Computers & Industrial Engineering 81 (2015) 158–166

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Block approach to the cyclic flow shop scheduling

Wojciech Bożejko [a,*], Mariusz Uchroński [b], Mieczysław Wodecki [c]

[a] Department of Automatics, Mechatronics and Control Systems, Faculty of Electronics, Wrocław University of Technology, Janiszewskiego 11-17, 50-372 Wrocław, Poland
[b] Wrocław Centre of Networking and Supercomputing, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
[c] Institute of Computer Science, University of Wrocław, Joliot-Curie 15, 50-383 Wrocław, Poland

ARTICLE INFO

Article history:
Received 31 May 2014
Received in revised form 10 November 2014
Accepted 5 January 2015
Available online 16 January 2015

Keywords:
Scheduling
Cyclic flow shop problem
Block properties

ABSTRACT

The cyclic flow show problem with machine setups is considered in this paper. It relies in producing of a set of certain elements in fixed intervals of time (cycle time). Process optimization is reduced to minimization of cycle time, i.e., the time after which the next batch of the same elements may be produced. Since the problem is strongly NP-hard, in order to solve it an approximate algorithm was used. There is presented a graph model of a problem and the so called block eliminating properties capable of reducing, in a significant way, neighborhood used in the tabu search algorithm. Conducted computational experiments confirm high efficiency of the proposed technique.

1. Introduction

For many years, one could observe an increasing market demand for diversity (multiassortment) of production. This may be provided, among many other issues, by means of cyclic production. In fixed intervals of time (cycle time) a certain 'batch' of assortment (a mix of kit, a set) is produced. Process optimization is typically reduced to minimization of cycle time. Proper selection of mix and cycle time enables not only to meet demand, but also to improve efficacy and effectiveness of machinery use. 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.

A comprehensive overview of the state of knowledge concerning the cyclic task scheduling problem can be found in the work of Levner, Kats, Lopez, and Cheng (2010) analyzing the issues of computational complexity of algorithms for solving various types of cycle scheduling problem. Here, in particular NP-difficult problems of various cyclic types including a variety of criterion functions and additional constraints (*no wait*, *no buffer*, etc.) are considered.

In the scientific work by Panwalkar, Dudek, and Smith (1973) on task scheduling it was found that 75% of problems occurring in practice requires at least one setup dependent on the order of tasks execution. However, in 15% of the problems a setup of all tasks should be taken into consideration. Nevertheless, in the vast majority of works, in the field of scheduling setups are not taken into account at all. This applies both to single and multi-machine problems and to different goal functions.

Cyclic problems belong to unique, relatively little researched subclass of scheduling problems. However, more and more practitioners and theorists show interest in the above issues mainly due to their great practical importance and the attempt to overcome difficulties in constructing relatively efficient algorithms. Strong NP-hardness of the simplest versions of the above problem limits the scope of applications of exact algorithms only to instances of small size.

In this paper a multi-machine cyclic production system is considered, in which any element of the fixed batch (mix) passes successively through each of the machines (permutation flow shop, see Nowicki & Smutnicki (1996) and Grabowski & Wodecki (2004)). Between successively produced elements there must be a setup of machines performed. The problem consists in finding minimization of cycle time, i.e. the time after which the next batch of the same elements may be produced. There will be proven *strong NP-hardness* already for some special case of the problem under consideration. The hardness of the problem may be confirmed by the fact that some simplified version boils down to solving the problem of a traveling salesman. For this reason, in order to effectively determine solutions a fast approximate algorithm is used. There will be also the so called 'block elimination properties' proven which will be used in the construction of tabu search algorithm. They provide an indirect search of certain subsets of solution space not only accelerating calculations, but also, at the same time, improving quality of designated solutions.

* Corresponding author.
E-mail addresses: wojciech.bozejko@pwr.edu.pl (W. Bożejko), mariusz.uchronski@pwr.edu.pl (M. Uchroński), mwd@ii.uni.wroc.pl (M. Wodecki).

http://dx.doi.org/10.1016/j.cie.2015.01.004
0360-8352/

Continuous flow production systems are among the most commonly encountered in industry. Everywhere where the production process consists of the following successive stages, one deals with such systems. Each stage of production is realized in a separate slot supplied with specialized machinery. In the literature, there are also other names describing this problem such as: a hybrid flow-shop system or a flexible production line. The hybrid flow-shop problem with setups was considered, among many others, in the works (Bożejko, Gniewkowski, Pempera, & Wodecki, 2014; Cavory, Dupas, & Goncalves, 2005; Caggiano & Jackson, 2008; Dbrowski, Pempera, & Smutnicki, 2007; Fournier, Lopez, & Lan Sun Luk, 2002; Kampmeyer, 2006; Sawik, 2014; Sawik, 2012).

The work consists of six chapters. The first and the second chapter, based on literature results, include a brief introduction and basic definitions related to cyclic scheduling tasks. The next two chapters constitute the new, genuine results of the authors. There are presented and proven the so called 'block properties' enabling elimination of certain elements from the neighborhood of tabu search algorithm. The last two chapters include the results of computational experiments and conclusions.

