A parallel algorithm for optimal job shop scheduling of semi-constrained details processing on multiple machines

Daniela I. Borissova and Ivan C. Mustakerov

***Abstract*—**The paper presents an approach for a variant of constrained job shop scheduling where processing of some details is independent and other have fixed processing order (semi-constrained scheduling). The described approach aims to determine a schedule that minimizes the total makespan in such way that all given operations sequences are satisfied. For the goal, a parallel algorithm is proposed based on linear programming optimization tasks that are solved in parallel. The described approach for optimal job shop scheduling of semi-constrained details is numerically tested for real job shop scheduling problem.

***Keywords*—**Job shop scheduling, linear programming, minimal makespan, semi-constrained details processing.

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 very 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]-[8]. The real-life scheduling problems usually have to consider multiple no identical machines.

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)).

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 [9]. Different manufacturing environments induce different scheduling constraints, some of which may be very specific to the problem under consideration [10].

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 [11]-[15].

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 parallel applications. GPUs have hundreds of processing cores and with CUDA programming model [16] software and hardware architecture is available using of a variety of high level programming languages. This represented a new way to use the GPU as a general purpose parallel computer processor. This opens up new horizons in development and application of new approaches based on parallel algorithms [17].

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. An essential feature of the investigated job shop

ISBN: 978-1-61804-236-1

scheduling problem is that: 1) the processing of some details depends on processing of other details i.e. a group of details have predetermined order of processing and 2) the processing of the other parts is independent of each other. This variant of job shop scheduling can be named as semi-constrained job shop scheduling problem. It is approached in the paper by means of an algorithm based on parallel solving of a number of integer linear programming tasks. The main goal is to determine a schedule that minimizes the total makespan in such way that all details processing conforms to the given restrictions. The proposed parallel algorithm for optimal job shop scheduling of semi-constrained details processing on multiple machines 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 different machines. The details processing times on machines are deterministic and are known in advance. The problem is to determine the minimum makespan for all details processing according to requirements.

For clarity of presentation the investigated 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 as D_1 {O_{11}, O_{12}, O_{14}}, D_2 {O_{21}, O_{22}, O_{24}}, D_3 {O_{31}, O_{33}, O_{32}}, D_4 {O_{41}, O_{42}, O_{43}}, D_5 {O_{11}, O_{12}, O_{13}} and D_6 {O_{61}, O_{63}}. Due their post-processing specifics the details D_4, D_5 and D_6 should be processed in a sequential order. 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. The goal of the investigated scheduling problem is to determine a schedule that minimizes the total makespan. The described problem can be represented as machine-oriented Gantt chart visualizing the sequence of details processing as shown in Fig. 1

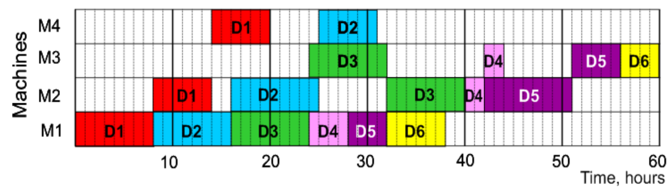

Fig. 1. Gantt chart for a schedule

III. Mathematical Model 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 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. Using those considerations, an optimization model can be formulated following the notations:

1) number of details indexed by $i \in \{1,2,\ldots,N\}$
2) number of machines for detail processing, indexed by $j \in \{1,2,\ldots,M\}$
3) job processing times $T_{i,j}$ of each detail *i* on machine *j* are known constants.
4) $x_{i,j}$, is the moment of time for starting of processing of detail *i* on machine *j*.

The scheduling problem is formalized via linear programming formulation that minimizes the makespan as:

$$\min \rightarrow \sum_{i=1}^{N} x_{i,end} \tag{1}$$

subject to

$$x_{i,j+1} - x_{i,j} \geq T_{i,j}, \ \forall i=1,2,\ldots,N, \tag{2}$$

$$x_{i+1,j} - x_{i,j} \geq T_{i,j}, \ \forall j=1,2,\ldots,M \tag{3}$$

$$x_{i,j} \geq 0 \tag{4}$$

The objective function (1) minimizes the processing end time of all details. The relation (2) expresses the restriction for operation sequence for each detail, while relation (3) illustrates the restriction for the details processing order (if any). For dependant details processing there exist a certain order of processing, but in case of independent details the processing order is not fixed. This way formulated model can be used to determine the optimal makespan for a given sequence of details processing. To find the optimal makespan among all of the possible sequences of independent details processing a parallel algorithm can be applied.

