On cyclic job shop scheduling problem

Wojciech Bożejko
Department of Automatics, Mechatronics and Control Systems
Faculty of Electronics
Wrocław University of Science and Technology
Janiszewskiego 11-17, 50-372 Wrocław, Poland
e-mail: wojciech.bozejko@pwr.edu.pl

Mieczysław Wodecki
Telecommunications and Teleinformatics Department
Faculty of Electronics
Wrocław University of Science and Technology
Janiszewskiego 11-17, 50-372 Wrocław, Poland
e-mail: mieczyslaw.wodecki@pwr.edu.pl

***Abstract*—Cyclic problems are unique and little researched. They belong to a subclass of scheduling problems, as in fact they relate to the so-called irregular criterion. Cyclic problems are object of interest primarily due to their strong practical importance and difficulty in obtaining adequately efficient algorithms solving particular cases with additional constraints arising from manufacturing practice. In this paper we present a survey of papers considering cyclic scheduling in application to manufacturing systems and we propose a parallel approach to the minimum cycle time determination.**

***Index Terms*—scheduling, optimization, cyclic, manufacturing, parallel algorithm**

I. Introduction

Production systems implementing multi-assortment production, in large quantities, with slowly changing in time, product mix, manufacture products exclusively in a cyclic manner. Modern systems of this type satisfy market demand either by a uniform production in time (whose range varies irregularly according to market demand) or by cyclic output providing the right mix of assortments. The latter approach is more economically attractive as it eliminates or reduces the size of the stored finished goods. In addition, it provides a systematic inventory replenishment of relatively small amount of goods at suppliers and generates systematic demand for intermediates, raw materials from deliverers. The above system also simplifies the process of supply chain management. It also enables relatively easy detection of defects that may indicate a deterioration of the quality of the production system parameters and\or manufactured products. Optimization of the operation of such a system is reduced to minimizing of the cycle period, for a fixed mixture of tasks (products) in the cycle, which results in increased system capacity and improvement of machine utilization. Thus, recently, one can observe a significant increase of interest in the problems of cyclic tasks scheduling theory. For they are usually important and difficult, mostly NP-hard problems, from the standpoint of not only theory, but also practice.

II. Cyclic problems in manufacturing

Real production systems generate optimization problems of large size. For this reason, the attention of scientists has focused on the design of heuristic optimization algorithms. Currently, it is possible to observe an intensive development of methods based on artificial intelligence: tabu search, simulated annealing, genetic search, swarms, formic, gregarious, neural networks, and many more. The effectiveness of the algorithms can be increased in three ways: through the implementation of useful properties of the problem, the use of advanced methods of intensification and diversification of the solution space search process, the use of aggregation methods of calculation or parallel processing.

One of the main objectives of the work is to develop an original methodology of aggregation (consolidation of various objects in order to achieve synergies), which is a creative development of the block approach idea that has proven to be effective in the construction of algorithms for solving problems with not only regular, but also much more difficult, irregular goal functions. This methodology is based on the sustainable use of methods of exploration and exploitation of the solution space. On one hand it reduces the operating area of local search (through intensification) and on the other hand - bigger of strategic dispersal of the search (by diversification). Current methods of aggregation concern in particular:

- solution components,
- vector coordinates,
- moves (the so-called multi-moves in local search methods),
- solutions correlated with subsets of the solution space (i.e. representatives),
- population in algorithms inspired by nature,
- local vector components with local optima and certain values of the objective function,
- trees in algorithms solving transport problems, etc.

Their effects include:

- elimination criteria,
- relation of partial order on the set of solutions,
- filtration - heuristic overview of the solution space of the set (parameterized) accuracy,
- multi-moves transferring calculations in new areas of the solution space,
- patterns - some attributes of optimal solutions (similarly like DNA code fragments they allow inheritance of good or bad properties attributed to certain areas of the solution space), etc.

