Cyclic Scheduling in Robotic Cells: An Extension of Basic Models in Machine Scheduling Theory

Eugene Levner[1], Vladimir Kats[2] and David Alcaide López De Pablo[3]
[1]Holon Institute of Technology, Holon, [2]Institute of Industrial Mathematics, Beer-Sheva,
[3]University of La Laguna, La Laguna, Tenerife
[1, 2] Israel, [3]Spain

1. Introduction

There is a growing interest on cyclic scheduling problems both in the scheduling literature and among practitioners in the industrial world. There are numerous examples of applications of cyclic scheduling problems in different industries (see, e.g., Hall (1999), Pinedo (2001)), automatic control (Romanovskii (1967), Cohen et al. (1985)), multi-processor computations (Hanen and Munier (1995), Kats and Levner (2003)), robotics (Livshits et al. (1974), Kats and Mikhailetskii (1980), Kats (1982), Sethi et al. (1992), Lei (1993), Kats and Levner (1997a, 1997b), Hall (1999), Crama et al. (2000), Agnetis and Pacciarelli (2000), Dawande et al. (2005, 2007)), and in communications and transport (Dauscha et al. (1985), Sharma and Paradkar (1995), Kubiak (2005)). It is, perhaps, a surprising thing that many facts in scheduling theory obtained as early as in the 1960s, are re-discovered and re-rediscovered by the next generations of researchers. About two decades ago, this fact was noticed by Serafini and Ukovich (1989).

The present survey uniformly addresses cyclic scheduling problems through the prism of the classical machine scheduling theory focusing on their features that are common for all aforementioned applications. Historically, the scheduling literature considered periodic machine scheduling problems in two major classes – called *flowshop* and *jobshop* - in which setup and transportation times were assumed insignificant. Indeed, many machining centers can quickly switch tools, so the setup times for these situations may be small or negligible. There are a lot of results about cyclic flowshop and jobshop problems with negligible setup/transportation times. Advantages of cyclic scheduling policies over conventional (non-cyclic) scheduling in flexible manufacturing are widely discussed in the literature, we refer the interested reader to Karabati and Kouvelis (1996), Lee and Posner (1997), Hall et al. (2002), Seo and Lee (2002), Timkovsky (2004), Dawande et al. (2007), and numerous references therein.

At the same time, modern flexible manufacturing systems are supplied by computer-controlled hoists, robots and other material handling devices such that the transportation and setup operation times are significant and should not be ignored. Robots have become a standard tool to serve cyclic transportation and assembling/disassembling processes in manufacturing of airplanes, automobiles, semiconductors, printed circuit boards, food

Source: Multiprocessor Scheduling: Theory and Applications, Book edited by Eugene Levner,
ISBN 978-3-902613-02-8, pp.436, December 2007, Itech Education and Publishing, Vienna, Austria

products, pharmaceutics and cosmetics. Robots have expanded production capabilities in the manufacturing world making the assembly process faster, more efficient and precise than ever before. Robots save workers from tedious and dull assembly line jobs, and increase production and savings in the processes. As larger and more complex robotic cells are implemented, more sophisticated planning and scheduling models and algorithms are required to perform and optimize these processes.

The cyclic scheduling problems, in which setup operations are performed by automatic transporting devices, constitute a vast subclass of cyclic problems. Robots or other automatic devices are explicitly introduced into the models and treated as special purpose machines. In this chapter, we will focus on three major classes of cyclic scheduling problems – flowshop, jobshop, and parallel machine shop.

The chapter is structured as follows. Section 2 is a historical overview, with the main attention being paid to the early works of the 1960s. Section 3 recalls three orthodox classes of scheduling theory: flowshop, jobshop, and PERT-shop. Each of these classes can be extended in two directions: (a) for describing periodic processes with negligible setups, and (b) for describing periodic processes in robotic cells where setups and transportation times are non-negligible. In Section 4 we consider an extension of the cyclic PERT-shop, called the cyclic FMS-shop and demonstrate that its important special case can be solved efficiently by using a graph approach. Section 5 concludes the chapter.

2. Brief Historical Overview

Cyclic scheduling problems have been introduced in the scheduling literature in the early 1960s, some of them assuming setup/transportation times negligible while other explicitly treating material handling devices with non-negligible operation times.

Cyclic Flowshop. Cuninghame-Greene (1960, 1962) has described periodic industrial processes, which in today's terminology might be classified as a *cyclic flowshop* (without setups and robots), and suggested an algebraic method for finding minimum cycle time using matrix multiplication in which one writes "addition" in place of multiplication and operation "max" instead of addition. This (max, +)–algebra has become popular in the 1980s (see, e.g. Cuninghame-Greene (1979), Cohen et al. (1985), Baccelli et al. (1992)) and is presently used for solving the cyclic flowshop without robots, see, e.g., Hanen (1994), Hanen and Munier (1995), Lee (2000), and Seo and Lee (2002).

Independently of the latter research, Degtyarev and Timkovsky (1976) and Timkovsky (1977) have studied so-called *spyral cyclograms* widely used in the Soviet electronic industry; they introduced a generalized shop structure which they called a "*cycle shop*". Using a more standard terminology, we might say that these authors have been the first to study a *flowshop with reentrant machines* which includes, as special cases, many variants of the basic flowshop, for instance, the reentrant flowshop of Graves et al. (1983), V-shop of Lev and Adiri (1984), cyclic robotic flowshop of Kats and Levner (1997, 1998, 2002). The interested reader is referred to Middendorf and Timkovsky (2002) and Timkovsky (2004) for more details.

Cyclic Robotic Flowshop. In the beginning of 1960s, a group of Byelorussian mathematicians (Suprunenko et al. (1962), Aizenshtat (1963), Tanaev (1964), and others) investigated cyclic processes in manufacturing lines served by transporting devices. The latters differ from other machines in their physical characteristics and functioning. These authors have introduced a cyclic robotic flowshop problem and suggested, in particular, a combinatorial method called *the method of forbidden intervals* which today is being developed further by different authors for various cyclic robotic scheduling problems (see, for example, Livshits et al. (1974), Levner et al. (1997), Kats et al. (1999), Che and Chu (2005a, 2005b), Chu (2006), Che et al. (2002, 2003)). A thorough review in this area can be found in the surveys by Hall (1999), Crama et al. (2000), Manier and Bloch (2003), and Dawande et al. (2005, 2007).

Cyclic PERT-shop. The following cyclic PERT-shop problem has originated in the work by Romanovskii (1967). There is a set S of n partially ordered operations, called *generic operations*, to be processed on machines. As in the classic (non-cyclic) PERT/CPM problem, each operation is done by a dedicated machine and there is sufficiently many machines to perform all operations; so the question of scheduling operations on machines vanishes. Each operation i has processing time $p_i > 0$ and must be performed periodically with the same period T, infinitely many times.

For each operation i, let $<i, k>$ denote the kth execution (or, repetition) of operation i in a schedule (here k is any positive integer). *Precedence relations* are defined as follows (here we use a slightly different notation than that given by Romanovskii). If a generic operation i precedes a generic operation j, the corresponding *edge* (i, j) is introduced. Any edge (i,j) is supplied by two given values, L_{ij} called the *length*, or *delay*, and H_{ij} called the *height* of the corresponding edge (i, j). The former value is any rational number of any sign while the latter is integer. Then, for a pair of operations i and j, and the given length L_{ij} and height H_{ij}, the following relations are given: for all $k \geq 1$, $t(i,k) + L_{ij} \leq t(j, k + H_{ij})$, where $t(i,k)$ is the starting time of operation $<i, k>$. An edge is called *interior* if its end-nodes belong to the same iteration (or, one can say "to the same block, or pattern") and *backward* (or, *recycling*) if its end-nodes belong to two consecutive blocks.

A schedule is called *periodic* (or *cyclic*) with *cycle time* T if $t(i, k) = t(i,1) + (k-1)T$, for all integer $k \geq 1$, and for all $i \in S$ (see Fig. 1). The problem is to find a periodic schedule (i.e., the starting time $t(i,1)$ of operations) providing a minimum cycle time T, in a graph with the infinite number of edges representing an infinitely repeating process.

Figure 1. The cyclic PERT graph (from Romanovskii, (1967))

In the above seminal paper of 1967, Romanovskii proved the following claims which have been rediscovered later by numerous authors.