IV. Parallel Algorithm for Minimal Makespan Determination

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

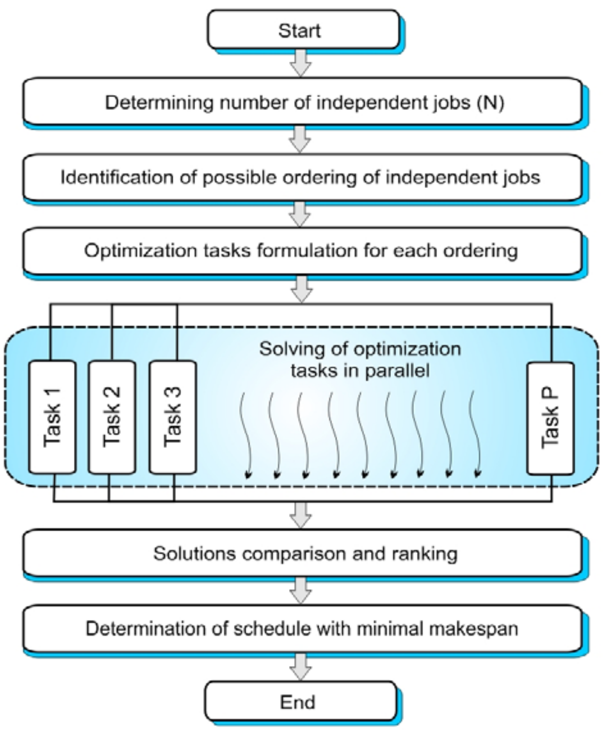

Fig. 1. Parallel algorithm for job shop scheduling

On the first step of the algorithm the independent jobs are to be defined and designated using name of the corresponding detail. If exist dependant jobs (as for details D4, D5 and D6) they are considered as one independent job named after first detail of processing sequence (D4). Then the overall number of independent details is determined. On the second step all possible orderings (permutations) for processing of independent details are defined. On the next step each of the details processing ordering is formalized by proper optimization task following the model (1) – (4). It is important to stress here that all formulated in this way optimization tasks are independent of each other. The solution of any of them does not depend on data or solution of other tasks. This makes them perfect candidates for using of parallel threads for their solving on step 4. Then, tasks solution results (makespan values) are compared and ranked. On the last step the schedule corresponding to the task with best solution with minimal makespan value is chosen as optimal job shop schedule.

V. Numerical Example

In deterministic job shop scheduling problem, is assumed that all processing times are fixed and known in advance, so using the input data from Table I, the optimization model (1) – (4) can be expressed as:

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

$$x_{1,2} - x_{1,1} \geq 8 \quad (6)$$

$$x_{1,4} - x_{1,2} \geq 6 \quad (7)$$

$$x_{1,end} - x_{1,4} \geq 6 \quad (8)$$

$$x_{2,2} - x_{2,1} \geq 8 \quad (9)$$

$$x_{2,4} - x_{2,2} \geq 9 \quad (10)$$

$$x_{2,end} - x_{2,4} \geq 6 \quad (11)$$

$$x_{3,3} - x_{3,1} \geq 8 \quad (12)$$

$$x_{3,2} - x_{3,3} \geq 8 \quad (13)$$

$$x_{3,end} - x_{3,2} \geq 8 \quad (14)$$

$$x_{4,2} - x_{4,1} \geq 4 \quad (15)$$

$$x_{4,3} - x_{4,2} \geq 2 \quad (16)$$

$$x_{4,end} - x_{4,3} \geq 2 \quad (17)$$

$$x_{5,2} - x_{5,1} \geq 4 \quad (18)$$

$$x_{5,3} - x_{5,2} \geq 9 \quad (19)$$

$$x_{5,5} - x_{5,3} \geq 4 \quad (20)$$

$$x_{5,end} - x_{5,5} \geq 5 \quad (21)$$

$$x_{6,3} - x_{6,1} \geq 6 \quad (22)$$

$$x_{6,end} - x_{6,3} \geq 4 \quad (23)$$

- to restrictions for the details priority processing:

$$x_{2,1} - x_{1,1} \geq 8 \quad (24)$$

$$x_{3,1} - x_{2,1} \geq 8 \quad (25)$$

$$x_{4,1} - x_{3,1} \geq 8 \quad (26)$$

$$x_{5,1} - x_{4,1} \geq 4 \quad (27)$$

$$x_{6,1} - x_{5,1} \geq 4 \quad (28)$$

$$x_{2,2} - x_{1,2} \geq 6 \quad (29)$$

$$x_{3,2} - x_{2,2} \geq 9 \quad (30)$$

$$x_{4,2} - x_{3,1} \geq 8 \quad (31)$$

$$x_{5,2} - x_{4,2} \geq 2 \quad (32)$$

$$x_{4,3} - x_{3,3} \geq 8 \quad (33)$$

$$x_{5,3} - x_{4,3} \geq 2 \quad (34)$$

$$x_{6,3} - x_{5,3} \geq 5 \quad (35)$$

$$x_{2,4} - x_{1,4} \geq 6 \quad (36)$$

$$x_{i,j} \geq 0 \quad (37)$$

The formulated task (5) – (37) 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. 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 (24) – (26) should be reformulated as:

$$x_{3,1} - x_{1,1} \geq 8 \quad (24)$$

$$x_{2,1} - x_{3,1} \geq 8 \quad (25)$$

$$x_{4,1} - x_{2,1} \geq 8 \quad (26)$$

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 (24) – (26) have to be changed as:

$$x_{1,1} - x_{2,1} \geq 8 \quad (24)$$

$$x_{3,1} - x_{1,1} \geq 8 \quad (25)$$

$$x_{4,1} - x_{3,1} \geq 8 \quad (26)$$

and again 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 $D_4 \rightarrow D_5 \rightarrow D_6 \rightarrow D_1 \rightarrow D_2 \rightarrow D_3$ the restrictions (24) – (36) are transformed to:

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

$$x_{6,1} - x_{5,1} \geq 4 \quad (25)$$

$$x_{1,1} - x_{6,1} \geq 6 \quad (26)$$

$$x_{2,1} - x_{1,1} \geq 8 \quad (27)$$

$$x_{3,1} - x_{2,1} \geq 8 \quad (28)$$

$$x_{5,2} - x_{4,2} \geq 2 \quad (29)$$

$$x_{1,2} - x_{5,2} \geq 9 \quad (30)$$

$$x_{22} - x_{12} \geq 6 \quad (31)$$

$$x_{3,2} - x_{2,2} \geq 9 \quad (32)$$

$$x_{5,3} - x_{4,3} \geq 2 \quad (33)$$

$$x_{6,3} - x_{5,3} \geq 5 \quad (34)$$

$$x_{3,3} - x_{6,3} \geq 4 \quad (35)$$

$$x_{2,4} - x_{1,4} \geq 6 \quad (36)$$

with the same objective function and the remaining restrictions.

In similar way, all possible combinations of detail processing sequences can be reflected in 24 different modifications of basic optimization task (5) – (37). The solving of all of the tasks can be done in parallel because all of the tasks are entirely independent of each other. The result of the solutions is 24 job shop schedules corresponding to different details processing sequences.

VI. Results Analysis and Discussion

The solutions of all optimization tasks corresponding to all possible combinations for details processing sequences along with their total makespan are shown in Table II.

The makespan values in tasks solutions vary within interval of 65 to 52 hours for different details processing sequences. Among them the optimal one with minimal makespan equal to 52 hours is for details processing sequence: D2 → D1 → D3 → D4 → D5 →D6.