978-1-5386-1122-7/18/$31.00 ©2018 IEEE

The proposed methodology takes into account the aggregation of a number of specific properties of the considered problems. This requires an in-depth theoretical study and skillful use of the results. The no free lunch theorem shows that this is a prerequisite to include in specialized (above average) algorithm structures. The research will be carried out in terms of their use in the most promising currently methods of construction of algorithms to solve virtually unexplored so far cyclical problems.

The proposed work is interdisciplinary in its character. It embraces: research to determine the important parameters of the processes related to operational planning, construction and study of the properties of problems and models, as well as construction of new efficient algorithms inspired by nature and intelligent methods of local search. The novelty of the work lies in the construction of qualitatively new exact and approximate algorithms using the vector calculation environment, parallel and distributed having regard to the use of the equipment of the GPU (Graphics Processing Units), cards equipped with thousands of computing cores whose popularity has been growing recently.

Quality of the best solutions set by the approximate algorithms depends largely on the number of analyzed by them solutions, and thus on the computation time. The large size of practical problems and exponentially increasing computation time (of optimal algorithms) cause that even the ever-increasing processing power of computers is not able to have a significant impact on the increase in size of problems within an acceptable time. Recent work of the authors in this area concern the use of parallel computing environments for the corresponding optimization algorithms. The design of parallel algorithms can significantly increase the number of considered solutions (per unit time) effectively using a multiprocessor computational environment, such as GPU devices, computing clusters or supercomputers.

III. State of the art

Production tasks scheduling problems have been considered for at least 50 years, however the attention of scientists was focused mainly on classical models of manufacturing systems, such as the flow shop and the job shop problems. In these models, it is assumed that in each position there is one machine, buffers capacity is unlimited, residence times in buffers is unrestricted, transportation and setup times are ignored, small, etc. Additionally, in the literature, hybrid flow shop or job shop systems have been considered, in which at least in one position there was more than one machine. Hybrid systems are also known as flexible systems due to their flexibility to choose the technological route. Classical methods of modeling, common until recently, were good enough for companies involved in mass production of low diversity of products.

In real production systems, apart from sequence limitations resulting from technological route of tasks, there are additional constraints. Unfortunately, taking into account the constraints of the real systems requires the development of new models and new dedicated optimization algorithms.

While reviewing the literature on scheduling production tasks, it can be seen that there is a very big contrast between the number of works devoted to the classic problems of scheduling and scheduling problems with additional constraints. For instance, for the classic job shop problem there were hundreds of algorithms proposed based on different optimization methods, whereas for the job shop without buffers and the no-wait job shop problem - only a few. Starting from the beginning of the century, there has been a dramatic increase in the number of publications devoted to scheduling in production systems with additional constraints. In particular, the object of intensive research are now no-wait flow shop problems with no storage and with setups for which a large number of various types of heuristic algorithms was developed. In fact, the most common systems are hybrid systems: flow shop and job shop problems, which differ substantially from the classic flow shop problem. Therefore, they require the development of not only new, far more advanced models, but also new optimization algorithms.

Overview of scheduling no-wait no storage problems with constraints can be found in [18]. The classic no-wait job shop problem with constraints was the subject of research in [6], [12], [22], [23]. In the work [12] there was an original method for scheduling order (sequence of tasks) developed. Based on this method there was a local search algorithm developed which operates on the permutation of tasks. In [6] a tabu search algorithm was developed and similarly as in the previous case, in the construction of schedule the heuristic method was used. No-wait and no storage job shop system with constraints was considered in [17], [22], [27]. Work [27] also includes scheduling of automated transport trolleys.

In case of modeling of real production systems the attention should be focused primarily on models of hybrid systems. Operational scheduling in the hybrid flow shop problem without buffers with robotic centers were considered in [11]. Mixed integer programming model for hybrid flow shop problem with limited capacity has been proposed in [24]. In [19] the enhancement of algorithm based on the method of intelligent swarm search for hybrid flow shop problem with limited capacity buffers was proposed. An efficient algorithm for solving a no-wait hybrid flow shop problem with constraints was proposed in [26]. The algorithm is based on a genetic method and it should be noted that the algorithm was developed for a system comprising only two stages. The problem of scheduling in hybrid flow shop problem with limited waiting time was considered in [15]. To our knowledge, for the hybrid job shop problem with no-wait or with no storage constraints there are no computational models and efficient algorithms for optimization developed.