- **Claim 1**. Let the heights of interior edges be 0 and the heights of backward edges 1. The minimum cycle time in a periodic PERT graph with the infinite number of edges is equal to the maximum circuit ratio in a corresponding double-weighted finite graph in which the first weight of the arc is its length and the second is its height: $T_{\min} = \max_C \sum L_{ij}/\sum H_{ij}$, where maximum is taken over all circuits C; $\sum L_{ij}$ denotes the total circuit length, and $\sum H_{ij}$ the total circuit height.

- **Claim 2**. The max circuit ratio problem and its version, called the max mean cycle problem, can be reformulated as linear programming problems. The dual to these problems is the parametric critical path problem.
- **Claim 3.** The above problems, namely, the max circuit ratio problem and the max mean cycle problem, can be solved by using the iterative Howard-type dynamic programming algorithm more efficiently than by linear programming. (The basic Howard algorithm is published in Howard (1960)).
- **Claim 4.** Mean cycle time counted for n repetitions of the first block in an optimal schedule differs from the optimal mean cycle time by $O(1/n)$.

The interested reader can find these or similar claims discovered independently, for example, in Reiter (1968), Ramchandani (1973), Karp (1978), Gondran and Minoux (1985), Cohen et al. (1985), Hillion and Proth (1989), McCormick et al. (1989), Chretienne (1991), Lei and Liu (2001), Roundy (1992), Ioachim and Soumis (1995), Lee and Posner (1997), Hanen (1994), Hanen and Munier (1995), Levner and Kats (1998), Dasdan et al. (1999), Hall et al. (2002). In recent years, the cyclic PERT-shop has been studied for more sophisticated modifications, with the number of machines limited and resource constraints added (Lei (1993), Hanen (1994), Hanen and Munier (1995), Kats and Levner (2002), Brucker et al. (2002), Kampmeyer (2006)).

3. Basic Definitions and Illustrations

In this section, we recall several basic definitions from the scheduling theory. Machine scheduling is the allocation of a set of machines and other well-defined resources to a set of given jobs, consisting of operations, subject to some pre-determined constraints, in order to satisfy a specific objective. A problem instance consists of a set of m machines, a set of n jobs is to be processed sequentially on all machines, where each operation is performed on exactly one machine; thus, each job is a set of operations each associated with a machine.

Depending on how the jobs are executed at the shop (i.e. what is the routing in which jobs visit machines), the manufacturing systems are classified as:

- flow shops, where all jobs are performed sequentially, and have the same processing sequence (routing) on all machines, or
- job shops, where the jobs are performed sequentially but each job has its own processing sequence through the machines,
- parallel machine shop, where sequence of operations is partially ordered and several operations of any individual job can be performed *simultaneously* on several parallel machines.

Formal descriptions of these problems can be found in Levner (1991, 1992), Tanaev et al. (1994a, 1994b), Pinedo (2001), Leung (2004), Shtub et al. (1994), Gupta and Stafford (2006), Brucker (2007), Blazewicz et al. (2007). We will consider their cyclic versions.

The cyclic shop problems are an extension of the classical shop problems. A problem instance again consists of a set of m machines and a set of n jobs (usually called *products,* or *part types*) which is to be processed sequentially on all machines. The machines are requested to process repetitively a *minimal part set*, or *MPS*, where the MPS is defined as the smallest integer multiple of the periodic production requirements for every product. In other words, let $r = (r_1, r_2, \dots, r_n)$ be the production requirements vector defining how many units of each product (j=1,…,n) are to be produced over the planning horizon. Then the MPS

is the vector $r_{MPS} = (r_1/q, r_2/q, \dots, r_n/q)$ where q is the greatest common divisor of integers $r_1, r_2, \dots, r_n$. Identical products of different, periodically repeated, replicas of the MPS have the same processing sequences and processing times, whereas different products within an MPS may require different processing sequences of machines and the processing times. The replicas of the MPS are processed through equal time intervals T called *cycle time* and in each cycle, exactly one MPS's replica is introduced into the process and exactly one MPS's replica is completed.
An important subclass of cyclic shop problems are the robotic scheduling problems, in which one or several robots perform transportation operations in the production process. The robot can be considered as an additional machine in the shop whose transportation operations are added to the set of processing operations. However, this "machine" has several specific properties: (i) it is *re-entrant* (that is, any product requires the utilization of the same robot several times during each cycle) and (ii) its setup operations, that is, the times of empty robots between the processing machines, are *non-negligible*.

3.1. Cyclic Robotic Flowshop

In the cyclic robotic flowshop problem it is assumed that a technological processing sequence (route) for n products in an MPS is the same for all products and is repeated infinitely many times. The transportation and feeding operations are done by robots, and the sequences of the robotic operations and technological operations are repeated cyclically. The objective is to find the cyclic schedule with the maximum productivity, that is, the minimum cycle time. In the general case, the robot's route is not given and is to be found as a decision variable.
A possible layout of the cyclic robotic flowshop is presented in Fig. 2.

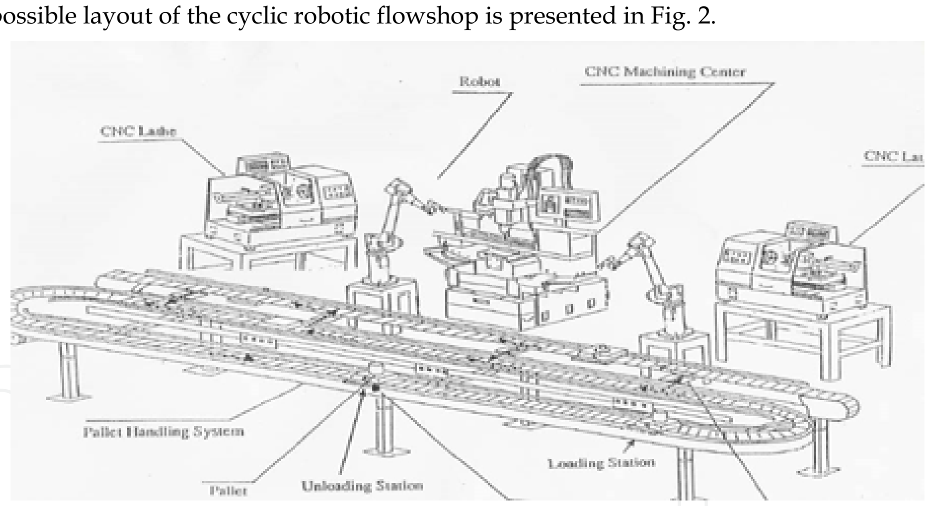


Figure 2. Cyclic Robotic Flowshop

A corresponding Gantt chart depicting coordinated movement of parts and robot is given in Fig. 3. Machines 0 and 6 stand for the loading and unloading stations, correspondingly. Three identical parts are introduced into the system at time 0, 47 and 94, respectively. The bold horizontal lines depict processing operations on the machines while a thin line depicts the route of a single robot between the processing machines. More details can be found in Kats and Levner (1998).

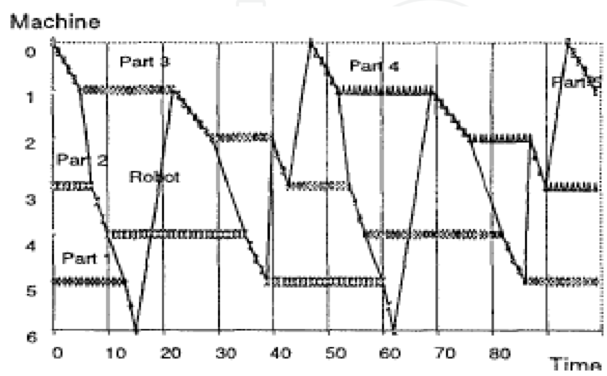


Figure 3. The Gantt chart for cyclic robotic flowshop (from Kats and Levner (1998))

3.2 Cyclic Robotic Jobshop

The cyclic robotic jobshop differs from cyclic robotic flowshop only in that each of n products in MPS has its own route as depicted in Fig. 4.