The corresponding schedules for each processing sequence are illustrated in Fig. 3. For the described example, 11 different makespans have been distinguished that could not be determined by intuitive considerations. Increasing the number of independent details will increase the number of processing sequences and therefore the number of tasks that must be solved but because of parallel algorithm for solution of each task, this will not affect the computational complexity. Despite the fact that integer problems are difficult to solve (in general they are NP-hard), the formulated optimization problems and numerical testing show quite acceptable solution times of few seconds by means of LINGO solver [18].

All real-life job shop scheduling problems have their own specifics. When analyzing the resulting schedules it can be seen, 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 parallel solution of a set optimization tasks can be used for other similar problems concerning optimal job shop scheduling.

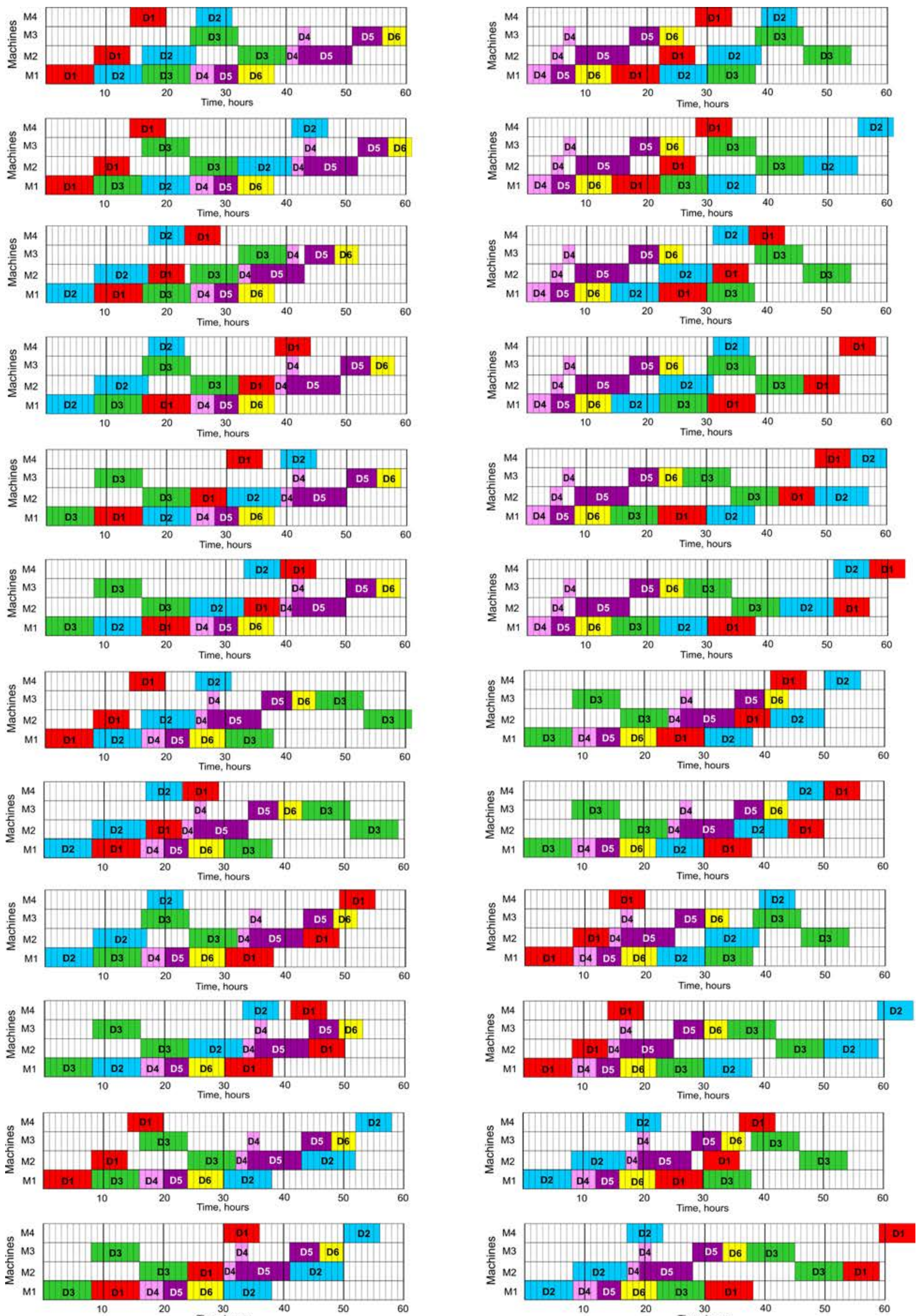

Fig. 3. Schedules for different details processing sequences

ISBN: 978-1-61804-236-1

TABLE II
SOLUTIONS RESULTS

Sequence of the details processing	Total makespan, hours
D_1, D2, D3, D4, D5, D6	60
D1, D3, D2, D4, D5, D6	61
D2, D1, D3, D4, D5, D6	**52**
D2, D3, D1, D4, D5, D6	58
D3, D1, D2, D4, D5, D6	59
D3, D2, D1, D4, D5, D6	59
D4, D5, D6, D_1, D2, D3	54
D4, D5, D6, D1, D3, D2	61
D4, D5, D6, D2, D1, D3	54
D4, D5, D6, D2, D3, D1	58
D4, D5, D6, D3, D1, D2	60
D4, D5, D6, D3, D2, D1	63
D1, D2, D4, D5, D6, D3	61
D2, D1, D4, D5, D6, D3	59
D3, D2, D4, D5, D6, D1	55
D2, D3, D4, D5, D6, D1	53
D1, D3, D4, D5, D6, D2	58
D3, D1, D4, D5, D6, D2	56
D3, D4, D5, D6, D1, D2	56
D3, D4, D5, D6, D2, D1	56
D1, D4, D5, D6, D3, D2	54
D1, D4, D5, D6, D2, D3	65
D2, D4, D5, D6, D1, D3	54
D2, D4, D5, D6, D3, D1	65
Minimal makespan: 52 hours	

VII. CONCLUSION

In this paper, a deterministic job shop scheduling approach for details processing on multiple machines based on integer linear programming model is described. The goal of described job shop scheduling is to determine the minimum makespan for a number of semi-dependant details (some with independent processing and other with dependant of each other processing) with different operations on different machines. To find the minimum of total makespan, a number of identical optimization tasks corresponding to all permutations of independent details processing sequences are