In the literature additional restrictions are usually considered in isolation. What follows from this assumption is that there are different methods for modeling and designing of heuristic algorithms. In particular, specific for the given constraint characteristics are used in the design of algorithms. For this reason, constructing algorithms for problems containing several constraints is not possible by a simple compilation of the characteristics of efficient algorithms for particular constraint. One of the papers in which a method for modeling of two no-wait no storage constraints is proposed is the work of [16]. In the work, for the manufacturing of cellular concrete blocks, there was proposed a graph model for production system in which both the no wait and no storage constraints were modeled in the form of additional arcs loaded respectively.

Optimization algorithms used for the production of additional systems can be used in other areas. The authors of [21] noted that trains vehicle routing problem can be modeled in the same way as hybrid job shop problem with no storage constraint.

Cyclic strategy for manufacturing is applied in flexible production systems using mass production. Long-term production plan is divided into uniform time intervals called cycles. In each cycle the same product mix is produced and the amount of each type depends on the overall demand for the given product. Cyclic production greatly simplifies the management of the production process because the plan set for the first cycle is reproduced for further periods. Production cycle helps to improve the quality of the planning and preparation of production through the experience gained during the implementation of the next cycle. It also simplifies supply chain management, as raw materials and intermediates needed for the production are required in repeated time points. Other important preferred feature of production cycle is prediction of a moment in which new series of products appear. This forecast enables systematic replenishment of own and contractors supplies.

Currently, the first works on algorithms supporting scheduling in cyclic production systems have started appearing. Undoubtedly, the majority of the papers is devoted to the most basic model of the production system, i.e. flow shop problem. Cyclic flow shop problem without additional restrictions is trivial, while the recurring instance of the case of no-wait constraint can be easily converted into an instance of the problem with the Cmax criterion (the time of job completion) and use one of the many existing algorithms. The flow shop problem with no storage constraint was the first problem requiring different, from the classical, way of modeling. Work [20] contains computational models for selected cyclic task scheduling problems, which can be used, among other issues, for instance in commercial optimization systems for problems of small size. The cyclical job shop problem without buffers was considered in [8]. In the work a tabu search algorithm in the form of a disjunctive graph was proposed.

The effect of the authors research on the cyclic production systems was the development of a cyclic graph models and optimization algorithms for cyclic flow shop problem without buffers [10], cyclic hybrid flow shop problem with setups [5], cyclic flow shop problem [2]. In case of the above mentioned problems there were certain properties identified which were used in the construction of efficient optimization algorithms. These properties are an extension of the block properties previously discovered for the classical scheduling problems.

It should be emphasized that, at present, there are no general methods for solving scheduling problems. Because of their multiplicity and specificity in the literature there is a large variety of specialized methods and ways of algorithms construction. Heuristic algorithms, in spite of strong progress in recent years, both in the theory of algorithms, as well as in terms of technologies, are often the only way to obtain solutions that are satisfactory from the point of view of practice, both as to the size of the resolved, within a reasonable time, examples, as well as to the goodness (distances from the optimal solution) of the obtained results. Techniques using multi-processor computers have definitely shorter history, but classic division of computer architectures seen from the present perspective was made a long time ago by Flynn [13]. However, it was only in the eighties that fast parallel algorithms appeared. Qualitative leap of designed algorithms appeared at a time when hardware manufacturers have figured out that a further increase of speed (clock frequency) of processors is already very expensive and this objective can be achieved much easier with the use of multi-core designs, i.e. in parallel computing environment. Todays processors by manufacturers like Intel and AMD have up to 15 cores (e.g. Intel Xeon E7-8890 v2), co-processor cards, such as Intel Xeon Phi - 61 cores of x86 type and GPUs (Graphic Processing Unit) used initially as exclusive graphics processors are now used as purely computational having even up to 2688 cores, e.g. nVidia Tesla K40 series products.

