Open job shop scheduling via enumerative combinatorics

Daniela I. Borissova and Ivan C. Mustakerov

***Abstract*—**The paper presents using of enumerative combinatorics for open job shop scheduling problems. Two approaches are described – one approach aims to determine a schedule that minimizes the total makespan by solution of a single optimization task and other that is based on solving in parallel of a number of optimization tasks. For the goal, a combinatorial optimization modeling is described and corresponding algorithm is proposed. The described approach of enumerative combinatorics for optimal job shop scheduling is numerically tested for real job shop scheduling problem.

***Keywords*—**Open job shop scheduling, mathematical programming, minimal makespan, enumerative combinatorics.

I. Introduction

THE scheduling is a key factor for manufacturing productivity. Effective manufacture scheduling can improve on-time delivery, reduce inventory, cut lead times, and improve the utilization of bottleneck resources [1].

One of the most studied combinatorial optimization problems is the job shop scheduling problem. Nevertheless, it still remains a challenging problem to solve optimally. From a complexity point of view, the problem is NP-hard i.e. it can be solved in nondeterministic polynomial time [2], [3].

The simplest scheduling problem is the single machine sequencing problem [4]. Minimizing the total makespan is one of the basic objectives studied in the scheduling literature. The shortest processing time dispatching rule will give an optimal schedule in the single machine case if the tool life is considered infinitely long [5]. The scheduling with sequence-dependent setups is recognized as being difficult and most existing results in the literature focus on either a single machine or several identical machines [6]-[9]. The real-life scheduling problems usually have to consider multiple no identical machines. Most of the processing machines needed to process the jobs are available in the manufacturer's own factory and are of fixed (finite) number. Sometimes, certain details must be ordered to a third party companies to complete very specific processing as molding for example. In cases like that, the processing schedules are to be agreed for delivery times from the third-party processing. That means generating a schedule to process all jobs, so as to minimize the total cost, including the satisfaction of the due dates of the jobs [10]. Different manufacturing environments induce different scheduling constraints, some of which may be very specific to the problem under consideration [11]. The classical job shop scheduling problem is one of the most typical and complicated problems formulated as follows: 1) a job shop consists of a set of different machines that perform operations of jobs; 2) each job is composed of a set of operations and the operation order on machines is prescribed; 3) each operation is characterized by the required machine and the processing time. In the last two decades, numerous techniques was developed on deterministic classical job shop scheduling, such as analytical techniques, rule-based approach and meta-heuristic algorithms and algorithms using dynamic programming [12]-[16].

Approximately up to 2004 the computers have had gradually increasing of CPU performance by increasing of operating frequency, and the need of multi core systems was not so obvious. NVIDIA has invented the graphics processing unit (GPU) that became a pervasive parallel processor to date. It has evolved into a processor with unprecedented floating-point performance and programmability and today's GPUs greatly outpace CPUs in performance, making them the ideal processor to accelerate a variety of data applications. Today, some GPUs have thousands of processing cores and with CUDA programming model [17] they offer a new way to use the GPU as a general purpose powerful processor. This opens up new horizons in development and application of new approaches based on parallel algorithms [18].

The proposed scheduling approach concerns a problem of scheduling for multiple details with fixed processing time and predetermined order of processing operations over different machines. Three general types of open job shop scheduling problems are investigated: 1) unconstrained – the processing of details is independent of each other; 2) constrained - the order of processing of details is given and cannot be changed and 3) "semi-constrained" – a group of details have predetermined order of processing while other details can be processed in any order. The objective in question for these open job shop scheduling problems is to determine processing schedule that minimizes the makespan in such way that all details processing conforms to the given requirements and restrictions. The constrained job shop scheduling problems are

D. I. Borissova is with the Institute of Information and Communication Technologies at Bulgarian Academy of Sciences, Sofia – 1113, Bulgaria, Department of Information Processes and Decision Support Systems (phone: 3952 9792055; e-mail: dborissova@iit.bas.bg).

I. C. Mustakerov is with the Institute of Information and Communication Technology at the Bulgarian Academy of Sciences, Sofia – 1113, Bulgaria, Department of Information Processes and Decision Support Systems (phone: 3952 9793241; e-mail: mustakerov@iit.bas.bg)).

more easily addressed by combinatorial optimization then unconstrained problems [19]. One possible way to deal with unconstrained and semi-constrained open job shop scheduling problems is by using of enumerative combinatorics. In the paper the mixed semi-constrained open job shop scheduling problem is investigated as generalization of the other two cases. An algorithm with two branches (aggregate optimization and parallel optimization) is described. The first branch is based on solution of single optimization task that takes into account all possible combinations of details processing order. The second branch is based on solving in parallel of a number of optimization tasks where each task corresponds to one of the possible variants of the details processing orders [20]. The proposed enumerative combinatorics approach for optimal open job shop scheduling is numerically tested for a real life example.

