Diophantine problem, cyclic scheduling, timetabling, multi-criteria optimization

Grzegorz BOCEWICZ[*], Krzysztof BZDYRA[*], Zbigniew BANASZAK[**]

CYCLIC SCHEDULING AND DIOPHANTINE PROBLEMS

Abstract

Cyclic scheduling concerns both kinds of questions following the deductive and inductive ways of reasoning. First class of problems concentrates on rules aimed at resources assignment as to minimize a given objective function, e.g. the cycle time, the flow time of a job. In turn, the second class focuses on a system structure designing as to guarantee the assumed qualitative and/or quantitative measures of objective functions can be achieved. The third class of problems can be seen, however as integration of earlier mentioned, i.e. treating design and scheduling or design and planning simultaneously. The complexity of these problems stems from the fact that system configuration must be determined for the purpose of processes scheduling, yet scheduling must be done to devise the system configuration. In that context, the contribution provides discussion of some Diophantine problems solubility issues, taking into.

1. INTRODUCTION

The way an enterprise's production capacity is used decides about its competitiveness. In that context studies aimed at designing of decision support systems (DSS) dedicated to discrete processes scheduling, and especially cyclic scheduling are of primary importance.

A cyclic scheduling problem is a scheduling problem in which some set of activities is to be repeated an indefinite number of times, and it is desired that the sequence be repeating. Cyclic scheduling problems arise in different application domains (such as manufacturing, time-sharing of processors in embedded systems, digital signal processing, and in compilers for scheduling loop operations for parallel or pipelined architectures) as well as service domains (cowering such areas as workforce scheduling (e.g., shift scheduling, crew scheduling), timetabling (e.g., train timetabling, aircraft routing and scheduling), and reservations (e.g., reservations with or without slack, assigning classes to rooms)) [4], [5], [7], [16], [17], [18]. The scheduling problems considered belong to the class of NP-hard ones and are usually formulated in terms of decision problems, i.e. as searching for an answer whether a solution possessing of assumed features exists or not [11].

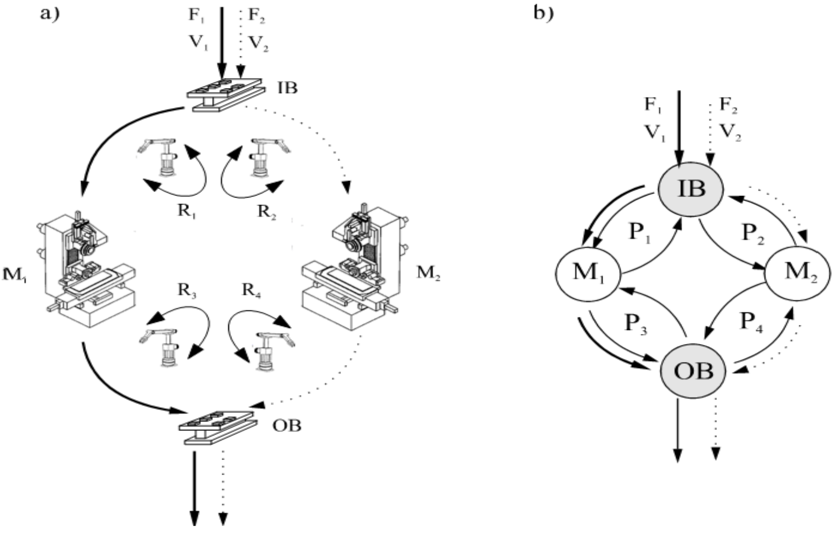
[*] Koszalin University of Technology, Dept. of Computer Science and Management, 75-354 Koszalin, Śniadeckich 2, Poland, e-mail: bocewicz@ie.tu.koszalin.pl

[**] Warsaw University of Technology, Faculty of Management, Dept. of Business Informatics, 02-524 Warszawa, Narbutta 85, e-mail: Z.Banaszak@wz.pw.edu.pl

More formally, a decision problem can be seen as a question stated in some formal system with a yes-or-no answer, depending on the values of some input parameters. The decision problems fall into two categories: decidable and non decidable problems [9], [14], [15]. In this context the primary question is to what kind of above mentioned problems belong the real-life cyclic scheduling ones.

A decision problem is called decidable or effectively solvable if there exists an algorithm which terminates after a finite amount of time and correctly decides whether or not a given number belongs to the set. A classic example of a decidable decision problem is the set of prime numbers. It is possible to effectively decide whether a given natural number is the prime one, by testing every possible nontrivial factor. A set which is not computable is called non computable or non decidable (undecidable). The relevant illustration provides the problem of deciding whether a Diophantine equation (multivariable polynomial equation) has a solution in integers.

In this paper we provide illustrative examples proving the Diophantine nature of timetabling originated cyclic scheduling models. In that context, the paper's objective is to provide the methodology employing the reverse approach while aimed at DSS designing. The rest of the paper is organized as follows: Section 2 describes the case of manufacturing processes cyclic scheduling. The concept of Diophantine problem is then recalled in Section 3. In Section 4, a case of timetabling like scheduling problem is investigated. Conclusions are presented in Section 5.



Legend:
R_i – the i-th robot; M_j – the j-th machine tool; IB, OB – the input and output buffers
P_i – the i-th cyclic work piece transportation/handling process,
F_i – the i-th production route,
→ - the cyclic process flow direction,
⇢ - the i-th production route.

Fig.1. Robotic cell: a) the cell layout, b) the model of cyclic processes

2. SYSTEMS OF CONCURRENT CYCLIC PROCESSES

Consider the flexible manufacturing cell composed of four industrial robots, two machine tools, and the input and output buffers shown in Fig. 1. Robots transporting work pieces from input buffer or machine tool to another machine tool or output buffer serve for work pieces handling – two kinds of work pieces processed along two different production routs. Production routes can be seen as cyclic repetitively executed processes supporting the work pieces flow along different production routs.

In general case the cells considered can be organized in much more complex flexible manufacturing systems (see Fig. 2). In such structures different cyclic scheduling problems can be considered. The typical problem concerns of work pieces routing, in which a single robot has to make multiple tours with different frequencies. The objective is to find a minimal makespan schedule in which the robots repeat their handlings with assumed frequencies.