However, increasing of the number of cores requires the use of specially designed algorithms the start of metaheuristic sequential algorithm for multi-core processor will use only one core, which is only part of the potential of the equipment. The specificity of optimization algorithms and procedures for the determination of the key elements of an instance of the problem (e.g. the value of the objective function, which is usually defined as recursive) make the automatic parallelization method not satisfactory. Thus, there is a strong need for specialized algorithms designed specifically to run in parallel computing environment for specific types of problems and algorithms.

In case of positive findings, the proposed method can be used whenever the access to multi-core computers is possible (in recent times it concerns virtually any personal computer). The proposed methods are particularly likely to be advantageously used in scheduling systems in real time (e.g. in queuing systems in data centers) and especially in scheduling of production lines in large automated factories. For a large number of tasks, multithreaded methods applied to metaheuristics give a much better solution as the sequential algorithm performing calculations of the same type in the same amount of time (e.g. tabu search). However, if the use of exact algorithm is indispensable (e.g., B & B), multi-threaded version will work even p times faster, where p is the number of available cores.

IV. Cyclic job shop problem

Job shop scheduling problems follow from many real-world cases, which means that they have good practical applications as well as industrial significance.

Let us consider a set of jobs $\mathcal{J} = \{1, 2, \ldots, n\}$, a set of machines $M = \{1, 2, \ldots, m\}$ and a set of operations $\mathcal{O} = \{1, 2, \ldots, o\}$. The set $\mathcal{O}$ is decomposed into subsets connected with jobs. A job j consists of a sequence of o_j operations indexed consecutively by (l_{j-1}+1, l_{j-1}+2, ..., l_j) which have to be executed in this order, where $l_j = \sum_{i=1}^{j} o_i$ is the total number of operations of the first j jobs, j = 1, 2, ..., n, l_0 = 0, $\sum_{i=1}^{n} o_i = o$. An operation i has to be executed on machine $v_i \in M$ without any idleness in time $p_i > 0$, $i \in \mathcal{O}$. Each machine can execute at most one operation at a time. A feasible solution constitutes a vector of times of the operation execution beginning S= (S_1, S_2, ..., S_o) such that the following constraints are fulfilled

$$S_{l_{j-1}+1} \geq 0, \quad j = 1, 2, \ldots, n, \tag{1}$$

$$S_i + p_i \leq S_{i+1},$$

$$i = l_{j-1} + 1,\ \ l_{j-1} + 2, \ldots, l_j - 1,\ \ j = 1, 2, \ldots, n, \tag{2}$$

$$S_i + p_i \leq S_j \quad \text{or} \quad S_j + p_j \leq S_i,$$

$$i, j \in O, \quad v_i = v_j, \quad i \neq j. \tag{3}$$

Certainly, $C_j = S_j + p_j$. An appropriate criterion function has to be added to the above constraints. The most frequent are the following two criteria: minimization of the time of finishing all the jobs and minimization of the sum of job finishing times. From the formulation of the problem we have $\mathcal{C}_j \equiv C_{l_j}$, $j \in \mathcal{J}$.

In the cyclic manufacturing system a set of jobs (MPS – Minimal Part Set) is executed repeatedly in so-called production cycles. The cyclical nature of the process requires the following restriction:

$$S_i^t + p_i \leq S_i^{t+1} + T, \tag{4}$$

where t is a cycle number, $t = 1, 2, \ldots$,S_i^t denotes starting time of the operation i in t-th repeating (i.e. in t-th MPS) and T denotes a cycle time. The cyclic job shop problem consists in determination of schedule S such, that the cycle time T is minimal and constrains (1)–(4) are fulfilled. The problem is strongly NP-hard (see [1]).