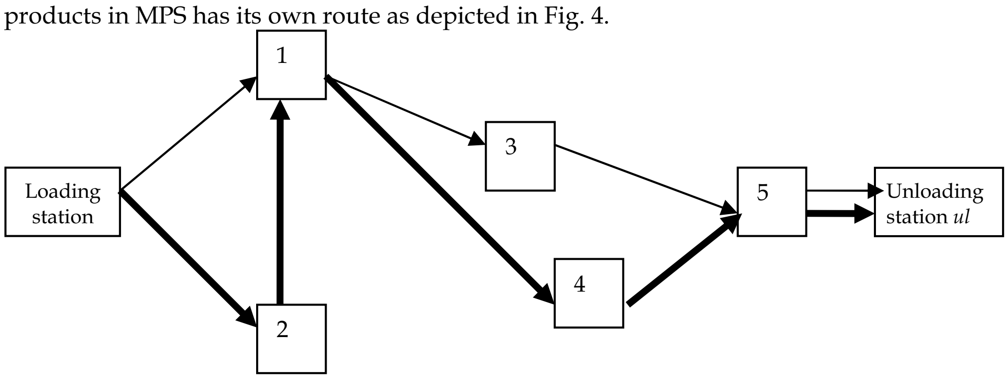


Fig. 4. An example of a simple technological network with two linear product routes and five processing machines, depicted by the squares, where ⟶ denotes the route for product *a*, and ➡ denotes the route for product *b* (from Kats et al. (2007))

The corresponding graphs depicting the sequence of technological operations and robot moves in a jobshop frame are presented in Fig. 5 and 6 .
The corresponding Gantt chart depicting coordinated movement of parts and robots in time is in Fig. 7, where stations 1 to 5 stand for the processing machines and stations 0 and 6 are, correspondingly, the loading and unloading ones. In what follows, we refer to the machines and loading/unloading stations simply as the *stations*.

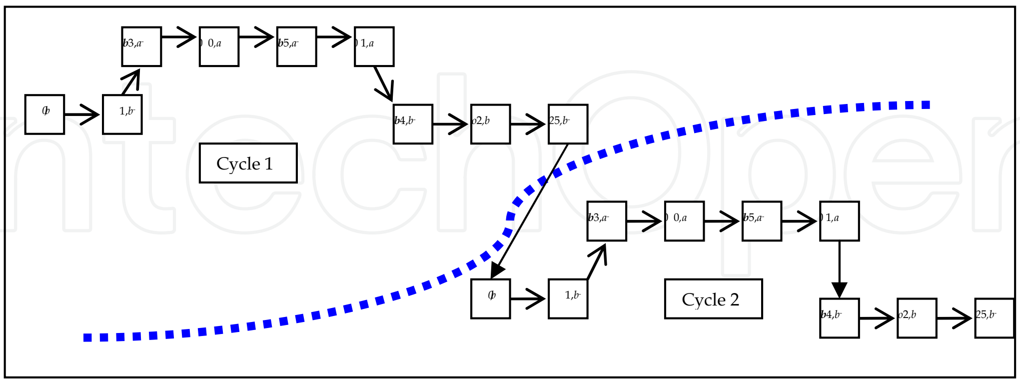


Figure 5. The sequence of robot operations in two consecutive cycles (from Kats et al. (2007))

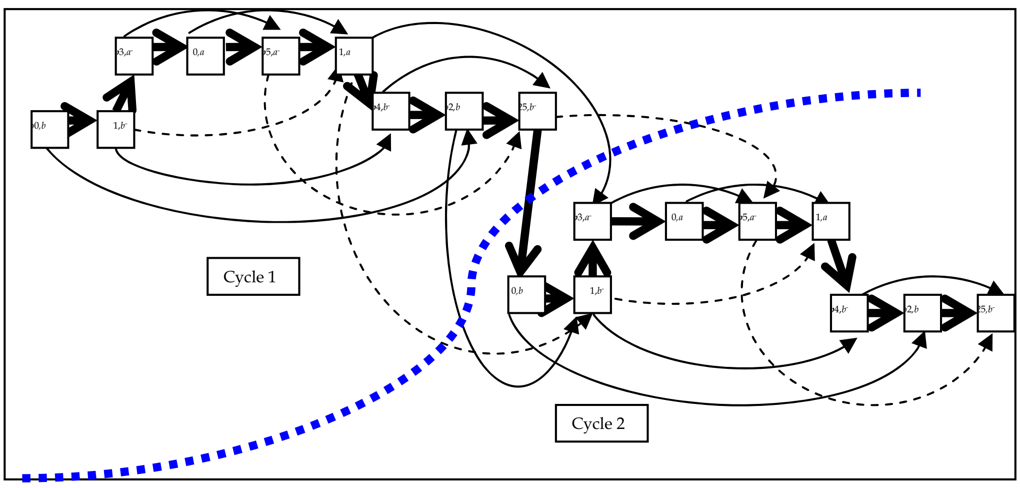


Figure 6. Graph depicting the sequence of processing operations and robot moves for two successive cycles (Kats et al. (2007)). The variables are presented as nodes and the constraints as arcs, where denotes the robot operation sequence, the processing time window constraints, setup time constraints, and the cut-off line between two cycles

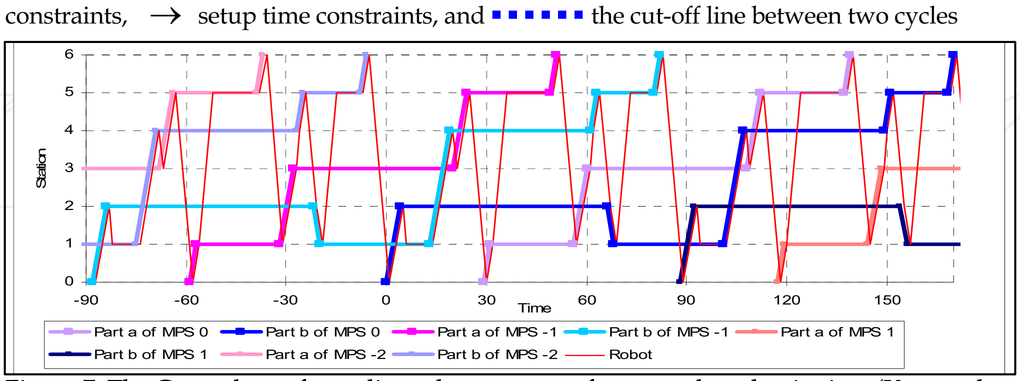


Figure 7. The Gantt chart of coordinated movement of parts and a robot in time (Kats et al. (2007))

3.3 Cyclic Robotic PERT Shop

This major class of cyclic scheduling problems which we will focus on in this sub-section, has several other names in the literature, for example, 'the basic cyclic scheduling problem', 'the multiprocessor cyclic scheduling problem', 'the general cyclic machine scheduling problem'. We will call this class the cyclic *PERT shop* due to its evident closeness to project scheduling, or PERT/CPM problems: when precedence relations between operations are given, and there is a sufficient number of machines, the parallel machine scheduling problem becomes the well-known PERT-time problem.

We define the cyclic *PERT shop* as follows: A set of n products in an MPS is given and the technological process for each product is described by its own PERT graph. A product may be considered as assembly consisting of several parts. There are three types of technological operations: a) operations which can be done in parallel on several machines, i.e. the parts consisting the assembly are processed separately; b) assembling operations; c) disassembling operations. There are infinitely many replicas of the MPS and a new MPS's replica is introduced in each cycle. In the cyclic *robotic* PERT shop, one or several robots are introduced for performing the transportation and feeding operations. The objective is to find the cyclic schedule and the robot route providing the maximum productivity, that is, the minimum cycle time.

Classes of scheduling problems	Subclasses of cyclic scheduling problems	Representative references
Cyclic Flowshop Models	*Models with negligible setups and no-robot*	Cuninghame-Greene (1960, 1962), Timkovsky (1977), Karabati and Kouvelis (1996), Lee and Posner (1997)
	Robotic models	Suprunenko et al. (1962), Tanaev (1964), Livshits et al. (1974), Phillips and Unger (1976), Kats and Mikhailetskii (1980), Kats (1982), Kats and Levner (1997a, 1997b), Crama et al. (2000), Dawande et al. (2005, 2007).
Cyclic Jobshop Models	*Models with negligible setups and no-robot*	Roundy (1992), Hanen and Munier (1995), Hall et al. (2002)
	Robotic models	Kampmeyer (2006), Kats et al. (2007)
PERT-shop Models	*Models with setups negligible, no-robot*	Romanovskii (1967), Chretienne (1991), Hanen and Munier (1995)
	Robotic models	Lei (1993), Chen et al. (1998), Levner and Kats (1998), Alcaide et al. (2007), Kats et al. (2007)

Remark. For completeness, we might mention three more groups of robotic (non-cyclic) scheduling problems which might be looked at as "atomic elements" of the cyclic problems: Robotic Non-cyclic Flowshop (Kise (1991), Levner et al. (1995a,1995b), Kogan and Levner 1998), Robotic Non-cyclic Jobshop (Hurink and Knust (2002)), and Robotic Non-cyclic PERT-shop (Levner et al. (1995c)). However, these problems lie out of the scope of the present survey.