II. Problem Description

There is a group of details that need to be processed on multiple machines. Some of these details are connected with each other through given order of processing while other can be processed in any order. All details have predetermined sequence of operations on particular machines. The processing times of details on machines are deterministic and are known in advance. The problem is to determine the processing schedule with minimal makespan for all details processing according to requirements.

For clarity of presentation the investigated open job shop problem will be explained by a real life example for a set of six details (jobs) with given sequences of operations that should be processed on four different machines with known processing time on each machine. All available data are summarized in Table I where operations' designation O_{ij} means processing of detail i on machine j and processing times are given in hours.

TABLE I
Input Data for Details Processing

Details (Jobs)	Operations	Processing time on M1	Processing time on M2	Processing time on M3	Processing time on M4
D_1	O_{11}	8			
	O_{12}		6		
	O_{14}				6
D_2	O_{21}	8			
	O_{22}		9		
	O_{24}				6
D_3	O_{31}	8			
	O_{33}			8	
	O_{32}		8		
D_4	O_{41}	4			
	O_{42}		2		
	O_{43}			2	
D_5	O_{51}	4			
	O_{52}		9		
	O_{53}			5	
D_6	O_{61}	6			
	O_{63}			4	

The sequence of operations for each detail are given by arrays of indexes of machines where these operations are to be performed: O_1 = [1, 2, 4], O_2 = [1, 2, 4], O_3 = [1, 3, 2], O_4 = [1, 2, 3], O_5 = [1, 2, 3] and O_6 = [1, 3]. Due their processing specifics the details D_4, D_5 and D_6 should be processed in a sequential order – i.e. D_5 detail to be processed after completion of the processing of the detail D_4 and D_6 is processed after the processing of the detail D_5.

All jobs cannot overlap on the machines and one job cannot be processed simultaneously by two or more machines. Each operation needs to be processed during an uninterrupted period of a given length on a given machine.

Taking all of these considerations into account a details processing schedule that minimizes the makespan has to be determined. The described problem can be represented as machine-oriented Gantt chart that visualizes the sequence of details processing on different machines as shown in Fig. 1.

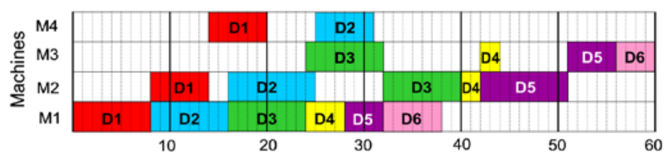

Fig. 1. Gantt chart for a processing schedule of 6 details on 4 machines (*D4*, *D5* and *D6* are processed in sequential order)

III. Mathematical Models Formulation

Most variants of job shop scheduling problem are NP-hard in the strong sense and thus defy ordinary solution methods. That is why new techniques are required to overcome difficulties and to be applied to particular manufacturing job shop scheduling problems. The generalized goal of most of optimal scheduling problems is to minimize the overall costs. Although many costs could be considered for optimization, the minimizing of details processing time duration is one of most frequently used. It provides the effective machines utilization and serves the optimization of details delivering and storage. The used enumerative combinatorics for modeling of open job shop scheduling is characterized by strong dependence of the specifics of the particular problem to be solved. The different distributions of operations among machines will require different mathematical models. That is why the proposed in the paper approach will be illustrated on the example data shown in Table I.

A. *Mathematical Model for Aggregate Optimization*

Aggregate optimization approach takes into account simultaneously all feasible permutations of the order of processing of the details. The formulation of the corresponding optimization model is based on data from Table I. There are 3 details (*D1*, *D2* and *D3*) that can be processed in any order. The group of dependant details *D4*, *D5* and *D6* can be considered as one independent detail and the number of all possible processing sequences of details can be calculated as number of permutations of 4, i.e. 4! = 24. To build a model that will choose the best possible processing sequence of details, 24 binary integer variables y_i are introduced and used as decision variable for a group of constraints that correspond to particular variant of details processing order.

$$\min\left(x_{1,end} + x_{2,end} + x_{3,end} + x_{4,end} + x_{5,end} + x_{6,end}\right) \quad (1)$$

subject to

- restrictions for each detail processing order over machines:
  - $x_{1,2} - x_{1,1} \geq 8$ (2)
  - $x_{1,4} - x_{1,2} \geq 6$ (3)
  - $x_{1,end} - x_{1,4} \geq 6$ (4)
  - $x_{2,2} - x_{2,1} \geq 8$ (5)
  - $x_{2,4} - x_{2,2} \geq 9$ (6)
  - $x_{2,end} - x_{2,4} \geq 6$ (7)
  - $x_{3,3} - x_{3,1} \geq 8$ (8)
  - $x_{3,2} - x_{3,3} \geq 8$ (9)
  - $x_{3,end} - x_{3,2} \geq 8$ (10)
  - $x_{4,2} - x_{4,1} \geq 4$ (11)
  - $x_{4,3} - x_{4,2} \geq 2$ (12)
  - $x_{4,end} - x_{4,3} \geq 2$ (13)
  - $x_{5,2} - x_{5,1} \geq 4$ (14)
  - $x_{5,3} - x_{5,2} \geq 9$ (15)
  - $x_{5,5} - x_{5,3} \geq 4$ (16)
  - $x_{5,end} - x_{5,5} \geq 5$ (17)
  - $x_{6,3} - x_{6,1} \geq 6$ (18)
  - $x_{6,end} - x_{6,3} \geq 4$ (19)
- restrictions for details priority processing order as follows: $D_1 \to D_2 \to D_3 \to D_4 \to D_5 \to D_6$:

$y_1(x_{2,1} - x_{1,1}) \geq 8y_1$ (20)

$y_1(x_{3,1} - x_{2,1}) \geq 8y_1$ (21)

$y_1(x_{4,1} - x_{3,1}) \geq 8y_1$ (22)

$y_1(x_{5,1} - x_{4,1}) \geq 4y_1$ (23)

$y_1(x_{6,1} - x_{5,1}) \geq 4y_1$ (24)

$y_1(x_{2,2} - x_{1,2}) \geq 6y_1$ (25)

$y_1(x_{3,2} - x_{2,2}) \geq 9y_1$ (26)

$y_1(x_{4,2} - x_{3,1}) \geq 8y_1$ (27)

$y_1(x_{5,2} - x_{4,2}) \geq 2y_1$ (28)

$y_1(x_{4,3} - x_{3,3}) \geq 8y_1$ (29)

$y_1(x_{5,3} - x_{4,3}) \geq 2y_1$ (30)

$y_1(x_{6,3} - x_{5,3}) \geq 5y_1$ (31)

$y_1(x_{2,4} - x_{1,4}) \geq 6y_1$ (32)

- for processing order $D_1 \to D_3 \to D_2 \to D_4 \to D_5 \to D_6$:

$y_2(x_{3,1} - x_{1,1}) \geq 8y_2$ (33)

$y_2(x_{2,1} - x_{3,1}) \geq 8y_2$ (34)

$y_2(x_{4,1} - x_{2,1}) \geq 8y_2$ (35)

$y_2(x_{5,1} - x_{4,1}) \geq 4y_2$ (36)

$y_2(x_{6,1} - x_{5,1}) \geq 4y_2$ (37)

$y_2(x_{2,2} - x_{1,2}) \geq 6y_2$ (38)

$y_2(x_{3,2} - x_{2,2}) \geq 9y_2$ (39)

$y_2(x_{4,2} - x_{3,1}) \geq 8y_2$ (40)

$y_2(x_{5,2} - x_{4,2}) \geq 2y_2$ (41)

$y_2(x_{4,3} - x_{3,3}) \geq 8y_2$ (42)

$y_2(x_{5,3} - x_{4,3}) \geq 2y_2$ (43)

$y_2(x_{6,3} - x_{5,3}) \geq 5y_2$ (44)

$y_2(x_{2,4} - x_{1,4}) \geq 6y_2$ (45)

- for processing order $D_2 \to D_1 \to D_3 \to D_4 \to D_5 \to D_6$:

$y_3(x_{1,1} - x_{2,1}) \geq 8y_3$ (46)

$y_3(x_{3,1} - x_{1,1}) \geq 8y_3$ (47)

$y_3(x_{4,1} - x_{3,1}) \geq 8y_3$ (48)

$y_3(x_{5,1} - x_{4,1}) \geq 4y_3$ (49)

$y_3(x_{6,1} - x_{5,1}) \geq 4y_3$ (50)

$y_3(x_{2,2} - x_{1,2}) \geq 6y_3$ (51)

$y_3(x_{3,2} - x_{2,2}) \geq 9y_3$ (52)

$y_3(x_{4,2} - x_{3,1}) \geq 8y_3$ (53)

$y_3(x_{5,2} - x_{4,2}) \geq 2y_3$ (54)

$y_3(x_{4,3} - x_{3,3}) \geq 8y_3$ (55)

$y_3(x_{5,3} - x_{4,3}) \geq 2y_3$ (56)

$y_3(x_{6,3} - x_{5,3}) \geq 5y_3$ (57)

$y_3(x_{2,4} - x_{1,4}) \geq 6y_3$ (58)

- for processing order $(D_4 \to D_5 \to D_6) \to D_1 \to D_2 \to D_3$:

$y_4(x_{5,1} - x_{4,1}) \geq 4y_4$ (59)

$y_4(x_{6,1} - x_{5,1}) \geq 4y_4$ (60)

$y_4(x_{1,1} - x_{6,1}) \geq 6y_4$ (61)

$y_4(x_{2,1} - x_{1,1}) \geq 8y_4$ (62)

$y_4(x_{3,1} - x_{2,1}) \geq 8y_4$ (63)