V. Cyclic schedules

Solving job shop problem instances with the minimal cycle time criterion one ca meet schedules for which the value of the minimal cycle time is not an integer number, although the data – times of operations execution on machines – are integers. This is a kind of paradox and forces us to analyze a few (specifically: m) consecutive MPSs to determine the non-integer value of the minimum cycle time $T(pi)$. Examples of such situations are presented in Figure ref fig1 (value of the minimum cycle time: $T = 29.5$) and Figure ref fig5 (value of the minimum cycle time: $T = 29.333(3)$). The schedule

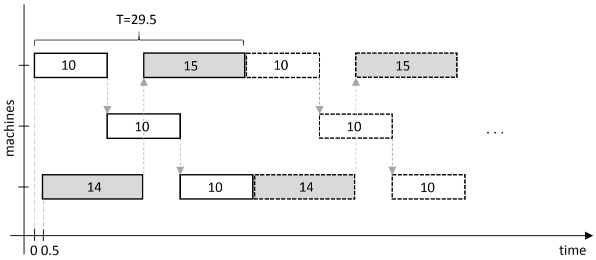

Fig. 1. Cyclic schedule.

shown in Figure ref fig1 requires the start of the first operation on the 3rd machine at time 0.5. Starting this operation at time 0 (Figure ref fig2) or 1 (Figure ref fig3) increases the minimum cycle time for the same sequence of operations on the machines to $T = 30$. Similarly, for the schedule shown in

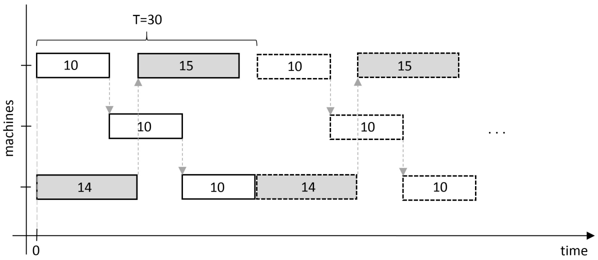

Fig. 2. Cyclic schedule.

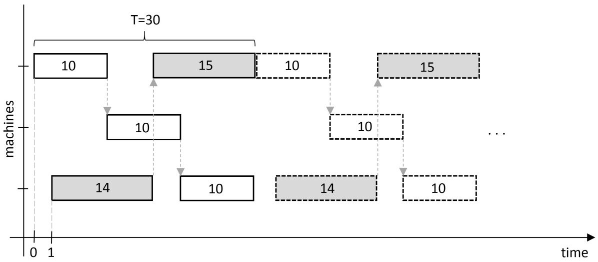

Fig. 3. Cyclic schedule.

Figure 4, starting of operation on machine 3 at a time other than $\frac{2}{3}$ and on machine 4 other than $\frac{1}{3}$, including integer values of the starting moments of these operations, the value of the minimum cycle time will be greater than $T = 29 + \frac{1}{3}$. In Figure 5 the trajectories of critical paths for the schedule presented in Figure 1 with cycle time $T = 29.5$. Setting a minimum cycle time requires observing where the critical path is returned to the same machine. The value of the minimum cycle time will be the sum of the values of weights on the critical path divided by 2 (they run – there are two – in this case, two MPSs), which for the given example gives the value $T = \frac{1}{2}(10{+}10{+}10{+}14{+}15) = 29.5$. A solution of the cyclic job shop can be represented by a permutation of operations from the set $\mathcal{O}$. Let Π be a set of all such permutations and let $T(\pi)$ denotes minimal cycle time for the solution represented

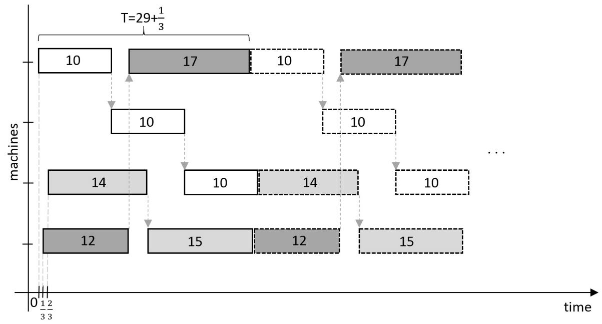

Fig. 4. Cyclic schedule.