Table 1. Classification of major cyclic scheduling problems

The cyclic robotic PERT shop problems differs from the cyclic robotic jobshop in two main aspects: a) the operations are *partially ordered*, in contrast to the jobshop where operations are linearly ordered; b) there are sufficiently many processing machines, due to which the sequencing of operations on machines vanishes. This type of problems is overviewed in more detail in surveys by Hall (1999) and Crama et al. (2000).
We conclude this section by the classification scheme for cyclic problems and the representative references (see Table 1).

4. The Cyclic Robotic FMS-shop

4.1. An Informal Description of the Cyclic Robotic FMS Shop

The cyclic robotic *FMS-shop* can be looked at as an extension of the cyclic robotic jobshop in which there given PERT-type (not-only-chain) precedence relations between assembly/disassembly operations for each product. In other view, the robotic FMS-shop can be looked at as a generalized cyclic robotic PERT-shop in which a *finite set* of machines performing the operations are given**.** In what follows, we assume that K PERT projects representing the technological processes for K products in an MPS are given and to be repeated infinitely many times on m machines.
Example. (Levner et al. (2007)). MPS consists of two products MPS ={a, b} with sequence of processing operations for products a and b given in the form of PERT graphs as shown in Fig. 8.

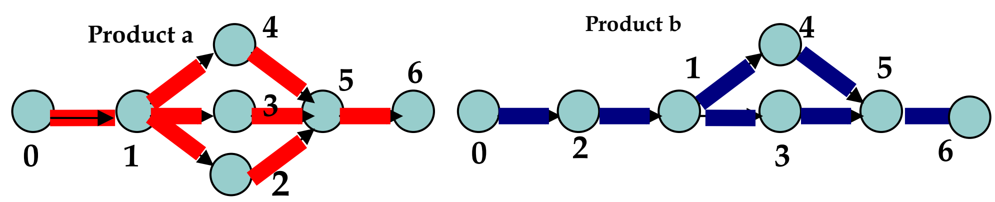


Figure 8. Two fragments of a technological network in which partially ordered (PERT-type) networks are given for two individual products in an FMS-shop

There are five processing machines and loading and unloading stations (stations 0 and 6 correspondingly). Infinite number of MPS replicas are waiting for processing and arrive periodically in process as shown in Fig. 9.

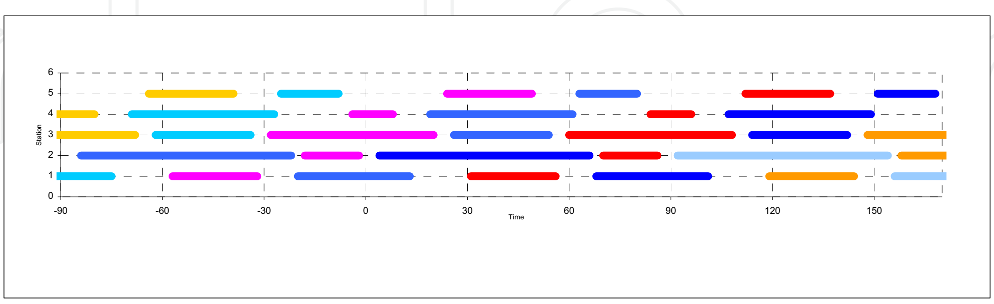


Figure 9. The Gantt chart of several MPS replicas arriving in the technological process through equal time intervals

We give the problem description basing on the model developed in Kats et al. (2007). The product (part type) processing time at any machine is not fixed, but defined by a pair of minimum and maximum time limits, called the *time window* constraints. The movements of parts between the machines and loading/unloading stations are performed by a robot, which travels in a non-negligible time. To move a part, the robot first travels to the station where the part is located, wait if the part is still in process, unload the part and then travels to the next station specified by a given sequence of material handling operations for the robot. The robot is supplied by multiple grippers in order to transport *several* parts *simultaneously* to an assembling machine or from an disassembling machine. There is no buffer available between the machines and each machine can process only one product at time. If different types of products are processed at the same machine, then a non-negligible setup time between the processing of these products may be required. The general problem is to determine the product sequence at each machine, the robot route and the exact processing time of each product at each machine so that the cycle time is minimized while the time windows, the setup times, and the robot traveling time constraints are satisfied.

Scheduling of the material handling operations of robots to minimize the cycle time, even with a single part per MPS and a single one-gripper robot, has been known to be NP-hard in strong sense (Livshits et al. (1974); Lei and Wang (1989)).

In this chapter, we are interested in a special case of the cyclic scheduling problem encountered in such a processing network. In particular, we solve the multiple-product problem of minimizing the cycle time for a processing network with a single multi-gripper robot, a *fixed* and known in advance sequence of material handling operations for the robot to be performed in each cycle and the known product sequence at each machine. Throughout the remaining analysis of this chapter, we shall denote this problem as **Q**.

Problem **Q** is a further extension of the scheduling problem **P** introduced and solved in Kats et al. (2007). The problem **P** is the jobshop scheduling problem where technological operations for each product are linked by simple chain-like precedence relations (see Fig. 5 above). Like in **P**, in problem **Q** the sequence of robot moves is assumed to be *fixed and known*. With this special case, the sequencing issue for the robot moves vanishes, and the problem reduces to finding the exact processing times from the given intervals. This case has been shown to be polynomial solvable by several researchers independently via different approaches. Representative work on this can be found in the work by Livshits et al. (1974), Matsuo et al. (1991), Lei (1993), Ioachim and Soumis (1995), Chen *et al.* (1998), Van de Klundert (1996), Levner et al. (1996, 1997), Levner and Kats (1998), Crama et al. (2000), Lee (2000), Lei and Liu (2001), Alcaide at al. (2007), Kats et al. (2007).

In this section, we analyze the properties of **Q** and show that it can be solved by the polynomial algorithm, originating from the *parametric critical path method* by Levner and Kats (1998) for the single-product version of the problem. Our main observation is that the technological processes for products presented by PERT-type graphs (see Fig. 8) can be treated by the same mathematical tools as more primitive processes presented by linear chains considered in Kats et al. (2007).

4.2. A formal analysis of problem Q

Each given instance of **Q** has a fixed sequence of material handling operations σ, and an associated MPS with K products and PERT-type precedence relations. The set of processing operations of a product in the MPS is not in the form of a simple chain like in problem **P,** but rather linked into a *technological graph,* containing assembling and disassembling operations. Let *G* denote the associated integrated *technological network* which integrates *K* technological graphs of all products in the MPS with the given sequence of processing operations on machines. In network *G*, each *node* specifies a machine or the loading station 0/unloading station *ul*, each *arc* specifies a particular precedence relationship between two consecutive processing operations of a product, and each *technological graph* to be performed for each product corresponds to a *subgraph* in network *G*.

Now, let Ω be the set of distinct stations/nodes in a given technological network *G*, *j* be the index to enumerate stations, $j \in \Omega,$ and *k* be the index for product, $1 \le k \le K.$ Each product *k* requires a total of n_k partially ordered processing operations with each operation taking place at a respective workstation. In each material handling operation the robot removes a product (or a "semi-product") from a station. Therefore, $n = K + \sum_{k=1,2,...,K} n_k$ is the total number of all operations to be performed by the robot in a cycle, including a total of *K* operations at station 0 (i.e., one for each product in the MPS to be introduced into the process in a cycle). The processing time for product *k* at station *j*, $p_{j,k}$, is a deterministic decision variable that must be confined within a given interval $[a_{j,k}, b_{j,k}]$, for $1 \le k \le K$, $j=1,2,\ldots,n_k$, and $j \ne 0,$ where parameters $a_{j,k}$ and $b_{j,k}$ are the given constants and define the time window constraints on the part processing time at workstation *j*. That is, after arriving at workstation *j*, a part of type *k* must immediately start processing and be processed there for a time interval no less than $a_{j,k}$ and no more than $b_{j,k}$. In the practices of assembling shops, the violating of the time window constraints, $a_{j,k} \le p_{j,k} \le b_{j,k}$, may deteriorate the product quality and cause a defect product.