In general case, a system of repetitive manufacturing processes consists of a set of processes sharing common resources while following a distributed mutual exclusion protocol (see Fig. 3). Each process P_i, (i=1,2,...,n), representing one product processing, executes periodically a sequence of the operations using resources defined by $Z_i = (R_{i1}, R_{i2}, ..., R_{\mathrm{i}})$, where $l(i)$ denotes a length of production route. The operations times are given by a sequence $ZT_{\mathrm{i}} = (r_{i1}, r_{i2}, ..., r_{il(i)})$, where $r_{i1}, r_{i2}, ..., r_{il(i)} \in N$ are defined in the uniform time units (N – set of natural numbers). For instance the system shown in Fig.1 consists of six resources and five processes. The resources R_3, R_4, are shared ones, since each one is used by at least two processes, and the resources R_1, R_2, R_5, R_6 are non-shared because each one is exclusively used by only one process. The processes P_1, P_2, P_3, P_4, P_5 are executing operations using resources given by the sequences: $Z_1 = (R_1, R_3)$, $Z_2 = (R_2, R_3)$, $Z_3 = (R_3, R_4)$, $Z_4 = (R_4, R_5)$, $Z_5 = (R_4, R_6)$.

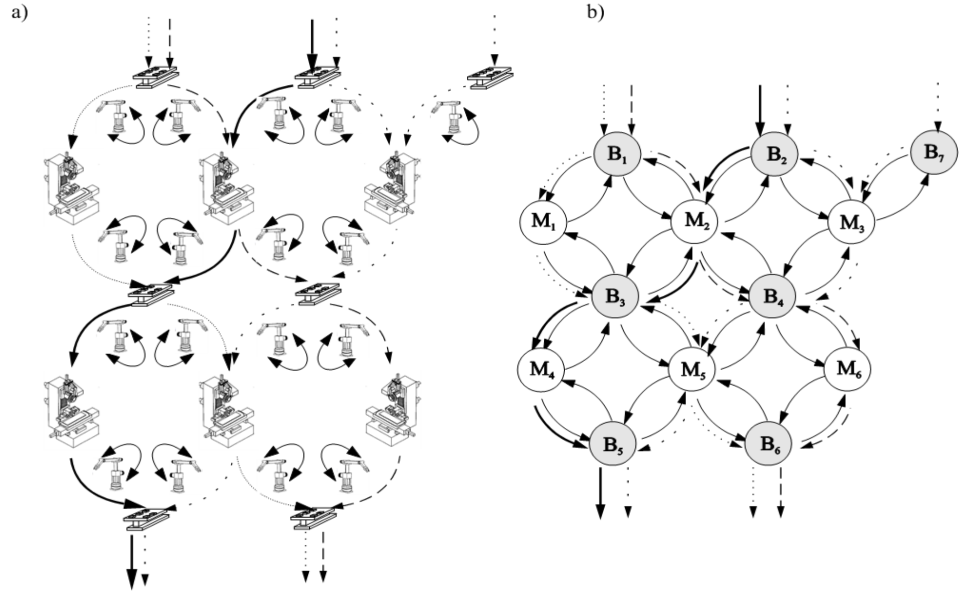


Fig.2. Structure of the Flexible Manufacturing System; a) the fractal-like layout structure, b) the graph model of the fractal-like layout structure

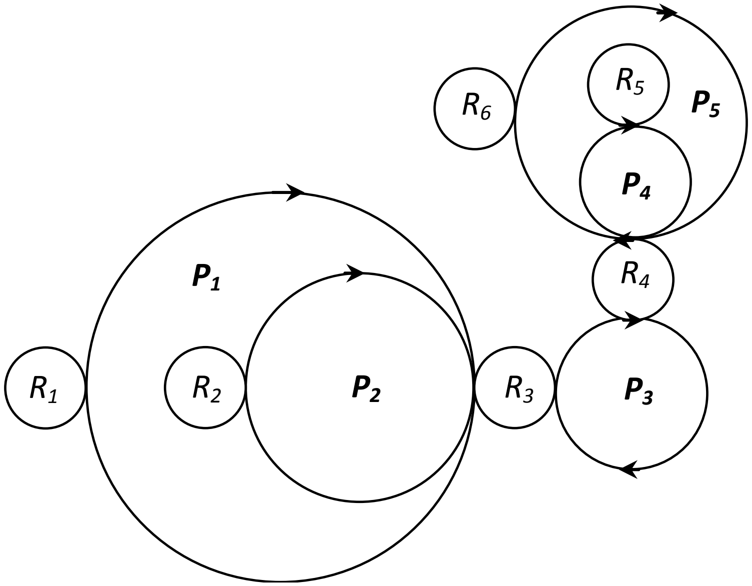


Fig.3. Repetitive concurrent processes

Different problems can be stated for the considered class of systems. For example, the following questions can be formulated [1], [2]: Does there exist such an initial processes allocation that leads to a steady state in which no process waits to the access to the common shared resources? What set of priority dispatching rules assigned to shared resources guarantee, if either, the same rate of resources utilization?

In general case the questions may regard to the qualitative features of system behaviour such as deadlock and/or conflicts avoidance [8], [12]. For example they may be aimed at conditions satisfaction of which guarantees system's repetitiveness for a given initial state and/or the dispatching rules allocation.

In order to illustrate the Diophantine's character of the model considered let us consider the system of concurrently flowing cyclic processes shown in Fig. 4. At the initial state (see Fig. 4 a) the steady state cyclic system behaviour, illustrated by Gantt's chart (see Fig. 4 b), is characterized by periodicity T=5 (obtained under assumption $t_{C1}= t_{E1} = t_{F1} = t_{C2}= t_{B2} = t_{D2} = t_{D3}= t_{A3} = t_{E3} =1$).

Note that besides of the initial processes allocation (see the sequence $S_0 = (F,C,A)$) the cyclic steady state behavior depends on routings direction as well as priority rules determining the order in which processes made their access to the common shared resources (for instance in case of resource D, $\sigma_D = \sigma(P_2,P_3)$ – the priority rule determines the order in which processes can access to the common shared resources D, i.e. at first to the process P_2, then to the process P_3, then P_2 and once again to P_3, and so on). For instance changing the priority rules into $\sigma_C = \sigma(P_2,P_1)$, $\sigma_D = \sigma(P_2,P_3)$, $\sigma_E = \sigma(P_3,P_1)$ is guaranteeing the cyclic steady state behavior, and if $\sigma_C = \sigma(P_2,P_1)$, $\sigma_D = \sigma(P_3,P_2)$, $\sigma_E = \sigma(P_1,P_3)$, then the resultant state is a deadlock one. In turn, for the initial state of processes allocation $S_0 = (F,B,A)$ (i.e., the process P_1 is allocated to the resource F, the process P_2 to the resource B , and the process P_3 to A) the resultant steady state is characterized with the periodicity $T = 6$ (see Fig. 5).