$y_4(x_{5,2} - x_{4,2}) \geq 2y_4$ (64)

$y_4(x_{1,2} - x_{5,2}) \geq 9y_4$ (65)

$y_4(x_{22} - x_{12}) \geq 6y_4$ (66)

$y_4(x_{3,2} - x_{2,2}) \geq 9y_4$ (67)

$y_4(x_{5,3} - x_{4,3}) \geq 2y_4$ (68)

$y_4(x_{6,3} - x_{5,3}) \geq 5y_4$ (69)

$y_4(x_{3,3} - x_{6,3}) \geq 4y_4$ (70)

$y_4(x_{2,4} - x_{1,4}) \geq 6y_4$ (71)

and so on ………………………

$$\sum_{k=1}^{24} y_k = 1,\ y_k \in \{0,1\},\ k = 1, 2, \dots 24 \quad (72)$$

where $x_{i,j} \geq 0$ are integers representing the moments of time when processing of detail i on machine j starts.

The overall details processing time duration (makespan) can be defined as difference between end processing moment of the last detail and start processing moment of the first detail and if the processing starts at moment zero moment then the objective can be minimization of the end processing moment of the last detail. It is unknown in advance which of details will be processed last. That is why the objective function (1) is composed as additive function of end moments for processing of all details. In this way minimizing of the sum of end processing moments of all details will correspond to minimal makespan.

The group of constraints (2) – (19) represents technological order for processing of each of details. For example, the constraint (2) describes the fact that the processing of the detail 1 on the machine 2 should begin at the earliest after the end of processing it on a machine 1 i.e. 8 hours or more later. The group of constraints (20) – (32) describes the order of processing of the details relative to one another taking into account that one machine can process one detail at a time. For example, constraint (20) defines that detail 1 should be processed before detail 2 or in other words, detail 2 should wait at least 8 hours after start of processing of detail 1 on machine 1.

In this aggregated mathematical model for open job shop scheduling all 24 variants of details processing order are taken into consideration by introduced decision variables y_k. To determine the minimal makespan schedule via enumerative combinatorics a single solution of the formulated above optimization task is needed.

B. Mathematical Model for Parallel Optimization

Parallel approach is based on assumption that determination of one feasible schedule does not depend on determination of other feasible schedules. For example, if details processing sequence is D1→ D2 → D3 → D4 → D5 → D6, the optimization task is formulated as follows:

$$\min\left(x_{1,end} + x_{2,end} + x_{3,end} + x_{4,end} + x_{5,end} + x_{6,end}\right) \quad (73)$$

- subject to restrictions for detail processing order over machines:

$$x_{1,2} - x_{1,1} \geq 8 \quad (74)$$
$$x_{1,4} - x_{1,2} \geq 6 \quad (75)$$
$$x_{1,end} - x_{1,4} \geq 6 \quad (76)$$
$$x_{2,2} - x_{2,1} \geq 8 \quad (77)$$
$$x_{2,4} - x_{2,2} \geq 9 \quad (78)$$
$$x_{2,end} - x_{2,4} \geq 6 \quad (79)$$
$$x_{3,3} - x_{3,1} \geq 8 \quad (80)$$
$$x_{3,2} - x_{3,3} \geq 8 \quad (81)$$
$$x_{3,end} - x_{3,2} \geq 8 \quad (82)$$
$$x_{4,2} - x_{4,1} \geq 4 \quad (83)$$
$$x_{4,3} - x_{4,2} \geq 2 \quad (84)$$
$$x_{4,end} - x_{4,3} \geq 2 \quad (85)$$
$$x_{5,2} - x_{5,1} \geq 4 \quad (86)$$
$$x_{5,3} - x_{5,2} \geq 9 \quad (87)$$
$$x_{5,5} - x_{5,3} \geq 4 \quad (88)$$
$$x_{5,end} - x_{5,5} \geq 5 \quad (89)$$
$$x_{6,3} - x_{6,1} \geq 6 \quad (90)$$
$$x_{6,end} - x_{6,3} \geq 4 \quad (91)$$

- subject to restrictions for the details priority processing:

$$x_{2,1} - x_{1,1} \geq 8 \quad (92)$$
$$x_{3,1} - x_{2,1} \geq 8 \quad (93)$$
$$x_{4,1} - x_{3,1} \geq 8 \quad (94)$$
$$x_{5,1} - x_{4,1} \geq 4 \quad (95)$$
$$x_{6,1} - x_{5,1} \geq 4 \quad (96)$$
$$x_{2,2} - x_{1,2} \geq 6 \quad (97)$$
$$x_{3,2} - x_{2,2} \geq 9 \quad (98)$$
$$x_{4,2} - x_{3,1} \geq 8 \quad (99)$$
$$x_{5,2} - x_{4,2} \geq 2 \quad (100)$$
$$x_{4,3} - x_{3,3} \geq 8 \quad (101)$$
$$x_{5,3} - x_{4,3} \geq 2 \quad (102)$$
$$x_{6,3} - x_{5,3} \geq 5 \quad (103)$$
$$x_{2,4} - x_{1,4} \geq 6 \quad (104)$$
$$x_{i,j} \geq 0 \quad (105)$$