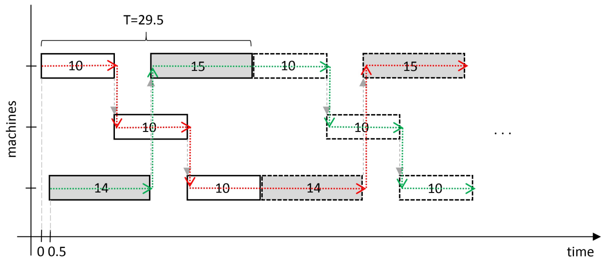

Fig. 5. Cyclic schedule.

by the permutation π. General method of the considered cyclic job shop problem can be presented as follows.

Let $\pi \in \Pi$; $\Theta \subset \Pi$; $\pi^* := \pi$; $T(\pi^*) := T(\pi)$;
repeat
Step 1. Generate a permutation $\beta \in \Theta$;
Step 2. Determine a cycle time $T(\beta)$;
if $T(\beta) < T(\pi^*)$ **then** $\pi^* := \beta$;
$\Theta := \Theta \setminus \{\beta\}$;
until $\Theta = \emptyset$.

If $\Theta = \Pi$ we have an exact algorithm, otherwise it is an approximated algorithm. If a generating function is based on neighborhoods researching, we obtain a local search algorithm.

The most time-consuming element of the algorithm is the procedure of the cycle time determination in the Step 2 of the algorithm. The run-time of the sequential method using the algorithm based on the searching the graph in accordance with the topological order is $O(m^2 o)$. We propose two methods formulated as theorems to determine in parallel the minimum cycle time for the operation order π. Before, we will need a fast method of parallel shortest paths determination.

A. Parallel determination of the longest paths in graphs

In the further part of the work a parallel algorithm for the shortest paths determination in graph will be proposed based on the matrix multiplication method (see Cormen et al. [9]), with the computational complexity $O(\log^2 n)$ using $O(n^3/\log n)$ processors. Algorithm is based on the idea of Gibbons and Rytter [14] developed in the work of Steinhöfel et al. [25] and then at Bożejko [7]. Since the algorithm requires simultaneous reading and there is no parallel write to the memory cells, the CREW PRAM model is used.

Let $A = [a_{ij}]^{n\times n}$ be the matrix of the distance of a given graph, which is the input of the algorithm. The output algorithm will return a matrix $A' = [a'_{ij}]^{n\times n}$ defined as follows:

$a'_{ij} = 0$ for $i = j$,
$a'_{ij} = \max\{a_{i_0 i_1} + a_{i_1 i_2} + \ldots + a_{i_{k-1} i_k}\}$ for $i \neq j$,

where the maximum is determined for all possible index strings $i_0, i_1, i_2, \ldots, i_k$, $i = i_0$ and $j = i_k$. The value a'_{ij} is the length of the longest path from i to j in the graph with the distance matrix A.

Algorithm ParLongestPaths(A,A')
Step 1. parfor all (i,j), $i,j = 1,2,\ldots,n$ **do**
$m_{ij} \leftarrow a_{ij}$
Step 2. repeat $\log_2 n$ **times**
parfor all (i,j,k), $i,j,k = 1,2,\ldots,n$ **do**
$q_{ijk} \leftarrow m_{ij} + m_{jk}$
parfor all (i,j), $i,j = 1,2,\ldots,n$ **do**
$m_{ij} \leftarrow \max\{m_{ij}, q_{i1j}, q_{i2j}, \ldots, q_{inj}\}$
Step 3. parfor all (i,j), $i,j = 1,2,\ldots,n$ **do**
if $i \neq j$ **then** $a'_{ij} \leftarrow m_{ij}$ **else** $a'_{ij} \leftarrow 0$

Computational complexity of the whole method is $O(\log^2 n)$ with using $O(n^3/\log n)$ processors.

B. Parallel minimum cycle time determination

Theorem 1: The value of the minimum cycle time in a cyclic job shop problem can be determined in the time $O(\log^2 o)$ on $O(o^3/\log o)$-processors CREW PRAM machine.