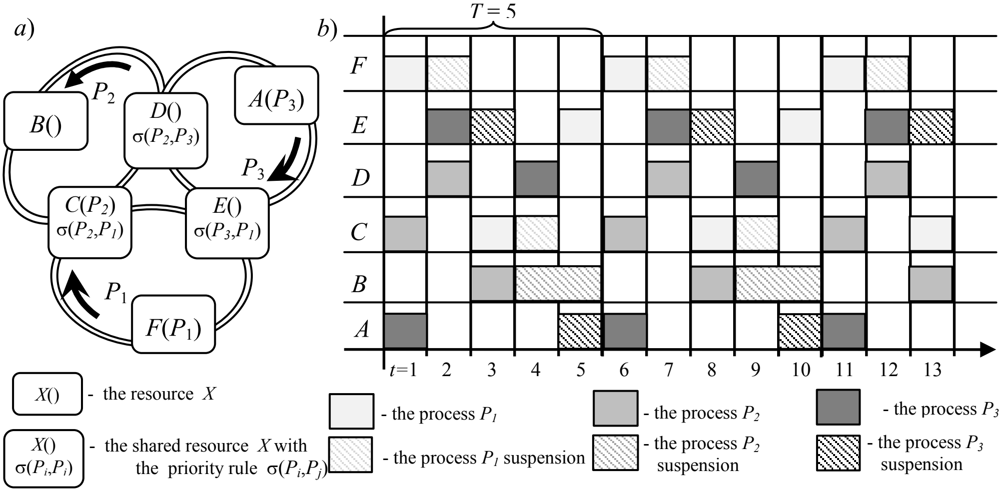


Fig.4. System of concurrently flowing cyclic processes: initial state a), Gantt's chart encompassing the cyclic steady state b)

In general case the periodicities of the cyclic steady states as well as corresponding sets of dispatching rules can be calculated from the linear Diophantine equation of the following form: $3y + 3x = T + z$. The formulae considered have been obtained due to the following transformations following the assumptions below:

- the initial state and set of dispatching rules guarantee there exists admissible solution (i.e. cyclic steady state)
- the structure of the graph model is consistent

Consider the following set of equations:

i) $x\cdot (t_{C1}+ t_{E1} + t_{F1}) + y\cdot t_{C2} + z\cdot t_{E3} = T$

ii) $y\cdot (\mathrm{t}_{C2}+ \mathrm{t}_{B2} + t_{D2}) + x\cdot t_{C1} + z\cdot t_{D3} = T$

iii) $z\cdot (t_{D3}+ t_{A3} + t_{E3}) + x\cdot t_{E1} + y\cdot t_{D2} = T$

where:

t_{ij} – the execution time of the operations executed on the i-th resource along the j-th process,

T – periodicity of the system of concurrently executed cyclic processes.

Subtracting the equation iii) from the equation ii) the resulting equation has the form:

$$y\cdot t_{C2}+y\cdot t_{B2}+x\cdot t_{C1}–z\cdot t_{A3}–z\cdot t_{E3}–x\cdot t_{E1} = 0$$

and after its adding to the equation i), the resultant formulae has the form:

$$y\cdot(2\cdot t_{C2}+ t_{B2}) +x\cdot(2\cdot t_{C1} + t_{F1})–z\cdot t_{A3}= T\,.$$

Consequently, the obtained equation:

$$y\cdot N +x\cdot M = T + z\cdot K$$

is Diophantine equation, where $N = M = 3$ and $K = 1$ under the following assumption $t_{C1} = t_{E1} = t_{F1} = t_{C2} = t_{B2} = t_{D2} = t_{D3} = t_{A3} = t_{E3} = 1$, i.e. takes the following form: $3y + 3x = T + z$. First three solutions are shown in the Table 1.

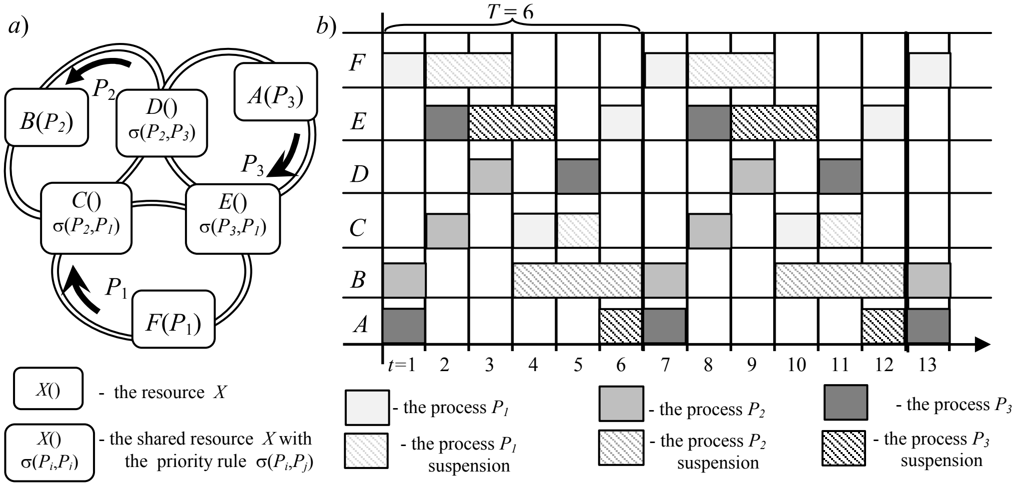


Fig.5. Gantt's chart of the cyclic steady state with periodicity equal to 6 units of time

Tab. 1. First three solutions of the Diophantine equation: $3\boldsymbol{y} + 3\boldsymbol{x} = \boldsymbol{T} + \boldsymbol{z}$

T	***x***	***y***	***z***
5	1	1	1
6	1	2	3
10	2	2	2

Other examples of systems composed of concurrently flowing cyclic processes follow from traffic route organization, e.g. employing the "green wave" concept. The main goal of centralized or decentralized traffic light control is just response to the question: Does there exist such control guaranteeing each route in each direction provides "green wave" flow vehicles? In some sense an alternative question is: Does there exist such traffic light control guaranteeing some places on a urban traffic route map can be linked by "green waves"? Solution to the last question, using the" ring-like green wave" concept, is shown in Fig. 6. That is easy to observe (see the dashed line) there exist possibility to synchronize three different "ring-like green waves" as to guarantee the "green wave" connecting two apartment blocks A and C. Such solution seems to be of great importance in cases caused by sport or show as well as morning or afternoon rush hour management.