formulated. The main contribution of the paper is using of the developed model in an algorithm based on solving of all formulated tasks in parallel. The execution of the algorithm provides a set of job shop optimal schedules for all possible details processing sequences. Then the best schedule and corresponding processing sequence in sense of minimal makespan are determined.

As extensions and future investigations, a possible direction is to explore how increasing number of identical machines will influence the algorithmic and computational difficulties.

For large scale job shop problems, where the total makespan could be essentially bigger, this approach can contribute not only to reduce the makespan via schedules optimization, but also to decrease the overall production time and costs.

ACKNOWLEDGMENT

The research work reported in the paper is partly supported by the project AComIn "*Advanced Computing for Innovation*", grant 316087, funded by the FP7 Capacity Programme (Research Potential of Convergence Regions).

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, no. 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, no. 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, no. 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, no. 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] 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.

[10] P. Baptiste and C. L. Pape, "Disjunctive constraints for manufacturing scheduling: principles and extensions", *Int. Journal of Computer Integrated Manufacturing*, vol. 9, no. 4, 1996, pp. 306-310.

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

[12] R. R. Weber, P. Varaiya and J. Walrand, "Scheduling jobs with stochastically ordered processing times on parallel machines to minimize expected flowtime", *Journal of Applied Probability*, vol. 23, no. 3, 1986, pp. 841-847.

[13] D. Golenko-Ginzburg and A. Gonik, "Optimal job-shop scheduling with random operations and cost objectives", *Int. Journal of Production Economics*, vol. 76, no. 2, 2002, pp. 147-157.

[14] S. R. Lawrence and E.C. Sewell, "Heuristic optimal static and dynamic schedules when processing times are uncertain", *Journal of Operations Management,* vol. 15, no. 1, 1997, pp. 71-82.

[15] 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.

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

[17] 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.

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