Proof. Let us construct a graph which represents minimal sums of operations durations between the first operations on each machine. For the example from the Fig. 5 it looks as in Fig. 6. The longest paths in the graph shown in Fig. 6

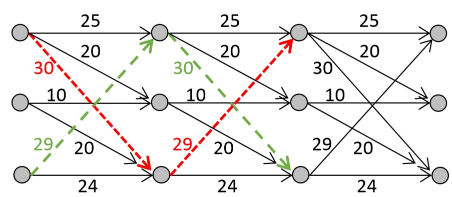

Fig. 6. Graph representing minimal sums of operations durations between the first operations on each machine.

can be determined using the **ParLongestPaths** algorithm in time $O(\log^2 m)$ on $O(m^3/\log m)$-processors CREW PRAM machine. As a result, we get a matrix of m^2 lower bounds of the minimum cycle time, from which the one – the largest – is the exact value of the minimum cycle time $T(\pi)$. A minimum of m^2 values can be determined in time $O(\log m^2) = O(2\log m) = O(\log m)$ on CREW PRAM with $O(m^2/\log m^2) = O(m^2/\log m) = O(m^3/\log m)$ processors. Unfortunately, it is necessary to take into account the time for determining weights in the graph which is $O(\log^2 o)$

on $O(o^3/\log o)$-processors CREW PRAM and determines the total complexity $O(\max\{\log^2 o, \log^2 m\}) = O(\log^2 o)$ and the number of processors $O(\max\{o^3/\log o, m^3/\log m)\}) = O(o^3/\log o)$ of the entire method.

Theorem 2: The value of the minimum cycle time in a cyclic job shop problem can be determined in the time $O(o+\log^2 m)$ using $O(m^3/\log m)$ - processors parallel computing environment with shared memory.

Proof. To determine the longest paths in the graph presented in Fig. 6, one should use the method based on vertex review in topological order. This method has a sequential complexity of $O(mo)$ resulting from the fact that m times one should perform the procedure of reviewing vertices in topological order, each time starting from the next source – the vertex representing the execution of the first operation on a machine. This procedure can be easily parallelized by using m processors obtaining parallel runtime $O(o)$. Then, the longest paths should be determined, which can be done in the time $O(\log^2 m)$ with using $O(m^3/\log m)$ processors and next determine the maximum from m^2 computed values, in time $O(\log m)$ using $O(m^2/\log m) = O(m^3/\log m)$ processors. The total parallel runtime will be $O(\max\{o, \log^2 m\}) = O(o + \log^2 m)$ and a number of processors $O(\max\{m, m^3/\log m)\}) = O(m^3/\log m)$.

VI. Conclusion

In the paper we present a survey of cyclic scheduling problems together with a proposition of two methods of determination of cyclic schedules in parallel way. We analyze cyclic schedule from the cycle time value point of view and observe that integer data values does not implies integer values of the minimal cycle time. We propose two parallel procedures with different complexities to the minimal cycle time determination designed to be run on computing systems with shared memory – GPU or multicore CPU.

Acknowledgement

The paper was partially supported by the National Science Centre of Poland, grant OPUS no. DEC 2017/25/B/ST7/02181.