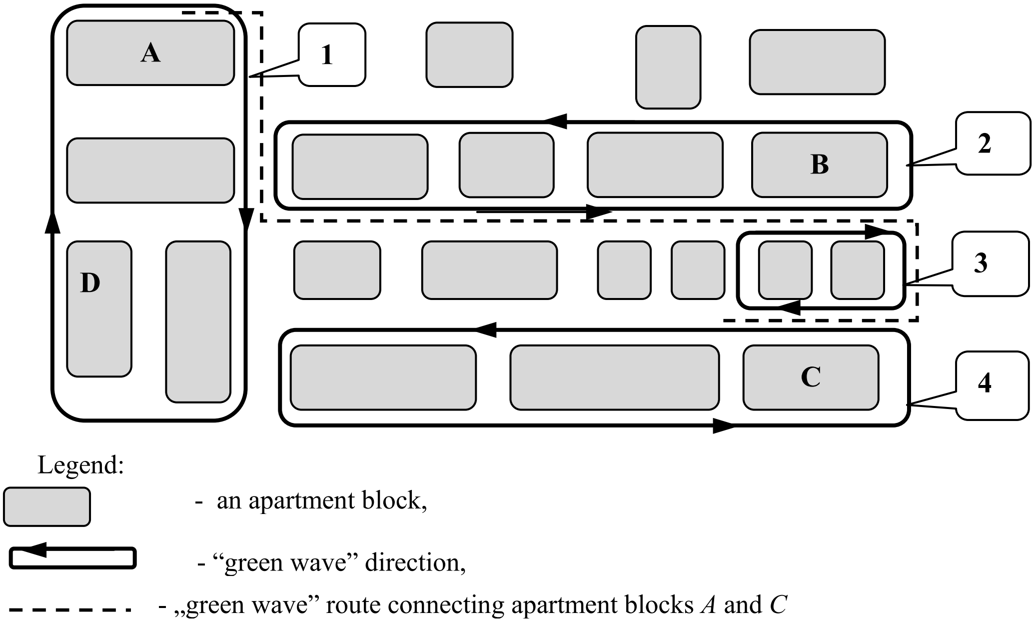


Fig.6. Idea of traffic route control based on the „ring-like green wave” concept

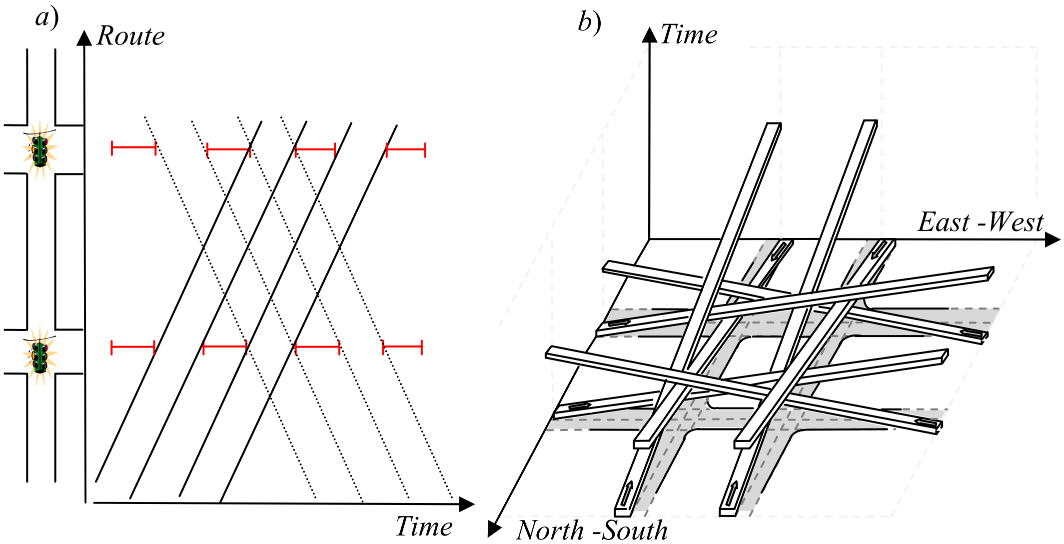


Fig.7. Illustration of the „green wave” traffic route control

In case of the first question, the response is positive, however in regular structures of routes and cross-routes. The solution follows from the observations that an organization of two directions ”green wave” route requires the same (in general case multiple of the some distance) distances between subsequent route-crosses (see Fig. 7 a)). In general case, the same regards of routes network, see Fig. 7 b).

3. DIOPHANTINE PROBLEMS

A Diophantine equation (named in honour of the 3rd-century Greek mathematician Diophantus of Alexandria) is an indeterminate polynomial equation that allows the variables to be integers only, i.e. an equation involving only sums, products, and powers in which all the constants are integers and the only solutions of interest are integers [14], [15]. For example, $3x + 7y = 1$ or $x^2 - y^2 = z^3$, where x, y, and z are integers (see Pythagoras' theorem or Fermat's last theorem). Besides of polynomial the exponential Diophantine equations where an additional variable or variables occur as exponents, can be considered, e.g., Ramanujan-Nagell equation, $2^n - 7 = x^2$.

Therefore, the Diophantine problems have fewer equations than unknown variables and involve finding integers, which satisfy all equations. The following questions belong to the most frequently asked: Are there any solutions?, Are there any solutions beyond some that are easily found by inspection?, Are there finitely or infinitely many solutions?, Can all solutions be found, in theory?, Can one in practice compute a full list of solutions?

The solvability of all Diophantine problems proposed in 1900, by David Hilbert and known as Hilbert's 10th problem has been settled negatively [6], [10]. In 1970, Matiyasevich proved that no general algorithm exists for determining whether a given Diophantine equation is soluble [15]. So, although such an algorithm does exist for the solution of first-order Diophantine equations in general case, however, Diophantine problems are unsolvable.