The formulated task (73) – (105) takes into account the details processing sequence $D_1 \rightarrow D_2 \rightarrow D_3 \rightarrow D_4 \rightarrow D_5 \rightarrow D_6$. To define the minimum makespan for other processing sequence this task should be reformulated. The group of restrictions for details priority processing (24) to (36) has to be changed to correspond to other possible details processing sequence. As it was mentioned, there are 3 details (D1, D2 and D3) that can be processed in any order. The group of dependant details D4, D5 and D6 can be considered as one independent detail and the number of all possible processing sequences can be calculated as number of permutations of 4, i.e. number of different processing sequences that have to be evaluated is equal to 4! = 24.

For example, if details processing sequence is $D_1 \rightarrow D_3 \rightarrow D_2 \rightarrow D_4 \rightarrow D_5 \rightarrow D_6$ the restrictions (92) – (94) should be reformulated as:

$$x_{3,1} - x_{1,1} \geq 8 \quad (106)$$
$$x_{2,1} - x_{3,1} \geq 8 \quad (107)$$
$$x_{4,1} - x_{2,1} \geq 8 \quad (108)$$

The objective function (5) and the rest of restrictions remain the same.

If details processing sequence is $D_2 \rightarrow D_1 \rightarrow D_3 \rightarrow D_4 \rightarrow D_5 \rightarrow D_6$ the restrictions (93) – (94) have to be changed as:

$$x_{1,1} - x_{2,1} \geq 8 \quad (109)$$
$$x_{3,1} - x_{1,1} \geq 8 \quad (110)$$

$$x_{4,1} - x_{3,1} \geq 8 \quad (111)$$

The objective function and the rest of restrictions remain the same.

If the group of details D_4, D_5 and D_6 is to be processed in the first place i.e. details processing order is $D_4 \to D_5 \to D_6 \to D_1 \to D_2 \to D_3$ the restrictions (93) – (104) are transformed to the same objective function, modified restrictions:

$$x_{5,1} - x_{4,1} \geq 4 \quad (112)$$

$$x_{6,1} - x_{5,1} \geq 4 \quad (113)$$

$$x_{1,1} - x_{6,1} \geq 6 \quad (114)$$

$$x_{2,1} - x_{1,1} \geq 8 \quad (115)$$

$$x_{3,1} - x_{2,1} \geq 8 \quad (116)$$

$$x_{5,2} - x_{4,2} \geq 2 \quad (117)$$

$$x_{1,2} - x_{5,2} \geq 9 \quad (118)$$

$$x_{22} - x_{12} \geq 6 \quad (119)$$

$$x_{3,2} - x_{2,2} \geq 9 \quad (120)$$

$$x_{5,3} - x_{4,3} \geq 2 \quad (121)$$

$$x_{6,3} - x_{5,3} \geq 5 \quad (122)$$

$$x_{3,3} - x_{6,3} \geq 4 \quad (123)$$

$$x_{2,4} - x_{1,4} \geq 6 \quad (124)$$

The restrictions for detail processing order over machines (74) – (91) remain the same.

In similar way, all possible combinations of detail processing sequences reflecting in 24 different modifications of basic optimization task (73) – (105) are done.

In this approach the formulated aggregated optimization model is decomposed on number of similar optimization tasks corresponding to the permutations for independent details processing. Each task solution is independent of other tasks solutions and defines some value of makespan. When all of these values are defined it is easy to compare them and to choose the details processing sequence with minimal makespan. The number of tasks to be solved is equal to number of details permutations. Each of these optimization tasks has not simple structure and can be solved by means of modern multicore processors as NVIDIA GPU.

IV. Generalized Algorithm for Optimal Open Job Shop Scheduling

To find the optimal processing schedule that is minimal in the sense of shortest makespan, a generalized algorithm for optimal job shop scheduling of semi-constrained details processing on multiple machines shown in Fig. 2 is proposed.

The algorithm consists of two general branches – for aggregate optimization and for parallel optimization. The two first steps of the algorithm are common for both of branches. At the first step the number of jobs that can be processed in any sequence to each other is determined.

