, V.; Soloviev, V. Intelligent System for Estimation of the Spatial Position of Apples Based on yolov3 and Real Sense Depth Camera D415. Symmetry 2022, 14, 148.
9. Andriyanov, N.A., Dementiev, V.E., Tashlinskiy, A.G. Detection of objects in the images: from likelihood relationships toward scalable and efficient neural networks. Computer Optics 2022, Vol. 46(1). DOI: 10.18287/2412-6179-CO-922.
10. Wu, Kan; Shen, Yichi .A unified view on planning, scheduling and dispatching for a factory. Advanced Engineering Informatics 2020, 46, 101188. 10.1016/j.aei.2020.101188.
11. Zhou, Tong; Tang, Dunbing; Zhu, Haihua; Zhang, Zequn. Multi-agent reinforcement learning for online scheduling in smart factories. Robotics and Computer-Integrated Manufacturing 2021, 72, 102202. 10.1016/j.rcim.2021.102202.
12. Luo, Qiang; Deng, Qianwang; Gong, Guiliang; Zhang, Like; Han, Wenwu; Li, Kexin. An efficient memetic algorithm for distributed flexible job shop scheduling problem with transfers. Expert Systems with Applications 2020, 160, 113721. 10.1016/j.eswa.2020.113721.
13. Li, Haoran; Li, Xinyu; Gao, Liang. A discrete artificial bee colony algorithm for the distributed heterogeneous no-wait flowshop scheduling problem. Applied Soft Computing 2021, 100, 106946. 10.1016/j.asoc.2020.106946.
14. Zhou, Bin; Bao, Jinsong; Li, Jie; Lu, Yuqian; Liu, Tianyuan; Zhang, Qiwan. A novel knowledge graph-based optimization approach for resource allocation in discrete manufacturing workshops. Robotics and Computer-

Integrated Manufacturing 2021, 71, 102160. 10.1016/j.rcim.2021.102160.
15. Lu, Houjun; Wang, Sai. A study on multi-ASC scheduling method of automated container terminals based on graph theory. Computers & Industrial Engineering 2019, 129, 404-416. 10.1016/j.cie.2019.01.050.
16. Garey, M.R.; Johnson, D.S.; Sethi, Ravi. The Complexity of Flowshop and Jobshop Scheduling. Mathematics of Operations Research 1976, 1, 2, 117-129.
17. The Scheduling Zoo. Available online: http://www-desir.lip6.fr/~durrc/query/ (accessed 25 November 2021)
18. Korovin, D.I. Logical principles in organization of production. Ivanovo State University: Ivanovo, 2006.
19. Shaw, A. The Logical Design of Operating System. PRENTICE-HALL, INC. ENGLEWOOD CLIFFS: N.J. 1974.
20. Liu, Q.; Shao, X.; Yang, J.-P.; Zhang, J.-S. Multiscale modeling and collaborative manufacturing for steelmaking plants. Chinese Journal of Engineering 2021, 43 (12), 1698-1712.
21. Wang,B.; Liu, F.; Lin, W.; Ma, Z.; Xu, D. Energy-efficient collaborative optimization for VM scheduling in cloud computing. Computer Networks 2021, 201, № 108565.
22. Unsal, O. An extended formulation of moldable task scheduling problem and its application to quay crane assignments. Expert Systems with Applications 2021, 185, № 115617.
23. Lin, Y.; Miao, S.; Yang, W.; Yin, B.; Tu, Q.; Ye, C. Day-ahead optimal scheduling strategy of virtual power plant for environment with multiple uncertainties. Electric Power Automation Equipment 2021, 41 (12), 143-150.
24. Zheng, X.; Liang, S.; Xiong, X. A hardware/software partitioning method based on graph convolution network. Design Automation for Embedded Systems 2021, 25 (4), 325-351.
25. Huang, Y.: Xu, H.; Gao, H.: Ma, X.; Hussein, W. SSUR: An Approach to Optimizing Virtual Machine Allocation Strategy Based on User Requirements for Cloud Data Center. IEEE Transactions on Green Communications and Networking 2021, 5(2), 670-681. DOI: 10.1109/TGCN.2021.3067374.
26. Ma, X.; Xu, H.; Gao, H.; Bian M. Real-time Multiple-Workflow Scheduling in Cloud Environments. IEEE Transactions on Network and Service Management(TNSM) 2021, 18(4), 4002-4018. DOI: 10.1109/TNSM.2021.3125395.

Korovin Dmitriy I. Department of Data Analysis and Machine Learning, Financial University under the Government of the Russian Federation, Moscow, Russia. Professor. Doctor of Economic Sciences. Number of publications: 57. Research interests: mathematical modeling, machine learning, graph analytics, e-mail: DIKorovin@fa.ru

Romanova Ekaterina V. Department of Data Analysis and Machine Learning, Financial University under the Government of the Russian Federation, Moscow, Russia. Associate Professor, Candidate of Physical and Mathematical Sciences. Number of publications: 25. Research interests: mathematical modeling, machine learning, NLP, e-mail: EkVRomanova@fa.ru

Muminova Svetlana R. Department of Data Analysis and Machine Learning, Financial University under the Government of the Russian Federation, Moscow, Russia. Associate Professor, candidate of technical sciences. Number of publications: 15. Research interests: mathematical modeling, machine learning, e-mail: SRMuminova@fa.ru

Osipov Alexey V. Department of Data Analysis and Machine Learning, Financial University under the Government of the Russian Federation, Moscow, Russia. Moscow Technical University of Communications and Informatics, Moscow, Russian Federation. Candidate of Physical and Mathematical Sciences. Number of publications: 20. Research interests: information security, artificial intelligence, machine learning. E-mail: avosipov@fa.ru.

Pleshakova Ekaterina S. Department of Data Analysis and Machine Learning, Financial University under the Government of the Russian Federation, Moscow, Russia. Leading Researcher Moscow Technical University of Communications and Informatics, Moscow, Russian Federation. Candidate of Technical Sciences. Number of publications: 46. Research interests: information security, machine learning, e-mail: espleshakova@fa.ru

Chernyshov Lev N. Department of Data Analysis and Machine Learning, Financial University under the Government of the Russian Federation, Moscow, Russia. Associate Professor, Candidate of Physical and Mathematical Sciences. Number of publications: 26. Research interests: mathematical modeling, machine learning, e-mail: LNCHernyshov@fa.ru

Gataullin Sergey T. Dean of the Faculty of Digital Economy and Mass Communications Candidate of Economic Sciences. Number of publications: 51. Research interests: mathematical methods of decision making, economic and mathematical modeling, information security, machine learning, artificial intelligence, e-mail: stgataullin@fa.ru