In that context, Diophantine equations fall into three classes: those with no solutions, those with only finitely many solutions, and those with infinitely many solutions. For example, the equation $6x - 9y = 29$ has no solutions, but the equation $6x - 9y = 30$, which upon division by 3 reduces to $2x - 3y = 10$, has infinitely many. In case of $x^2 + y^2 = z^2$ infinite set of solutions (3, 4, 5), (5, 12, 13), (7, 24, 25), (9, 40, 41),..., there exists, however, in case of $x^3 + y^3 = z^3$ the relevant set is empty. In turn, $x = 20$, $y = 10$ is a solution, and so is $x = 20 + 3t$, $y = 10 + 2t$ for every integer t, positive, negative, or zero. This is called a one-parameter family of solutions, with t being the arbitrary integer parameter.

In order to analyze the polynomial Diophantine equations the few general approaches, i.e. based on *Hasse principle* and *infinite descen*t method, are usually employed. In case of the exponential Diophantine equations, however, a similar general theory is not yet available so an ad hoc or trial and error methods are used so far. Consider a given set of Diophantine equations modelling the system considered. Assuming the set of variables and Diophantine equations type (polynomial or exponential) encompass system's structure while the set of solutions following given features of system functioning as system's behavior, one can state the following questions:

Does there exist a control procedure enabling to guarantee an assumed system behavior subject to system's structure constraints?

Does there exist the system's structure such that an assumed system behavior can be achieved?

Therefore, taking into account non decidability of Diophantine problems one can easily realize that not all behaviors can be obtained under constraints imposed by system's structure. The similar observation concerns the system's behavior that can be achieved in systems possessing specific structural characteristics. That means, the exhaustive searching for assumed control in the system at hand can be replaced by designing of a system aimed at the behavior requested.

4. TIMETABLING

As it was already mentioned the timetabling understood as classes planning can be seen as an example of cyclic scheduling problem. Below considered case, belonging to a class of multi-criteria decision problems can be also modeled as a Diophantine one.

Consider the set of lectures $P=\{A_1, B_1, A_2, B_2\}$. Lectures A_1 and A_2 belong to the group A, and lectures B_1 and B_2 belong to the group B. Given are two classes teams E_1 and E_2. To the team E_1 there are assigned 5 units of time regarding lectures form the group A (including A_1 and A_2) and 4 units of time from the group B (including B_1 and B_2). To the team E_2 there are assigned t 8 units of time regarding lectures form the group A and 6 units of time from the group B. Moreover given are periods limiting classes duration in particular teams. So, in case of the team E_1 the period limiting lecture A_1 is equal $t_{1,A1} = 2$; in case of B_1, $t_{1,B1} = 2$; in case of A_2, $t_{1,A2} = 3$; and B_2, $t_{1,B2} = 2$; the relevant duration times are included in the sequence $T_1 = (2,2,3,2)$. In case of the team E_2 the relevant duration times are denoted in the similar way and are included in the sequence $T_2 = (4,2,4,4)$. That means:

$$t_{1,A1}+t_{1,A2} = 5;\ t_{1,B1}+t_{1,B2} = 4;\ t_{2,A1}+t_{2,A2} = 8;\ t_{2,B1}+t_{2,B2} = 6.$$

Classes A_1 and B_1 are conducted by the lecturer L_1, while A_2 and B_2 by the lecturer L_2. The timetable sought should be free of gaps as for teams as for lecturers. In other words we are looking for permutations Q_1, Q_2 encompassing gaps free timetables:

$$Q_1 = (q_{1,1}, q_{1,2}, q_{1,3}, q_{1,4});\ q_{1,1} \neq q_{1,2} \neq q_{1,3} \neq q_{1,4};\ q_{1,1}, q_{1,2}, q_{1,3}, q_{1,4} \in P;$$
$$Q_2 = (q_{2,1}, q_{2,2}, q_{2,3}, q_{2,4});\ q_{2,1} \neq q_{2,2} \neq q_{2,3} \neq q_{2,4};\ q_{2,1}, q_{2,2}, q_{2,3}, q_{2,4} \in P.$$

Permutations Q_1, Q_2 do not allow the cases where the same classes A_1, A_2, B_1, B_2 are simultaneously conducted for different teams. That means, in case of A_1, for instance, the terms of this class beginning, i.e. $x_{1,A1}, x_{2,A1}$, in different teams, i.e., E_1, E_2 have to follow the formulae below:

$$(x_{1,A1} + t_{1,A1} \leq x_{2,A1}) \vee (x_{2,A1} + t_{2,A1} \leq x_{1,A1}).$$

Therefore, the course A_1 conducted for E_1 will either precede or succeed its repetition for E_2. The similar formulas can be considered for B_1, A_2, B_2:

$$(x_{1,B1} + t_{1,B1} \leq x_{2,B1}) \vee (x_{2,B1} + t_{2,B1} \leq x_{1,B1}),$$
$$(x_{1,A2} + t_{1,A2} \leq x_{2,A2}) \vee (x_{2,A2} + t_{2,A2} \leq x_{1,A2}),$$
$$(x_{1,B2} + t_{1,B2} \leq x_{2,B2}) \vee (x_{2,B2} + t_{2,B2} \leq x_{1,B2}).$$

Moreover, analogously the similar conditions can be formulated for lecturers L_1, L_2. Those mean, in case of L_1, for instance, the terms of his duties beginning $x_{1,A1}, x_{2,B1}, x_{2,A1}, x_{1,B1}$ and concerning classes A_1 and B_1, have to follow the formulas below:

$$\left(x_{1,A1} + t_{1,A1} \leq x_{2,B1}\right) \vee \left(x_{2,B1} + t_{2,B1} \leq x_{1,A1}\right),$$
$$\left(x_{2,A1} + t_{2,A1} \leq x_{1,B1}\right) \vee \left(x_{1,B1} + t_{1,B1} \leq x_{2,A1}\right).$$

Analogously the similar conditions can be formulated for the lecturer L_2. So, assuming the lecturer L_2 conducts the courses A_2, B_2, the relevant formulas are as follows:

$$\left(x_{1,A2} + t_{1,A2} \leq x_{2,B2}\right) \vee \left(x_{2,B2} + t_{2,B2} \leq x_{1,A2}\right),$$
$$\left(x_{2,A2} + t_{2,A2} \leq x_{1,B2}\right) \vee \left(x_{1,B2} + t_{1,B2} \leq x_{2,A2}\right).$$