References

[1] W. Bożejko, A. Gnatowski, J. Pempera, M. Wodecki, "Parallel tabu search for the cyclic job shop scheduling problem", Computers & Industrial Engineering 113, pp. 512–524, 2017.
[2] W. Bożejko, A. Gnatowski, R. Idzikowski, M. Wodecki, "Cyclic flow shop scheduling problem with two-machine cells", in: Special issue on Discrete Systems: Theory and Applications (Bożejko W., Wodecki M., Eds.), Archives of Control Sciences, Vol. 27, No. 2., pp. 151–168, 2017.
[3] W. Bożejko, J. Pempera, M. Wodecki, "A fine-grained parallel algorithm for the cyclic flexible job shop problem", in: Special issue on Discrete Systems: Theory and Applications (Bożejko W., Wodecki M., Eds.), Archives of Control Sciences Vol. 27, No. 2., pp. 169–182, 2017.
[4] W. Bożejko, Z. Chaczko, M. Uchroński, M. Wodecki, "Parallel patterns determination in solving cyclic flow shop problem with setups",in: Special issue on Discrete Systems: Theory and Applications (Bożejko W., Wodecki M., Eds.), Archives of Control Sciences, Vol. 27, No. 2., pp. 183–196, 2017.
[5] W. Bożejko, Gniewkowski L., Pempera J., Wodecki M., "Cyclic hybrid flow-shop scheduling problem with machine setups", Procedia Computer Science Volume 29, pp. 2127–2136, 2014.
[6] W. Bożejko , Makuchowski M., "A fast hybrid tabu search algorithm for the no-wait job shop problem", Computers & Industrial Engineering 56, 4, pp. 1502–1509, 2009.
[7] W. Bożejko, "Solving the flow shop problem by parallel programming", Journal of Parallel and Distributed Computing 69, 470–481, 2009.
[8] P. Brücker, T. Kampmeyer, "Cyclic job shop scheduling problems with blocking", Annals of Operations Research 159, pp. 161–181, 2008.
[9] T.H. Cormen, Leiserson C.E., Rivest R.L., "Introduction to Algorithms", 2nd revised edition, MIT Press, 2001.
[10] P. Dąbrowski, Pempera J., Smutnicki C., "Minimizing cycle time of the flow line-genetic approach with gene expression", Lecture Notes in Computer Science Vol. 4431, pp. 194–201, 2007.
[11] A. Elmi, S. Topaloglu, "A scheduling problem in blocking hybrid flow shop robotic cells with multiple robots", Computers & Operations Research 40, 10, pp. 2543–2555, 2013.
[12] J. M. Framinan, Ch. Schuster, "An enhanced timetabling procedure for the no-wait job shop problem: a complete local search approach", Computers & Operations Research 33, 5, pp. 1200–1213, 2006.
[13] Flynn M.J., "Very high-speed computing systems", Proceedings of the IEEE 54, pp. 1901–1909, 1966.
[14] A. Gibbons, Rytter W., "Efficient parallel algorithms", Cambridge University Press, 1988.
[15] C. Gicquel, L. Hege, M. Minoux, W. van Canneyt, "A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints", Computers & Operations Research 39, 3, pp. 629–636, 2012.
[16] J. Grabowski, J.Pempera, "Sequencing of jobs in some production system", European Journal of Operational Research 125, 3, pp. 535–550, 2000.
[17] H. Gröflin, A. Klinkert, "A new neighborhood and tabu search for the Blocking Job Shop", Discrete Applied Mathematics 157, 17, pp. 3643–3655, 2009.
[18] N.G. Hall, Sriskandarajah C. "A survey of machine scheduling problems with blocking and no-wait in process", Operations Research 44, pp. 510-525, 1996.
[19] J. Li, Q. Pan, "Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm", Information Sciences 316, pp. 487–502, 2015.
[20] Kampmeyer T. "Cyclic scheduling problems", PhD theses, University of Osnabruck, Germany, 2006.
[21] S.Q. Liu, E. Kozan, "Scheduling trains as a blocking parallel-machine job shop scheduling problem", Computers & Operations Research 36, 10, pp. 2840–2852, 2009.
[22] A. Mascis, D. Pacciarelli, "Job-shop scheduling with blocking and no-wait constraints", European Journal of Operational Research 143, 3, pp. 498–517, 2002.
[23] J. Pan, H. Huang, "A hybrid genetic algorithm for no-wait job shop scheduling problems", Expert Systems with Applications 36, 3, 2, pp. 5800–5806, 2009.
[24] T. Sawik, "Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers", Mathematical and Computer Modelling 31, 13, pp. 39–52, 2000.
[25] K. Steinhöfel, Albrecht A., Wong C.K., "Fast parallel heuristics for the job shop scheduling problem", Computers & Operations Research 29, pp. 151–169, 2002.
[26] S. Wang, M. Liu, "A genetic algorithm for two-stage no-wait hybrid flow shop scheduling problem", Computers & Operations Research 40, 4, pp. 1064–1075, 2013.
[27] C. Zeng, J.Tang, Ch. Yan, "Scheduling of no buffer job shop cells with blocking constraints and automated guided vehicles", Applied Soft Computing 24, pp. 1033–1046, 2014.