For any given instance of **Q** sequence σ, σ = <([i], *r*[*i*], *f*(*i*)), *i*=1,2, …,*n*> specifies a total of *n* (material handling) operations to be performed by the robot in each cycle. The *i*th operation in σ, ([i], r[*i*], *f*(*i*)) where $1 \le i \le n,\ [i] \in \Omega \setminus \{ul\},\ \ r[i] \in \{1,2,\ldots,K\},$ $f(i) \in \{keep, load\}$ consists of the following sequential motions:

- *Unload product* $\boldsymbol{r}[\boldsymbol{i}]$ *from station* [*i*];
- *If f*(*i*) = *load, then transport product* $\boldsymbol{r}[\boldsymbol{i}]$ *to the next station on its technological route, s*[*i*], $s[i] \in \Omega,$ *and load product* $\boldsymbol{r}[\boldsymbol{i}]$ *to station s[i] which include the loading of all parts of the product kept by grippers.*
- *If f(i) = keep, then keep the unloaded product in gripper.*
- *Travel to station [i+1], where* $[i+1] \in \Omega \setminus \{ul\},$ *and wait if necessary. When i=n,* [*n*+1] = *0.*

In each cycle, the given sequence of operations, *σ*, is performed exactly once, so that exactly one MPS is introduced into the process and exactly one MPS is completed and sent to station *ul*. In this infinite cyclic process, parts being moved and processed within a cycle could belong to different MPS's replicas introduced in different cycles and full processing time (life cycle) of one MPS could be much longer than cycle time *T*.

Network *G* introduces two types of precedence relationships. The first type of relationships ensures the *processing time window constraints*, and the second type refers to the *setup time* *constraints on sharing stations*. The latter incorporates the corresponding setup times into the model when two or more part types are to be processed at the same station.

Let time moment 0 be a reference time point in the infinite cyclic process and assume, without loss of generality, that the current cycle starts at time 0. Let MPS(q) be the qth replica of the MPS such that its first operation starts at time $q \cdot T$, where q= 0, ±1, ±2,…

Let $\boldsymbol{z}_{[i],r[i]}$ be the moment when part $r[i] \in MPS(0)$ is removed from station [i]. Then

$$\boldsymbol{t}_{[i],r[i]} \equiv \boldsymbol{z}_{[i],r[i]} (\mathrm{mod}\ \boldsymbol{T}) = \boldsymbol{z}_{[i],r[i]} - \boldsymbol{h}_{[i],r[i]} \cdot \boldsymbol{T} \tag{2}$$

is the moment within interval [0, T) when part $r[i] \in MPS(-h_{[i],r[I]})$ is removed from station [i]

To make a formal definition for problem **Q**, let's introduce the following additional notation:

$L_{[i]}$	The part loading time at station [i], $[i] \in \Omega \setminus \{ul\}$;
$U_{[i]}$	The part unloading time at station [i], $[i] \in \Omega \setminus \{0\}$;
$d_{[i],[i']}$	The robot traveling time from stations [i] to [i'];
$g_{[i]}^{a,b}$	The pre-specified setup time at shared station [i] between the processing of part a and the processing of part b, where a, b $\in$ {1,…, K};
Φ	The given set of paired technological operations;
$Y_{[i]}$	Sequence (σ)-dependent binary constants: $Y_{[i]}$ =1 if (s[i], r[i]) and ([i], r[i]) are in the same cycle, and $Y_{[i]}$ = 0 otherwise (see Kats et al. (2007)).

Problem **Q** can be described in the same terms as **P** in Kats et al. (2007):

$$\mathbf{Q}\text{:}\quad Minimize \quad T$$

subject to

The multigripper robot traveling time constraints

For all i, 1 ≤ i ≤ n, such that f(i) = load

$$t_{[i],r[i]} + U_{[i]} + d_{[i],s[i]} + L_{s[i]} + d_{s[i],\,[i+1]} \le t_{[i+1],r[i+1]} \tag{3a}$$

For all i, 1≤i≤n, such that f(i) = keep

$$t_{[i],r[i]} + U_{[i]} + d_{[i],\,[i+1]} \le t_{[i+1],r[i+1]}, \tag{3b}$$

where $t_{[n+1],r[n+1]} = t_{[1],r[1]} + T$.

The processing time window constraints

For all i, 1 ≤ i ≤ n, such that f(i) = load

if $Y_{[i]}$ = 0

$$\begin{aligned} t_{s[i],r[i]} - t_{[i],r[i]} &\ge U_{[i]} + d_{[i],s[i]} + L_{s[i]} + a_{s[i],r[i]}, \\ t_{s[i],r[i]} - t_{[i],r[i]} &\le U_{[i]} + d_{[i],s[i]} + L_{s[i]} + b_{s[i],r[i]}. \end{aligned} \tag{4a}$$

if $Y_{[i]} = 1$

$$T + t_{s[i],r[i]} - t_{[i],r[i]} \geq U_{[i]} + d_{[i],s[i]} + L_{s[i]} + a_{s[i],r[i]}, \tag{4b}$$

$$T + t_{s[i],r[i]} - t_{[i],r[i]} \leq U_{[i]} + d_{[i],s[i]} + L_{s[i]} + b_{s[i],r[i]}.$$

The setup time constraints on sharing stations

For all $i' < i,\ 1 < i', i \leq n,\ [i'] = [i],\ and\ ([i], r[i'], r[i]) \in \Phi$

$$t_{[i],r[i]} - t_{[i'],r[i']} \geq g_{[i]}^{r[i'],r[i]}, \tag{5a}$$

$$(T - t_{[i],r[i]}) + t_{[i'],r[i']} \geq g_{[i']}^{r[i],r[i']} \tag{5b}$$

The non-negativity condition

All variables T, $\boldsymbol{t}_{[i],r[i]}, 1 \leq \boldsymbol{i} \leq \boldsymbol{n},$ *are non-negative.*

Constraints (3) ensure the robot to have enough time to operate and to travel between the starting times of two consecutive operations in sequence σ. Constraints (4) enforce the part processing time at a station to be in given windows. Constraints (5) ensure the required setup time at the shared stations to be guaranteed.

The processing time window constraints (4a)-(4b) ensure $a_{j,k} \leq p_{j,,k} \leq b_{j,k}$, where $p_{s[i],r[i]}$ stands for the actual processing time of part $r[i]$ in station $s[i]$ and is determined by the optimal solution to **Q**. The "no-wait" requirement means that a part, once introduced into the process, must be in the status of either being processed at a station or being transported by a material handling robot.

One can easily observe that the relationships (3) - (6) are of the same form as those in the model **P**, and thus an extension of simple chains to the PERT-graphs for each product does not change the inherent mathematical structure of the model suggested by Kats et al. (2007), and the complexity of the algorithm proposed for solving **P**.

4.3. A Polynomial Algorithm for Scheduling the FMS Shop

In this section, we develop results contained in Alcaide et al. (2007) and Kats et al. (2007). Our considerations are based on the strongly polynomial algorithm for solving problem **P** suggested by Kats et al. (2007). However, for reader's convenience, we present the algorithm for problem **Q** in a simplified form, following the scheme and notation developed in Levner and Kats (1998). To do so, let's start with the following result.

PROPOSITION 1. *Problem* **Q** *is a parametric critical path (PCP) problem defined upon a directed network* $G_P = (V, A)$ *with parameter-dependent arc lengths.*

The proof is along the same line as for problem **P** in Kats et al. (2007).

The algorithm below for solving **Q** is called the *Parametric Critical Path* (*PCP*) algorithm. As that for problem **P,** it consists of three steps (Table 2 below). The first step assigns initial labels to nodes in a given network G_P, the second step corrects the labels, and the third step, based on the labels obtained, finds the set Λ of all feasible cycle times or discovers if this set is empty.