In particular, the guarantee the timetable sought will gaps free requires the beginnings $x_{1,A1}$, $x_{1,B1}$, $x_{1,A2}$, $x_{1,B2}$, $x_{2,A1}$, $x_{2,B1}$, $x_{2,A2}$ of lectures A_1, A_2, B_1, B_2 have to follow lectures order constrains assumed by permutations Q_1 and Q_2. That means the following equations have to hold:

$$x_{1,q_{1,1}} \cdot x_{2,q_{2,1}} = 0,$$
$$x_{1,q_{1,i}} = x_{1,q_{1,i-1}} + t_{1,q_{1,i-1}} \qquad for\ i = 2,3,4,$$
$$x_{2,q_{2,i}} = x_{2,q_{2,i-1}} + t_{2,q_{2,i-1}}, \qquad for\ i = 2,3,4.$$

The above conditions guarantee the timetable for E_1, E_2 is gaps free. The similar equations have to hold for lecturers L_1, L_2. Therefore, in case of the lecturer L_1 the gaps free timetable requires the following equation holds:

$$\begin{aligned} t_{1,A1} + t_{2,B1} + t_{2,A1} + t_{1,B1} = \\ = \max\{&|x_{1,A1} - t_{2,B1} - x_{2,B1}|, |x_{1,A1} - t_{2,A1} - x_{2,A1}|, |x_{1,A1} - t_{1,B1} - x_{1,B1}|, \\ &|x_{2,B1} - t_{2,A1} - x_{2,A1}|, |x_{2,B1} - t_{1,B1} - x_{1,B1}|, |x_{2,A1} - t_{1,B1} - x_{1,B1}|\}, \end{aligned}$$

and for the lecturer L_2:

$$\begin{aligned} t_{1,A2} + t_{2,B2} + t_{2,A2} + t_{1,B2} = \\ = \max\{&|x_{1,A2} - t_{2,B2} - x_{2,B2}|, |x_{1,A2} - t_{2,A2} - x_{2,A2}|, |x_{1,A2} - t_{1,B2} - x_{1,B2}|, \\ &|x_{2,B2} - t_{2,A2} - x_{2,A2}|, |x_{2,B2} - t_{1,B2} - x_{1,B2}|, |x_{2,A2} - t_{1,B2} - x_{1,B2}|\}. \end{aligned}$$

Finally, the following equations have to hold in order to guarantee the gabs free timetable

$$
\begin{cases}
t_{1,A1} + t_{1,A2} = 5 \\
t_{1,B1} + t_{1,B2} = 4 \\
t_{2,A1} + t_{2,A2} = 8 \\
t_{2,B1} + t_{2,B2} = 6 \\
E\left(\left(x_{1,A1} + t_{1,A1} \le x_{2,A1}\right) \vee \left(x_{2,A1} + t_{2,A1} \le x_{1,A1}\right)\right) = 1 \\
E\left(\left(x_{1,B1} + t_{1,B1} \le x_{2,B1}\right) \vee \left(x_{2,B1} + t_{2,B1} \le x_{1,B1}\right)\right) = 1 \\
E\left(\left(x_{1,A2} + t_{1,A2} \le x_{2,A2}\right) \vee \left(x_{2,A2} + t_{2,A2} \le x_{1,A2}\right)\right) = 1 \\
E\left(\left(x_{1,B2} + t_{1,B2} \le x_{2,B2}\right) \vee \left(x_{2,B2} + t_{2,B2} \le x_{1,B2}\right)\right) = 1 \\
E\left(\left(x_{1,A1} + t_{1,A1} \le x_{2,B1}\right) \vee \left(x_{2,B1} + t_{2,B1} \le x_{1,A1}\right)\right) = 1 \\
E\left(\left(x_{2,A1} + t_{2,A1} \le x_{1,B1}\right) \vee \left(x_{1,B1} + t_{1,B1} \le x_{2,A1}\right)\right) = 1 \\
E\left(\left(x_{1,A2} + t_{1,A2} \le x_{2,B2}\right) \vee \left(x_{2,B2} + t_{2,B2} \le x_{1,A2}\right)\right) = 1 \\
E\left(\left(x_{2,A2} + t_{2,A2} \le x_{1,B2}\right) \vee \left(x_{1,B2} + t_{1,B2} \le x_{2,A2}\right)\right) = 1 \\
x_{1,q_{1,1}} \cdot x_{2,q_{2,1}} = 0 \\
x_{1,q_{1,i}} = x_{1,q_{1,i-1}} + t_{1,q_{1,i-1}}, \quad dla\ i = 2,3,4 \\
x_{2,q_{2,i}} = x_{2,q_{2,i-1}} + t_{2,q_{2,i-1}}, \quad dla\ i = 2,3,4 \\
t_{1,A1} + t_{2,B1} + t_{2,A1} + t_{1,B1} = \max\{R_1\} \\
t_{1,A2} + t_{2,B2} + t_{2,A2} + t_{1,B2} = \max\{R_2\}
\end{cases}
$$

where:

$$
\begin{aligned}
R_1 = \{&\left|x_{1,A1} - t_{2,B1} - x_{2,B1}\right|, \left|x_{1,A1} - t_{2,A1} - x_{2,A1}\right|, \left|x_{1,A1} - t_{1,B1} - x_{1,B1}\right|, \\
&\left|x_{2,B1} - t_{2,A1} - x_{2,A1}\right|, \left|x_{2,B1} - t_{1,B1} - x_{1,B1}\right|, \left|x_{2,A1} - t_{1,B1} - x_{1,B1}\right|\}, \\
R_2 = \{&\left|x_{1,A2} - t_{2,B2} - x_{2,B2}\right|, \left|x_{1,A2} - t_{2,A2} - x_{2,A2}\right|, \left|x_{1,A2} - t_{1,B2} - x_{1,B2}\right|, \\
&\left|x_{2,B2} - t_{2,A2} - x_{2,A2}\right|, \left|x_{2,B2} - t_{1,B2} - x_{1,B2}\right|, \left|x_{2,A2} - t_{1,B2} - x_{1,B2}\right|\}.
\end{aligned}
$$

Of course, the above delivered set of equations can be seen as typical set of nonlinear Diophantine equations. So, the question is whether it is decidable or not. Consequently, assuming that classes A_1, B_1 are conducted by the lecturer L_1, while classes A_2, B_2 conducted by L_2, the sought timetable should guarantee time windows free schedule, both, for all teams and all lectures. The Oz Mozart [13] implementation of that problem results in response: Lack of solutions. It follows, from the exhaustive search of all possible solutions, that the all available schedule posses either at least one free time window, in class of teams, or in class of lectures. The illustrative example is shown in Fig 8. In the case considered the lecturer L_2 has the free time window (1 unit of time long) between the class A_2, from the group E_1, and the class B_2 from the group E_2.