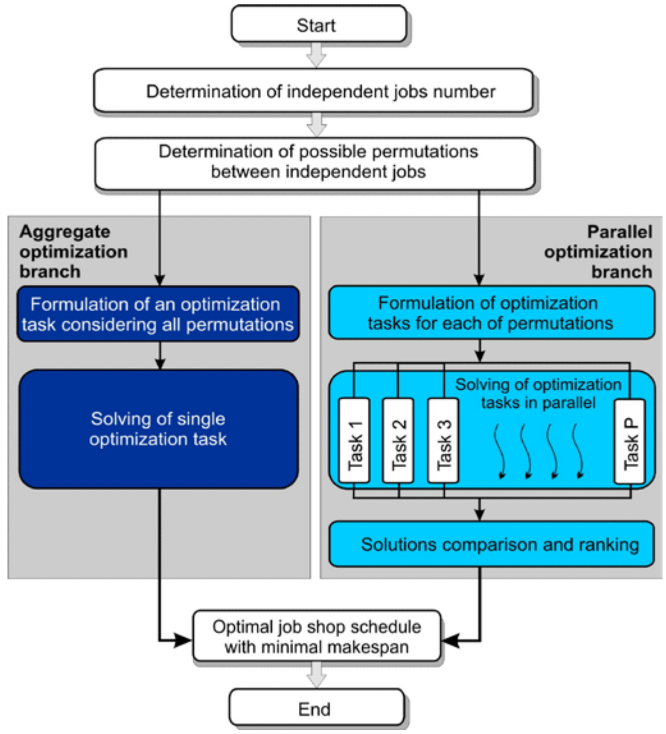

Fig. 2. Generalized algorithm for open job shop scheduling

If exist dependant jobs (as for details D4, D5 and D6) they are considered as one independent job. On the second step all possible orderings (permutations) for processing of independent details are defined.

Then depending on the available software and hardware capabilities, aggregate or parallel branch can be chosen. If there exist linear programming solver characterized by a large number of allowable restrictions and variables the aggregate branch of algorithm can be used. If a CUDA GPU is available its multicore potential can be used by the parallel branch of algorithm. It is based on formulation and solution in parallel of a variety of optimization tasks for each possible processing sequence. Then makespan for all of the details processing variants are compared and the schedule corresponding to minimal makespan is chosen as the best one.

V. Numerical Experimentation and Discussions

The two branches of algorithm are numerically tested by using the input data from Table I. The all of possible processing sequences are visualized in Fig. 3.

The makespan values for all of the possible variants of details processing sequences vary within interval of 65 to 54 hours. The using of aggregate optimization branch of algorithm defines as optimal the processing sequence D1 → D4 → D5 → D6 → D2 → D3 with makespan equal to 54 hours. By using of parallel optimization branch, 4 processing sequences shown in Table III have equal makespan with duration of 54 hours.