PARAMETRIC CRITICAL PATH (PCP) ALGORITHM

***Step* 1** . // *Initialization.*

Enumerate all the nodes of $V \cup \{f\}$ in an arbitrary order.

Assign labels $p^0(s) = p_1^0 = 0$, $p_j^0 = w(s \to j)$ if $j \neq s$;

$Pred(s) = \varnothing$, and $p^0(v) = -\infty$ to all other nodes v of $V \cup f$.

***Step* 2.** // *Label correction.*

For $i := 1$ *to* n -1 **do**

For each arc $e = (t(e), h(e)) \in A$ compute $max\{p^{i-1}(h(e)), p^{i-1}(t(e)) + w(e)\}$.

Calculate

$$p^i(h(e)) := \max_{u \in Pred(h(e))} \left\{ \max\left\{ p^{i-1}(h(e)), p^{i-1}(u) + w(u \to h(e)) \right\} \right\}. \quad (6)$$

//Notice that for $u \in Pred(h(e))$, $u \to h(e)$ denotes the existing arc from u to $h(e)$).

***Step* 3.** //*Finding all feasible T values or displaying 'no solution'.*

For each arc $e = (t(e), h(e)) \in A$ solve the following system of functional inequalities

$$p^{n-1}(t(e)) + w(e) \leq p^{n-1}(h(e)), \quad (7)$$

with respect to T.

Let Λ be the set of values of T satisfying (7) for *all* $e \in A$.

If $\Lambda \neq \varnothing$, then return Λ and stop. Otherwise return '*no solution*'.

At termination, the algorithm either produces the set Λ of all feasible T, or it reveals that $\Lambda = \varnothing$. In the case $\Lambda \neq \varnothing$, then $\Lambda = [T_{min}, T_{max}]$ is an interval.

Let Λ be the set of values of feasible T satisfying (6)-(7) for *all* $e \in A$.

If $\Lambda \neq \varnothing$, then return Λ and stop. Otherwise return 'No solution' and stop.

Table 2. The Parametric Critical Path (PCP) Algorithm

The algorithm terminates with a non-empty set, Λ, if there exists at least one feasible cycle time on G_P. By the definition of Λ, the optimal cycle time T^* is the minimal value in Λ. Once the value of T^* is known, the optimal values of all the t-variables in model **Q** (i.e., the optimal starting times of robot operations in sequence σ) are known as well, and the optimal processing time, $\boldsymbol{p}_{s[i],r[i]}$, where $\boldsymbol{a}_{s[i],r[i]} \leq \boldsymbol{p}_{s[i],r[i]} \leq \boldsymbol{b}_{s[i],r[i]}$, for each part

$r[i] = k \in \{1,2,...,K\}$ in each respective station $s[i]$ along its route, $1 \le i \le n$, can be found.

For each arc $e \in A(G_p)$, let $t(e)$, $h(e)$, and $w(e)$ denote the tail, the head, and the length of arc e, respectively. Let j denote node $([\boldsymbol{j}], \boldsymbol{r}[\boldsymbol{j}]),\ 2 \le \boldsymbol{j} \le \boldsymbol{n}+1,\ \forall([j], r[j]) \in V$, p_j^i denote the *distance label* of node j found at the i-th iteration of the PCP algorithm, and $(k \to j)$ denote the arc from node k to j. Let $N = n+1$ be the total number of nodes of G_P (counting for all the nodes in V plus the added dummy node f), and M the total number of iterations.

It is worth noticing that labels $p^i(u)$ in (6)–(7) are not numbers but the piecewise-linear functions of T.

PROPOSITION 2. *The Parametric Critical Path algorithm finds the optimal solution to problem* **Q** *correctly. The complexity of the parametric critical path algorithm is $O(n^4)$, in the worst case.*

The proof is identical to that for problem **P** in Kats et al. (2007).

The following example illustrates how an optimal schedule is obtained by the use of the proposed PCP algorithm.

Example (Continued). The sequence σ of robot moves is fixed and given:

σ = <(0,b_0,U), (2,b_0,L), (4,a_{-1},U), (1,b_{-1},U), (4,b_{-1},L), (3,a_{-1},U), (5,a_{-1},L), (3,b_{-1},L), (0,a_0,U), (1,a_0,L), (5,a_{-1},U), (6,a_{-1},L), (3,b_{-1},U), (1,a_0,U), (3,a_0,L), (4,b_{-1},U), (5,b_{-1},L), (2,b_0,U), (1,b_0,L), (2,a_0,L), (5,b_{-1},U), (6,b_{-1},L), (4,a_0,L), (2,a_0,U)>.

Here we use a more detailed description of robot operations given in the form of triplets (*, *, *). A number in the first position determines the processing machine or loading/unloading station, numbered 0 and 6, respectively. A symbol in the second position determines the product type (a or b); a corresponding subscript determines to which MPS replica the product belongs. A symbol in the last position determines that a product is either loaded (symbol L) or unloaded (symbol U).

Then the life cycle of the MPS is completed within two consecutive cycles $\sigma| \, |\sigma$, and is shown in Fig. 6. The Gantt chart of the movements of products and the robot under the optimal schedule are presented graphically in Fig.10. The minimum cycle time $T^* = 88$.

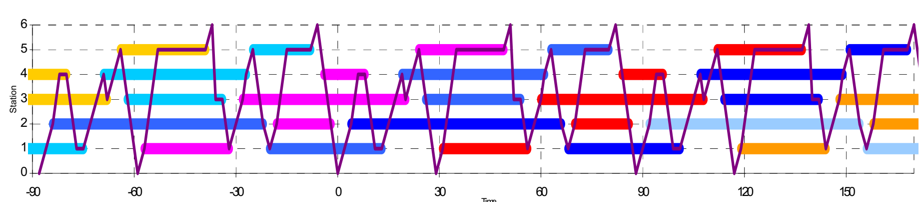


Figure 10. The Gantt chart of product processing operations and robot movements

We have studied a variation of the single multi-gripper robot cyclic scheduling problem with a fixed robot operation sequence and the time window constraints on the processing times. It generalizes the known single-robot single-product problems into the one involving a processing network, multiple products, and general precedence relations between the processing steps for different products in the form of PERT graphs. We reduced the problem to the parametric critical path problem and solved it in polynomial time by an extension to the Bellman-Ford algorithm. In particular, we simplified the description of the labeling procedure suggested by Kats et al. (2007) needed to solve the parametric version of the critical path problem in strongly polynomial time.

5. Concluding Remarks

Since Johnson's (1954) and Bellman's (1956) seminal papers, the machine scheduling theory have received considerable development and enhancement over the last fifty years. As a result, a variety of scheduling problems and optimization techniques have been developed. This chapter provides a brief survey of the evolution of basic cyclic scheduling problems and possible approaches for their solution started with a discussion of early works appeared in the 1960s. Although the cyclic scheduling problems are, in general, NP-hard, a graph approach described in the final sections of this chapter permits to reduce some special case to the parametric critical path problem in a graph and solve it in polynomial time. The proposed parametric critical path algorithm can be used to design new heuristic search algorithms for more general problems involving multiple multi-gripper robots, parallel machines/tanks at each workstation and more general scenarios of cyclic processes in the cells, like, for example, multi-degree periodic processes. These are the topics for future research.

6. Acknowledgements

This work has been partially supported by Spanish Government Research Projects DPI2001-2715-C02-02, MTM2004-07550 and MTM2006-10170, which are helped by European Funds of Regional Development. The first author gratefully acknowledges the partial support by the Spanish Ministry of Education and Science, grant SAB2005-0161.

7. References

V.S. Aizenshtat (1963). Multi-operator cyclic processes, *Doklady of the Byelorussian Academy of Sciences*, 7(4), 224-227 (Russian).

A. Agnetis and D. Pacciarelli (2000). Part sequencing in three-machine no-wait robotic cells, *Operations Research Letters,* 27(4), 185-192.

D. Alcaide, C. Chu, V. Kats, E. Levner, and G. Sierksma (2007). Cyclic multiple-robot scheduling with time-window constraints using a critical path approach, *Eur. J. of Oper. Res.*, 177, 147-162.

F.L. Baccelli, G. Cohen, J.P. Quadrat, G.J. Olsder (1992). *Synchronization and Linearity,* Wiley.

R. Bellman (1956). Mathematical aspects of scheduling theory. *Journal of Society of Industrial and Applied Mathematics* 4, 168–205.