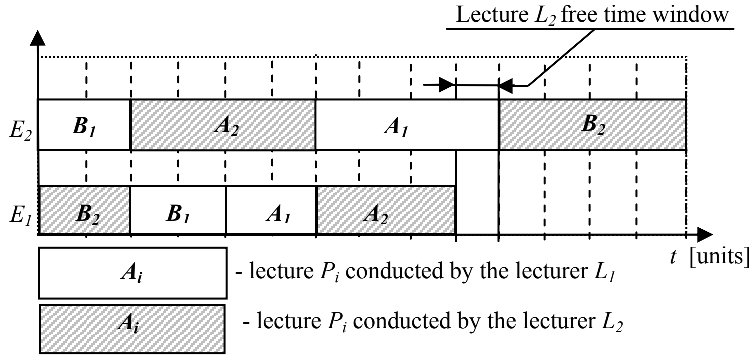


Fig.8. Gantt's chart with lecturer's free time window

Lack of feasible solutions still provides an opportunity to formulate so called reverse scheduling problem, i.e. the problem where besides of classes orders searched are their periods guaranteeing free time window schedules for both teams of students and lecturers. For the illustration let us consider the case assuming the periods of particular classes are unknown while the total sum of time units, devoted to the classes, remains the same as in the previous case, i.e.:

$t_{1,A1}+t_{1,A2} = 5$, (5 time units for group A of the team E_1);
$t_{1,B1}+t_{1,B2} = 4$, (4 time units for group A of the team E_1);
$t_{2,A1}+t_{2,A2} = 8$, (8 time units for group A of the team E_2);
$t_{2,B1}+t_{2,B2} = 6$, (6 time units for group A of the team E_2).

The considered question regards of: Does there exist such set of periods included in sequences T_1, T_2 as well as permutations of classes guaranteeing the schedules of student teams and lecturers do not contain free time Windows? The Oz Mozart implementation of that problem results in the following: the periods of classes corresponding to the teams E_1, E_2 are the following ones $T_1 = (3,3,2,1)$, $T_2 = (4,4,4,2)$. In case of the team E_1 the sequence of classes is the following one (A_1, B_1, A_2, B_2), while in case of E_2 the sequence of classes is the following one (A_2, B_2, A_1, B_1). The obtained solution is illustrated in Fig. 9.

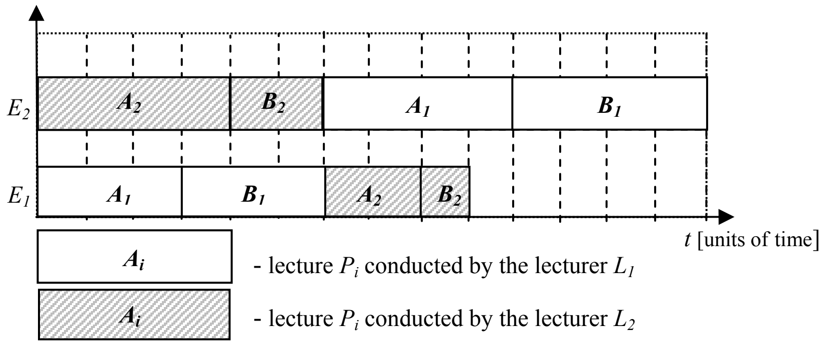
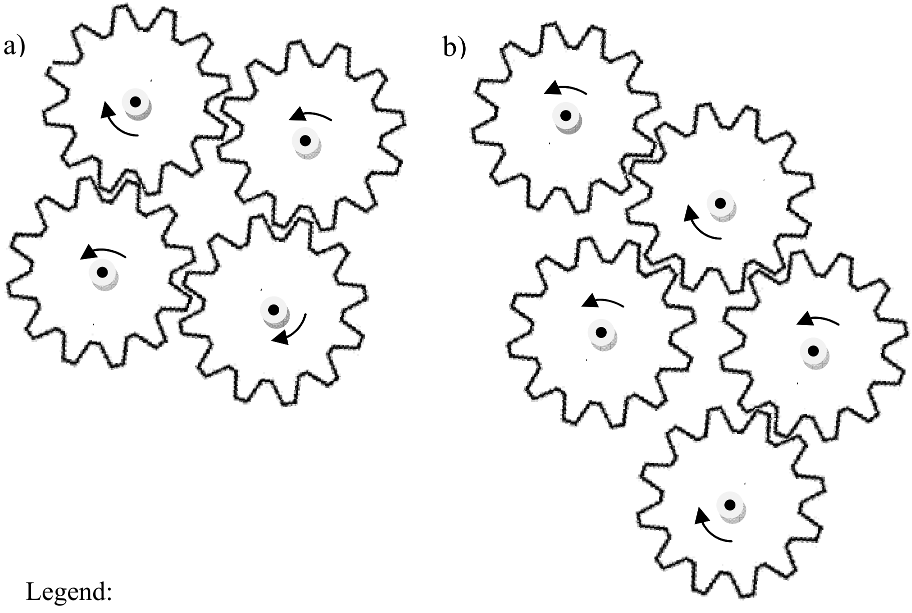


Fig.9. Gantt's chart without lecturer's free time window

That should be noted that the solution obtained, i.e. T_1 = (3,3,2,1), T_2 = (4,4,4,2) can be seen as a set of sufficient conditions guaranteeing required solution of the two criteria decision problem. Also the relationship between structure of scheduling problem (i.e. a number of teams, classes, periods, and so on) and its possible solutions encompassing the required system behavior (encompassing demanded features, e.g. lack of free time windows) can be easily observed. Consequently, it becomes quite obvious that not all behavior can be achieved in any system structure.

5. CONCLUDING REMARKS

Among many others real-life examples of cyclic processes the gear structure can be seen as a solution of some Diophantine problem. Consider cogwheel gear shown in Fig. 10 where cogwheels *A*, *B*, *C*, *D*, *E* have 20, 30, 15, 25 cogs respectively. In case of open chine structure see Fig, 10 b) the reduction of the three gears *A-B*; *B-D*, *D-E* equals to 20/30 30/15 15/25 = 0.8, as well as in the case of chine *A-B*; *B-D*, *D-C* that is 20/30 30/15 15/40 = 0.5, i.e. every two full revolutions of the cogwheel *A* correspond to just one full revolution of the cogwheel *C*.