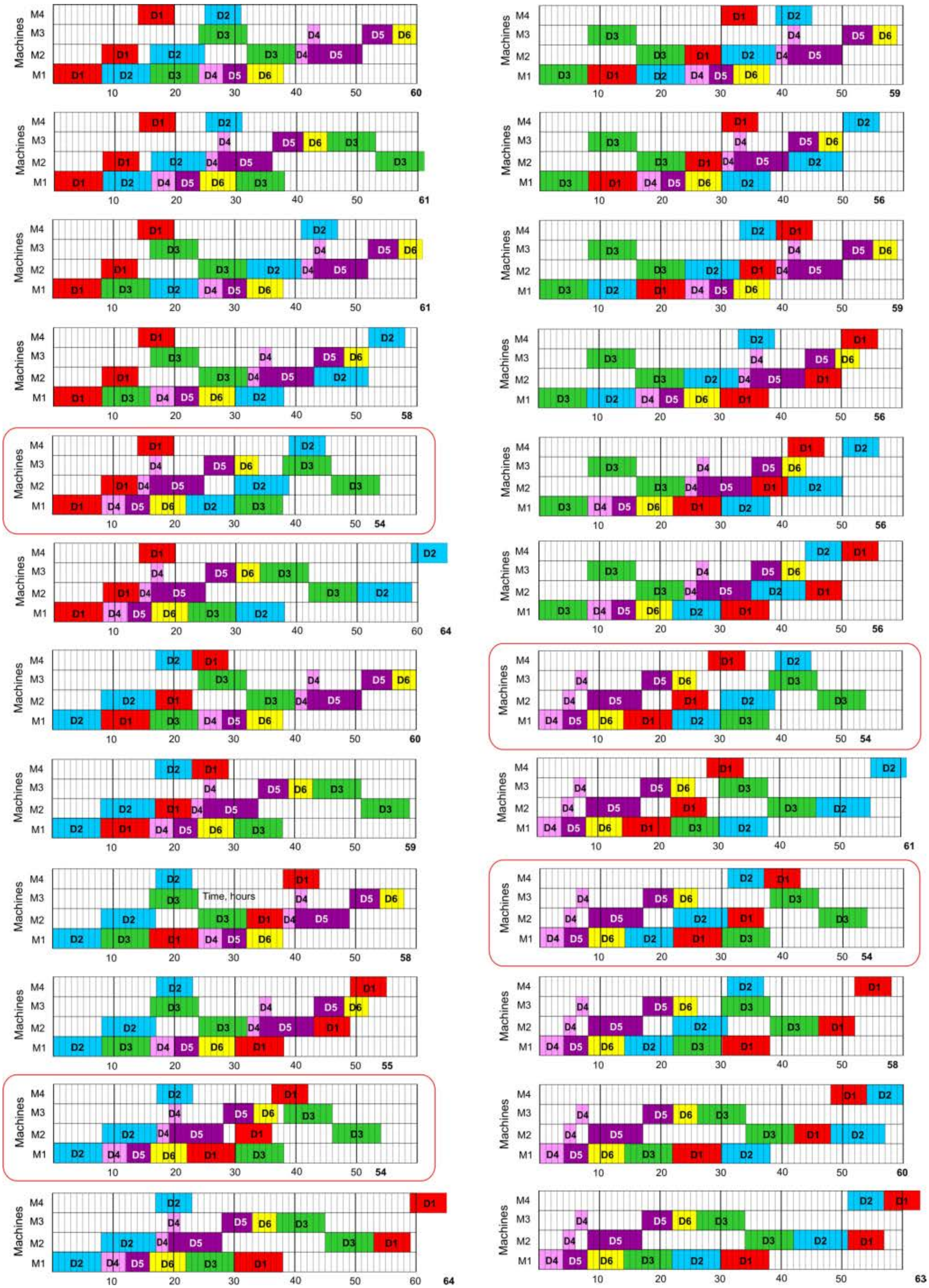

Fig. 3. Variants of open job shop schedules of the investigated problem

TABLE III
SOLUTIONS RESULTS

Optimal processing sequences	Makespan, hours
$D1 \to D4 \to D5 \to D6 \to D2 \to D3$	54
$D2 \to D4 \to D5 \to D6 \to D1 \to D3$	54
$D4 \to D5 \to D6 \to D1 \to D2 \to D3$	54
$D4 \to D5 \to D6 \to D2 \to D1 \to D3$	54

This means that parallel optimization branch supplies more informative results. To get the same information by using aggregate optimization branch the decision variable for the optimal solution has to be equalized to zero and the optimization task should be solved again. Then the solution will define next processing sequence with makespan equal to 54 hours. Again the corresponding decision variable should be zeroed to get the next processing sequence with makespan of 54 hours. This procedure is repeated until it is found a processing sequence with makespan greater than the minimum.

All formulated optimization tasks are solved by means of LINGO solver [21]. Despite the fact that open job shop integer problems are difficult to solve (in general they are NP-hard), the numerical testing show quite acceptable solution times of few seconds for the example in consideration.

The real-life job shop scheduling problems have their own specifics. With this approach the number of restrictions increases with increasing of number of variants of ordering of details processing. As it was mentioned earlier, the specific of problem is essential for scale of this increasing. It can be seen that the relationship between the processing details sequence and machines occupation have a significant impact on overall manufacturing process performance. For the described example, it turned out that Machine 1 is the busiest machine among the others. One possible approach to shorten the overall makespan is to consider more than one machine of type 1 and to estimate the influence of machine's number on the total makespan.

The proposed approach based on generalized algorithm for open job shop scheduling can be used for other similar problems concerning optimal job shop scheduling.

VI. CONCLUSION

The main contribution of the paper is using of the enumerative combinatorics approach for deterministic open job shop scheduling of unconstrained or semi-constrained details processing on multiple machines. The main goal is to determine the processing schedule with minimal makespan. An algorithm with aggregate optimization branch and parallel optimization branch is proposed. The both of branches are based on considering of all permutations of independent details processing sequences. The execution of the aggregate optimization branch provides a single optimal processing schedule. The using of parallel optimization branch defines schedules for all possible details processing sequences. The corresponding makespans are compared and the best schedule in sense of minimal makespan is determined. This branch of algorithm provides a full picture of all possible processing schedules to choose from. Depending on the specific of job shop problem it is possible to have a number of processing sequences with equal makespan. Parallel branch of the algorithm has the advantage of providing this information in a direct manner.

The disadvantage of the proposed enumerative combinatorics approach is the need of considering of all permutations for possible details processing sequences. This restricts the practical application to open job shop scheduling problems with relatively small number of independent details. When large scale open job shop problems can be decomposed on problems with small number of independent details the proposed approach can be effectively used. There exist effective algorithms for permutations determination and if the process of formulation of optimization tasks is automated the area of practical application could be extended. This is a direction for future investigations. Other direction for investigations of described approach is to explore how the increasing of the number of identical machines will contribute to decreasing of makespan.

REFERENCES

[1] D. Chen, P. B. Luh, L. S. Thakur and J. Moreno Jr., "Optimization-based manufacturing scheduling with multiple resources, setup requirements, and transfer lots", *IIE Transactions*, vol. 35, 2003, pp. 973-985.

[2] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and job shop scheduling", *Mathematics of Operations Research*, vol. 1(2), 1976, pp. 117-129.