J. Blazewicz, K. Ecker, E.Pesch, G. Schmidt, J. Weglarz, et al. (2007). *Handbook of Scheduling Theory*, Springer.

P. Brucker (2007). *Scheduling Algorithms*, Berlin, Springer, 5th edition..

P. Brucker, C. Dhaenens-Flipo, S. Knust, S.A. Kravchenko, and F. Werner (2002). Complexity results for parallel machine problems with a single server, *Journal of Scheduling* 5, 429-457.

A. Che and C. Chu (2005a). A polynomial algorithm for no-wait cyclic hoist scheduling in an extended electroplating line, *Operations Research Letters,* 33, 274-284.

A. Che and C. Chu (2005b). Multidegree cyclic scheduling of two robots in a no-wait flowshop, *IEEE Transactions on Automation Science and Engineering*, 2(2), 173–183.

A. Che, Chu C., and Chu F.(2002). Multicyclic hoist scheduling with constant processing times, *IEEE Transactions on Robotics and Automation*, 18/1, 69–80.

A. Che, C. Chu, and E. Levner (2003). A polynomial algorithm for 2-degree cyclic robotscheduling, *European Journal of Operational research*, 145(1), 31-44.

H. Chen, C. Chu, J.M. Proth (1998). Cyclic scheduling of a hoist with time window constraints, *IEEE Transactions on Robotics and Automation, 14,* 144-152, 1998.

C. Chu (2006). A faster polynomial algorithm for 2-cyclic robotic scheduling, *Journal of Scheduling*, October, 9 (5), 453-468.

G. Cohen, D. Dubois, J.P. Quadrat, and M.Viot (1985). A linear system theoretic view of discrete event processes and its use for performance evaluation in manufacturing, *IEEE Transactions on Automatic Control*, 30(1), 210-220, March.

Y. Crama, V. Kats, J. Van de Klundert, E. Levner (2000). Cyclic scheduling in robotic flowshops, *Annals of Operations Research, 96*, 97-124.

P. Chretienne (1991). The basic cyclic scheduling problem with deadline, *Discrete Applied Mathematics, 30,* 109-123.

R.A. Cuninghame-Greene (1960). Process synchronization in a steelworks, - a problem of feasibility, *Proceedings of the 2nd International Conference on Operational Research.*

R.A. Cuninghame-Greene (1962). Describing industrial processes with interference and approximation their steady-state behaviour, *Operational Research Quarterly,* 13(1), 95-100. March.

R.A. Cuninghame-Greene (1979). *Minimax Algebra*, Springer-Verlag.

A Dasdan, S.S. Irani, R.K. Gupta (1999). Efficient algorithms for optimum cycle mean and optimum cost to time ratio problems, *Proceedings of the 36th ACM/IEEE conference on Design automation*, *New Orleans, Louisiana, United States*, 37 - 42.

W. Dauscha, H.D. Modrow, A. Neumann (1985). On cyclic sequence types for constructing cyclic schedule, *Zeitschrift fur Operations Research,* 29, 1-30.

M. Dawande, Geismer H.N, Sethi S.P., Sriskandarajah C. (2005). Sequencing and scheduling in robotic cells: Recent developments, *Journal of Scheduling*, 8(5), 387-426.

M.N. Dawande, H.N. Geismer, S. P.Sethi, and C. Sriskandarajah (2007). *Througput Optimization in Robotic Cells*, Springer.

Yu.I. Degtyarev and V.G. Timkovsky (1976). On a model of optimal planning systems of flow type. *Automation Control Systems*, 1, 69-77 (in Russian).

M. Gondran and M. Minoix (1985). *Graphes et algorithmes*, Eyrolles, Paris.

S.C. Graves, H.C. Meal, D. Stefek, A.H. Zeghmi (1983). Scheduling of re-entrant flow shops, *Journal of Operations Management*, 3, 197-207.

J. N.D. Gupta and E. F. Stafford Jr. (2006). Flowshop scheduling research after five decades, *European Journal of Operational Research*, 169, 3, 699-711.

N.G Hall (1999). Operations research techniques for robotic system planning, design, control and analysis, Chapter 30 in *Handbook of Industrial Robotics*, vol. II, ed. S.Y. Nof, John Wiley, 543–577.

N.G. Hall., T.-E. Lee and M.E. Posner (2002). The complexity of cyclic shop scheduling problems, *Journal of Scheduling*, 5 (2002) 307–327.

C. Hanen (1994). Study of a NP-hard cyclic scheduling problem: The recurrent job-shop, *European Journal of Operational Research*, 72, 82-101.

C. Hanen and A. Munier (1995). Cyclic scheduling on parallel processors: An Overview, in P. Chretienne, E.G. Coffman, Jr., J.K. Lenstra, and Z.Liu (eds.), *Scheduling Theory and its Applications*, Wiley, 194-226.

H. Hillion and J.M. Proth (1989). Performance evaluation of a job-shop system using timed event graphs, *IEEE Transactions on Automatic Control,* AC-34, 3-9.

R.A Howard (1960). *Dynamic Programming and Markov Processes*, Wiley, N.Y.

J. Hurink and S.Knust (2002). A tabu search algorithm for scheduling a single robot in a job-shop environment, *Discrete Applied Mathematics*, 119, 181-203.

I. Ioachim and F. Soumis (1995). Schedule efficiency in a robotic production cell, *The International Journal of Flexible Manufacturing Systems*, 7, 5-26.

S.M. Johnson (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly* 1, 61–68.

T. Kampmeyer (2006). *Cyclic Scheduling Problems*. PhD theses, University of Osnabruck, Germany.

S. Karabati and P. Kouvelis (1996). The interface of buffer design and cyclic scheduling decisions in deterministic flow lines*, Annals of Operations Research*, 50, 295-317.

R.M. Karp (1978). A Characterization of the Minimum Cycle Mean in a Digraph, *Discrete Math.*, 23: 309-311.

V. Kats (1982). An exact optimal cyclic scheduling algorithm for multioperator service of a production line, *Automation and Remote Control*, 43(4), part 2, 538–543,1982.

V. Kats, L.Lei, E. Levner (2007). Minimizing the cycle time of multiple-product processing networks with a fixed operation sequence and time-window constraints, *European Journal of Operational Research*, in press.

V. Kats and E. Levner (1997a). A strongly polynomial algorithm for no-wait cyclic robotic flowshop scheduling, *Operations Research Letters*, 21, 171-179.

V. Kats and Levner E. (1997b). Minimizing the number of robots to meet a given cyclic schedule, *Annals of Operations Research*, 69, 209–226.

V. Kats, E. Levner (1998). Cyclic scheduling of operations for a part type in an FMS handled by a single robot: a parametric critical-path approach, *The International Journal of FMS*, 10, 129-138.

V. Kats and E. Levner (2002). Cyclic scheduling on a robotic production line, *Journal of Scheduling,* 5, 23-41.

V. Kats and E. Levner (2003). Polynomial algorithms for periodic scheduling of tasks on parallel processors, in: L.T. Yang and M. Paprzycki (eds). *Practical Applications of Parallel Computing: Advances in Computation Theory and Practice,* vol. 12, Nova Science Publishers, Canada, 363-370.

V. Kats, E. Levner, and L. Meyzin (1999). Multiple-part cyclic hoist scheduling using a sieve method, *IEEE Transactions on Robotics and Automation*, 15(4), 704–713.

V.B. Kats and Z.N. Mikhailetskii (1980), Exact solution of a cyclic scheduling problem, *Automation and Remote Control* 4, 187-190.

H. Kise (1991). On an automated two-machine flowshop scheduling problem with infinite buffer, *Journal of the Operations Research Society of Japan*, 34(3) 354-361.

J.J. Van de Klundert (1996). *Scheduling Problems in Automated Manufacturing,* Dissertation 96-35, Faculty of Economics and Business Administration, University of Limburg, Maastricht, the Netherlands.

K. Kogan, and E. Levner (1998). A polynomial algorithm for scheduling small-scale manufacturing cells served by multiple robots, *Computers and Operations Research* 25(1), 53-62.

W. Kubiak (2005). Solution of the Liu-Layland problem via bottleneck just-in-time sequencing, *Journal of Scheduling*, 8(4), 295 – 302.