Legend:
Z_i - the number of cogs of the i-th cogwheel; ↶ - direction of the cogwheel turn

Fig. 10. Examples of cogwheel structures: closed loop a), and open b)

Of course in case of the closed loop structure, see Fig. 10 a) the reduction is equal to 1:

$$Z_A/Z_B \cdot Z_B/Z_C \cdot Z_C/Z_D \cdot Z_D/Z_A = 1$$

where: Z_i – the number of cogs of the *i*-th cogwheel, $Z_i \in N$.

In case someone is looking for the period T of above consider cogwheel gears the following set of Diophantine equations has to take into account:

$$x \cdot Z_A = T;\ y \cdot Z_B = T;\ z \cdot Z_C = T;\ u \cdot Z_D = T$$

where: $x, y, z, u \in N$ – the numbers of full revolutions of relevant cogwheels in the period T.

Of course, $T = NWW(Z_A, Z_B, Z_C, Z_D)$, i.e. in case $x = y = z = u = n$ the gear consists of four same cogwheels $Z_A = Z_B = Z_C = Z_D$.

Note, however that closed loop structures have to consist of even n ≥ 4 numbers of cogwheels following the equation $Z_A/Z_B \cdot Z_B/Z_C \cdot Z_C/Z_D \cdot Z_D/Z_A = 1$. So, under these conditions someone may ask whether using a given amount of assumed cogwheels that is possible or not to design the cogwheel gear of encompassing the arbitrarily assumed period T.

The discussion presented explains while some timetabling like problems have an non decidable character. Diophantine models of sequential scheduling problems impose necessity for development of sufficient conditions guaranteeing the solvability of problems considered. Since Diophantine equations can be treated as a set of constraints, the constraint programming [1], [2], [3], i.e. descriptive in their character, languages can be directly implemented.

The alternative approach expresses itself in the possibility of reverse formulation of a timetabling problem. In both cases, i.e. In case of sufficient conditions development and strait, and reverse timetabling solution, the constraint programming environment seems to be well suited.

References

[1] BACH I., BOCEWICZ G., BANASZAK Z.: *Constraint programming approach to time-window and multi-resource-constrained projects portfolio prototyping*. In: Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2008, N.T. Nguyen et al. (Eds.), Lecture Notes in Artificial Intelligence 5027, Springer-Verlag, Berlin, Heidelberg, 2008, 767–776.

[2] BOCEWICZ G., BANASZAK Z., WÓJCIK R.: *Design of admissible schedules for AGV systems with constraints: a logic-algebraic approach*. In: Agent and Multi-Agent Systems: Technologies and Applications, Nguyen N.T., Grzech A., Howlett R.J., Jain L.C. (Eds.), Lecture Notes in Artificial Intelligence 4496, Springer-Verlag, Berlin, Heidelberg, 2007, 578-587.

[3] BARTÁK R.: *Incomplete Depth-First Search Techniques: A Short Survey*. Proceedings of the 6th Workshop on Constraint Programming for Decision and Control, Ed. Figwer J., 2009, 7-14.

[4] BIRGER RAA, EL-HOUSSAINE AGHEZZAF, WOUT DULLAERT: *Cyclic scheduling of multiple tours with multiple frequencies for a single vehicle*. International Journal of Logistics Systems and Management 2009 - Vol. 5, No.3/4 pp. 214 - 227

[5] CAI X., LI K.N.: *A genetic algorithm for scheduling staff of mixed skills under multi-criteria*. European Journal of Operational research, 125, 2000, 359-369.

[6] DAVIS, M.: *Hilbert's Tenth Problem is Unsolvable*. Amer. Math. Monthly 80, 1973, 233-269.

[7] ERNST A.T., JIANG H., KRISHNAMOORTHY M., OWENS B., SIER D.: *An annotated bibliography of personnel scheduling and rostering*. Annals of Operations Research, 127, 2009, 21-144.

[8] GAUJAL B., JAFARI M., BAYKAL-GURSOY M., ALPAN G.: *Allocation sequences of two processes sharing a resource*. IEEE Trans. on Robotics and Automation, Vol.11, No. 5, 1995, 748-353.

[9] GUY R. K.: *Diophantine Equations*. Ch. D in Unsolved Problems in Number Theory, 2nd ed. New York: Springer-Verlag, 1994, 139-198.

[10] JONES, J. P. AND MATIYASEVICH, YU. V.: *Exponential Diophantine Representation of Recursively Enumerable Sets*. Proceedings of the Herbrand Symposium, Marseilles, 1981. Amsterdam, Netherlands: North-Holland, 1982, 159-177.

[11] PINEDO M. L.: *Planning and scheduling in manufacturing and services.* Springer-Verlag, New York 2005

[12] POLAK M., MAJDZIK P., BANASZAK Z., WÓJCIK R.: *The performance evaluation tool for automated prototyping of concurrent cyclic processes*. Fundamenta Informatice, April, 2004, Editor-in-Chief, A. Skowron, ISO Press, Vol. 60, No.1-4, 2004, pp.269-289.

[13] SCHUTLE H., SMOLKA G., WURTZ J.: *Finite Domain Constraint Programming in Oz*. German Research Center for Artificial Inteligence, Germany, D-66123 Saarbrucken, 1998.

[14] SPRINDZUK, VLADIMIR G.: *Classical Diophantine equations*. Lecture Notes in Mathematics, 1559. Springer-Verlag, Berlin, 1993. 228 pp. ISBN 3-540-57359-3.

[15] SMART NIGIEL P.: *The Algorithmic Resolution of Diophantine Equations*. London Mathematical Society Student Text, 41. Cambridge University Press, Cambridge, 1998.

[16] SOON-KI HEO, KYU-HWANG LEE, HO-KYUNG LEE, IN-BEUM LEE, AND JIN HYUN PARK: *A New Algorithm for Cyclic Scheduling and Design of Multipurpose Batch Plants.* Ind. Eng. Chem. Res., 2003, 42 (4), 836–846.

[17] TAN B., KARABATI S.: *Stochastic Cyclic Scheduling Problem In Synchronous Assembly And Production Lines*. Journal of the Operational Research Society, Volume 49, Number 11, 1 November 1998 , 1173-1187.

[18] VON KAMPMEYER T.: *Cyclic scheduling problems*, Ph.D. Dissertation, Fachbereich Mathematik/Informatik, Universität Osnabrück, 2006.