[3] Yu. N. Sotskov and N. V. Shakhlevich. "NP-hardness of shop-scheduling problems with three jobs", *Discrete Applied Mathematics*, vol. 59, 1995, pp. 237-266.

[4] S. J. Mason, P. Qu, E. Kutanoglu and J. W. Fowler, "The single machine multiple orders per job scheduling problem", available: http://ie.fulton.asu.edu/files/shared/workingpapers/MOJ_Paper.pdf

[5] M. S. Akturk, J. B. Ghosh and E. D. Gunes, "Scheduling With Tool Changes to Minimize Total Completion Time: A Study of Heuristics and Their Performance", *Naval Research Logistics*, vol. 50(1), 2003, pp.15-30.

[6] S. C. Kim and P. M. Bobrowski. "Impact of sequence-dependent setup time on job shop scheduling performance". *Int. Journal of Production Research*, vol. 32(7), 1994, pp. 1503-1520.

[7] I. M. Ovacik, and R. Uzsoy, "Rolling horizon algorithm for a singlemachine dynamic scheduling problem with sequence-dependent setup times", *Int. Journal of Production Research*, vol. 32(6), 1994, pp. 1243-1263.

[8] H. L. Young, K. Bhaskaran, and M. Pinedo, "A heuristic to minimize the total weighted tardiness with sequence-dependent setups", *IIE Transactions*, vol. 29, no. 1, 1997, pp. 45–52.

[9] T. Yamada and R. Nakano. *Genetic algorithms in engineering systems*, The Institution of Electrical Engineers, 1997, ISBN: 0 85296 902 3, A.M.S. Zalzala and P.J. Fleming Eds., Chapter 7: Job-shop scheduling, pp. 134-160.

[10] J. Wang, P. B. Luh, X. Zhao and J. Wang, "An Optimization-Based Algorithm for Job Shop Scheduling", *Sadhana*, vol. 22, 1997, pp. 241-256.

[11] P. Baptiste and C. L. Pape, "Disjunctive constraints for manufacturing scheduling: principles and extensions", *Computer Integrated Manufacturing*, vol. 9(4), 1996, pp. 306-310.

[12] M. Pinedo, "Stochastic scheduling with release dates and due dates", *Operations Research*, vol. 31, no. 3, 1983, pp. 559-572.

[13] R. R. Weber, P. Varaiya and J. Walrand, "Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime", *Applied Probability*, vol. 23(3), 1986, pp. 841-847.

[14] D. Golenko-Ginzburg and A. Gonik, "Optimal job-shop scheduling with random operations and cost objectives", *Production Economics*, vol. 76(2), 2002, pp. 147-157.

[15] S. R. Lawrence and E.C. Sewell, "Heuristic optimal static and dynamic schedules when processing times are uncertain", *Operations Management*, vol. 15(1), 1997, pp. 71-82.

[16] J. A. S. Gromicho, J. J. van Hoorn, F. Saldanha-da-Gama and G. T. Timmer. "Solving the job-shop scheduling problem optimally by dynamic programming", *Computers & Operations Research*, vol. 39, 2012, pp. 2968-2977.

[17] NVIDIA. Whitepaper NVIDIA's Next Generation CUDA Compute Architecture: Fermi, 2009.

[18] S. A. Mirsoleimani, A. Karami and F. Khunjush. "A Parallel Memetic Algorithm on GPU to Solve the Task Scheduling Problem in Heterogeneous Environments", *Genetic and Evolutionary Computation Conference*, 2013, pp. 1181-1188.

[19] I. Mustakerov, D. Borissova. "Optimal Manufacturing Scheduling for Dependent Details Processing". *Computer and Information Engineering*. pISSN: 2010-3921, eISSN: 2010-393X, vol. 2(8), 2008, pp. 493-497.

[20] Borissova D., I. Mustakerov. "A parallel algorithm for optimal job shop scheduling of semi-constrained details processing on multiple machines". *Conference on Circuits, Systems, Communications and Computers*, July 17-21, 2014 Santorini Island, Greece, pp. 145-150

[21] Lindo Systems ver. 12, http://www.lindo.com

Dr. Daniela I. Borissova is Associated Professor in the Instutute of ICT at Bulgarian Academy of Sciences, department of Information Processes and Decision Support Systems. Office address: Sofia 1113, Acad. G. Bonchev St., Block 2, room 515.
E-mail: dborissova@iit.bas.bg.
Major fields of scientific research: decision support systems, modeling and optimization, operations research, night vision devices, software systems for information processing, Web-based applications.

Dr. Ivan C. Mustakerov is Associated Professor in the Instutute of ICT at Bulgarian Academy of Sciences, department of Information Processes and Decision Support Systems. Office address: Sofia 1113, Acad. G. Bonchev St., Block 2; room 509.
E-mail: mustakerov@iit.bas.bg.
Major fields of scientific research: operations research, systems modelling and optimization, decision support systems, engineering systems design, software systems for information processing, e-learning and Web-based applications.