T.E. Lee (2000). Stable earliest starting schedules for cyclic hob shops: A linear system approach, *The International Journal of Flexible Manufacturing Systems*, 12, 59-80.

T.E. Lee, M.E. Posner (1997). Performance measures and schedules in periodic job shops, *Operations Research,* 45(1), 72-91.

L. Lei (1993). Determining the optimal starting times in a cyclic schedule with a given route, *Computers and Operations Research*, 20, 807-816.

L. Lei and Q. Liu (2001). Optimal cyclic scheduling of a robotic processing line with two-product and time-window constraints, *INFOR,* 39 (2).

L. Lei and T.J. Wang (1989). *A proof: The cyclic scheduling problem is NP-complete*, Working Paper no. 89-0016, Rutgers University, April.

V. Lev and I. Adiri (1984). V-shop scheduling, *European Journal of Operational Research,* 18, 561-56.

E. Levner (1991). Mathematical theory of project management and scheduling, in: *Encyclopaedia of Mathematics*, M. Hazewinkel (ed.), Kluwer Academic Publishers, Dordrecht, vol.7, 320-322.

E. Levner (1992). Scheduling theory, in: *Encyclopaedia of Mathematics*, M. Hazewinkel (ed.), Kluwer Academic Publishers, Dordrecht, vol.8, 210-212.

E. Levner, V Kats (1998). A parametrical critical path problem and an application for cyclic scheduling, *Discrete Applied Mathematics*, 87, 149-158.

E. Levner, V. Kats, and D. Alcaide (2007). Cyclic scheduling of assembling processes in robotic cells, *Abstracts of the talks at the XXth European Chapter on Combinatorial Optimization,* ECCO-2007, Limassol, Cyprus, May 23-26.

E.V. Levner, V. Kats and V. Levit (1997). An improved algorithm for cyclic flowshop scheduling in a robotic cell, *European Journal of Operational Research*, 197, pp. 500-508.

E. Levner, V. Kats and C. Sriskandarajah (1996). A geometric algorithm for finding two-unit cyclic schedules in no-wait robotic flowshop, *Proceedings of the International Workshop in Intelligent Scheduling of Robots and FMS, WISOR-96,* Holon, Israel, HAIT Press, 101-112.

E. Levner, K. Kogan and I. Levin (1995a). Scheduling a two-machine robotic cell: A solvable case, *Annals of Operations Research*, 57, pp.217-232.

E. Levner, K. Kogan, O. Maimon (1995b). Flowshop scheduling of robotic cells with job-dependent transportation and setup effects, *Journal of Oper. Res. Society* 45, 1447-1455.

E. Levner, I. Levin and L. Meyzin (1995c). A tandem expert system for batch scheduling in a CIM system based on Group Technology concepts, *Proceedings of 1995 INRIA/IEEE Symposium on Emerging Technologies and Factory Automation* ITFA'95, v.1, 667-674, IEEE Press, Paris, France.

J.Y.-T. Leung (ed.) (2004). *Handbook of Scheduling: Algorithms, Models, and Performance Analysis,* Chapman & Hall/CRC, Boca Raton.

E.M. Livshits, Z.N. Mikhailetsky, and E.V. Chervyakov (1974). A scheduling problem in an automated flow line with an automated operator, *Computational Mathematics and Computerized Systems,* Charkov, USSR, 5, 151-155 (Russian).

M.A. Manier and C. Bloch (2003). A classification for hoist scheduling problems, International *Journal of Flexible Manufacturing Systems,* 15, 37-55.

H. Matsuo, J.S. Shang, R.S. Sullivan (1991). A crane scheduling problem in a computer-integrated manufacturing environment, *Management Science, 17,* 587-606.

S.T. McCormick, M.L. Pinedo, S. Shenker, B. Wolf (1989). Sequencing in an assembly line with blocking to minimize cycle time, *Operations Research, 37,* 925-935.

M. Middendorf and V. Timkovsky, (2002). On scheduling cycle shops: classification, complexity and approximation, *Journal of Scheduling*, 5(2), 135-169.

L.W. Phillips and P.S. Unger (1976). Mathematical programming solution of a hoist scheduling progrm, *AIIE Transactions*, 8(2), 219-225.

M. Pinedo (2001). *Scheduling: Theory, Algorithms and Systems,* Prentice Hal, N.J.

C. Ramchandani (1973). Analysis of asynchronous systems by timed Petri nets, PhD Thesis, MIT Technological Report 120, MIT.

R. Reiter (1968). Scheduling parallel computations, *Journal of ACM, 15(4),* 590-599.

I.V. Romanovskii (1967). Optimization of stationary control of a discrete deterministic process, *Kybernetika (Cybernetics)* , v.3, no.2, pp. 66-78.

R.O. Roundy (1992). Cyclic schedules for job-shops with identical jobs, *Mathematics of Operations Research*, 17, November, 842-865.

J.-W.Seo and T.-E.Lee (2002). Steady-state analysis and scheduling of cycle job shops with overtaking, *The International Journal of Flexible Manufacturing Systems,* 14, 291-318.

P. Serafini, W. Ukovich (1989). A mathematical model for periodic scheduling problems, *SIAM Journal on Discrete Mathematics*, *2*, 550-581.

S.P. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz and W. Kubiak (1992). Sequencing of parts and robot moves in a robotic cell, *The International Journal of FMS*, 4, 331-358.

R.R.K. Sharma and S.S. Paradkar (1995). Modelling a railway freight transport system, *Asia-Pacific Journal of Operational Research*, 12, 17-36.

A. Shtub, A., J. Bard and S. Globerson (1994). *Project Management,* Prentice Hall.

D.A. Suprunenko, V.S. Aizenshtat and A.S. Metel'sky (1962). A multistage technological process, *Doklady Academy Nauk BSSR*, 6(9) 541-522 (in Russian).

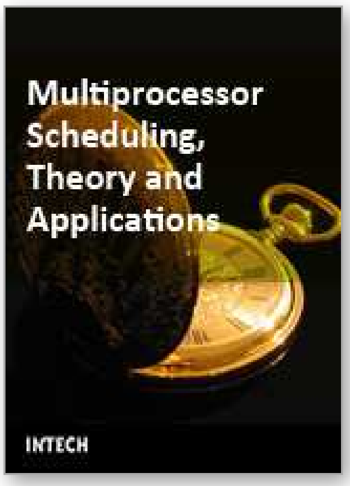
V.S. Tanaev (1964). A scheduling problem for a flowshop line with a single operator, *Engineering Physical Journal* 7(3) 111-114 (in Russian).

V.S. Tanaev, V.S.Gordon, and Ya.M. Shafransky (1994a). *Scheduling Theory*. *Single-Stage Systems,* Kluwer, Dordrecht.

V.S. Tanaev, Y.N. Sotskov and V.A. Strusevich (1994b). *Scheduling Theory*. *Multi-Stage Systems,* Kluwer, Dordrecht.

V.G. Timkovsky (1977). On transition processes in systems of flow type. *Automation Control Systems,* 1(3), 46-49 (in Russian).

V.G. Timkovsky (2004). Cyclic shop scheduling. In J.Y.-T. Leung (ed.) *Handbook of Scheduling: Algorithms, Models, and Performance Analysis,* Chapman & Hall/CRC, 7.1-7.22.



Multiprocessor Scheduling, Theory and Applications
Edited by Eugene Levner

ISBN 978-3-902613-02-8
Hard cover, 436 pages
Publisher I-Tech Education and Publishing
Published online 01, December, 2007
Published in print edition December, 2007

A major goal of the book is to continue a good tradition - to bring together reputable researchers from different countries in order to provide a comprehensive coverage of advanced and modern topics in scheduling not yet reflected by other books. The virtual consortium of the authors has been created by using electronic exchanges; it comprises 50 authors from 18 different countries who have submitted 23 contributions to this collective product. In this sense, the volume can be added to a bookshelf with similar collective publications in scheduling, started by Coffman (1976) and successfully continued by Chretienne et al. (1995), Gutin and Punnen (2002), and Leung (2004). This volume contains four major parts that cover the following directions: the state of the art in theory and algorithms for classical and non-standard scheduling problems; new exact optimization algorithms, approximation algorithms with performance guarantees, heuristics and metaheuristics; novel models and approaches to scheduling; and, last but least, several real-life applications and case studies.