2. Problem formulation

Considered in the paper system of manufacturing is an extension of *strongly NP-hard*, classical in theory of scheduling, permutation flow problem (denoted in literature by $F^*||C_{\max}$). It can be formulated as follows:

Problem: There is given a set of n tasks $\mathcal{J} = \{1, 2, \ldots, n\}$, to be carried out recurrently (in a repeated manner) on machines from the set $\mathcal{M} = \{1, 2, \ldots, m\}$. Any task should be performed consecutively, on each m machine $1, 2, \ldots, m$ (technological line). The task $j \in \mathcal{J}$ is a sequence m of operations $O_{1,j}, O_{2,j}, \ldots, O_{m,j}$. The operation $O_{k,j}$ corresponds to the activity of execution of j task on machine k, in time $p_{k,j}$ $(k = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n)$. After completion of certain operation and before the start of the next one there must be a setup of machine performed. Let $s^k_{i,j}$ $(k \in \mathcal{M},\ i \neq j\ i, j \in \mathcal{J})$ be a time of a setup of machine k between operation $O_{k,i}$ and $O_{k,j}$. There must be the order of tasks execution (the same on each machine) designated, which minimizes cycle time, i.e. the time of the beginning of tasks execution from the set $\mathcal{J}$ in the next cycle. The following restrictions must be fulfilled:

(a) each operation can be performed only by one determined by the production process, machine,
(b) no machine can perform at the same time more than one operation,
(c) production process order of operations execution must be preserved,
(d) execution of any operation cannot be interrupted before its completion,
(e) between successively executed, on the same machine, operations there must be a setup performed,
(f) each task is performed sequentially after the completion of cycle time.

The set of tasks $\mathcal{J}$ executed in a single cycle is called (*minimal part set*) – MPS. MPSs are processed directly one after the other in a cyclic manner. In each of the MPSs the tasks from the set $\mathcal{J}$ are performed on each machine in the same order (permutation flow shop). Thus, any order of tasks on machines can be represented by a permutation $\pi = (\pi(1), \ldots, \pi(n))$ of elements from the set $\mathcal{J}$. Let Φ be the set of all such permutations.

The considered in the paper problem boils down to such determining of the tasks permutations (i.e. moments of tasks execution start on machines that meet the constraints (a)–(f), that the cycle time (time after which any task is performed in the next MPS-e) was minimal. In brief, this problem will be denoted by **CFS**.

2.1. Mathematical model

Let $\pi \in \Phi$ be an order of tasks execution on machines (the same for all MPSs). By $[S^h]_{m \times n}$ it is denoted the matrix of the beginning of tasks execution in h-th MPS, where $S^h_{i,\pi(j)}$ is the starting time of task $\pi(j)$ on machine i in h-th MPS. It is assumed that not only the sequence of tasks is cyclically repeated in each of the MPSs, but that timetable of system operation (i.e., execution of the following MPSs) is cyclic. This means that there is a constant (period) $T(\pi)$ such that

$$S^{h+1}_{i,\pi(j)} = S^h_{i,\pi(j)} + T(\pi),\ \ i = 1, \ldots, m,\ j = 1, \ldots, n,\ h = 1, 2, \ldots \tag{1}$$

The period $T(\pi)$ is undeniably dependent on permutation π and is called *cycle time* of the system for the permutation $\pi \in \Phi$. The minimum value $T(\pi)$ will be called *minimum cycle time* and will be denoted by $T^*(\pi)$.

Optimal value of time of the cycle $T^*(\pi^*)$ (solution to the problem **CFS**) can be determined by solving the following optimization task: designate

$$T^*(\pi^*) = \min\{T^*(\pi) :\ \pi \in \Phi\}, \tag{2}$$

with constraints:

$$S^h_{i,\pi(j)} + p_{i,\pi(j)} \leqslant S^h_{i+1,\pi(j)},\ \ i = 1, \ldots, m-1,\ j = 1, \ldots, n, \tag{3}$$

$$S^h_{i,\pi(j)} + p_{i,\pi(j)} + s^i_{\pi(j),\pi(j+1)} \leqslant S^h_{i,\pi(j+1)},\ \ i = 1, \ldots, m,\ j = 1, \ldots, n-1, \tag{4}$$

$$S^h_{i,\pi(n)} + p_{i,\pi(n)} + s^i_{\pi(n),\pi(1)} \leqslant S^{h+1}_{i,\pi(1)},\ \ i = 1, \ldots, m, \tag{5}$$

$$S^{h+1}_{i,\pi(j)} \leqslant S^h_{i,\pi(j)} + T^*(\pi),\ \ i = 1, \ldots, m-1, \tag{6}$$

where $h = 1, 2, \ldots$.

The last constraint (6) is characteristic for cyclic production as it determines the relationship between beginning times of tasks execution in successively performed MPSs.

Without loss of generality, we can assume that the starting time of the first task performance on the first machine in the first MPS is $S^1_{1,\pi(1)} = 0$. For a fixed permutation $\pi \in \Phi$ and the first MPS, by

$$T_k(\pi) = \sum_{i=1}^{n-1} (p_{l,\pi(i)} + s^k_{\pi(i),\pi(i+1)}) + p_{l,\pi(n)} + s^k_{\pi(n),\pi(1)} \tag{7}$$

it is denoted the time of the tasks execution in order π, together with setups on k-th machine (this sum also includes setup time between the last operation $\pi(n)$ of the first MPS, and the first $\pi(1)$ operation of the second MPS). In short, this time will be called k-th *peak*.

It is easy to see that for permutation of tasks $\pi \in \Phi$ minimum cycle time is

$$T^*(\pi) = \max\{T_k(\pi) :\ k = 1, 2, \ldots, m\}. \tag{8}$$

2.2. The problem with zero setup time

Let us consider a simplified version of the problem **CFS**, in which machine setup times, between successively performed operations, are equal to zero. Thus, for any permutation $\pi \in \Phi$, $s^k_{\pi(i),\pi(i+1)} = 0,\ i = 1, 2, \ldots, n-1$ and $s^k_{\pi(n),\pi(1)} = 0,\ k = 1, 2, \ldots, m$. Then (7), time of the operation execution by k-th machine is