MODELING AND OPTIMIZATION FOR HIGH-THROUGHPUT-SCREENING SYSTEMS

Eckart Mayer *,[1] **Jörg Raisch** **,*

* *Systems and Control Theory Group, Max-Planck-Institut Dynamik komplexer technischer Systeme, 39106 Magdeburg, Germany, Fax: +49-391-6110-399*
** *Lehrstuhl für Systemtheorie technischer Prozesse Otto-von-Guericke-Universität 39016 Magdeburg, Germany*

Abstract: The problem of cyclic scheduling under the requirement of throughput maximization is considered for a special class of cyclically repeated batch processes. All batches have to follow an identical time scheme. The same resource may be visited more than once by the same batch and time window constraints may be stated by the user. It is shown that the cyclic scheduling problem can be transformed into a mixed integer linear optimization problem. The method's application to High-Throughput-Screening processes is demonstrated.

Keywords: High-Throughput-Screening, cyclic scheduling, throughput maximization, mixed integer optimization, discrete event systems

1. INTRODUCTION

Throughput maximization problems are common in many processes in chemical industry as well as in transportation or manufacturing systems, where a large number of units, e. g. batches or workpieces, have to be handled one after each other in the shortest possible time. This contribution deals with throughput maximization for a special type of cyclic systems where all units have to be handled in exactly the same time scheme. The method is applied to High-Throughput-Screening (HTS) systems. However, it is also applicable to similar cyclic processes, e. g. in traffic engineering or for iterative batch processes in chemical engineering.

High-Throughput-Screening plants are used for the analysis of large numbers of substances, for example to analyze their benefit for a specific pharmaceutical, biological or agricultural application. Although several hundreds of substances are aggregated within one batch, i. e. on one so called microplate, a large number of batches have to pass through the plant resources, e. g. incubators, liquid handling devices, transportation devices etc., in the same specific time scheme. The task of throughput maximization is to find an operating sequence which allows to finish work for all microplates as fast as possible. The HTS scheduling problem differs from other scheduling problems, e. g. in manufacturing or chemical engineering (Schilling and Pantelides, 1999; Löhl *et al.*, 1998) as it combines the following requirements:

- Some resources may be revisited several times by the same batch.
- There are no buffers between the resources. In contrast, a batch will allocate two resources simultaneously while being transferred between the resources.

[1] Supported by CyBio AG, Jena and the German Ministry of Economics and Technology.

- All batches have to pass the system in the same time scheme.
- The time scheme may be restricted by *due dates* or *time window* constraints.

Scheduling methods exist for several fixed types of HTS plants, e.g. (Murray and Anderson, 1996; Donzel *et al.*, 1997). Nevertheless, because of the large variety of screening tasks performed on HTS plants, it is very important to have flexible plants with the possibility of re-arranging the machines and transportation devices in order to adapt them to the requirements of each specific screening task. Thus, this paper presents a general method which yields the time-optimal sequence for arbitrary machine arrangements and screening tasks.

In many cases, due to the specific nature of the substances to be screened, operating schemes have to be strictly cyclic. Thus, the method presented here will be limited to such strictly cyclic operation, where the time distance between two consecutive batches ('*cycle time*') is always constant. All resources have capacity one, i. e. each resource may be occupied by at most one batch at any one time. For such a system, the goal of throughput maximization is equivalent to the determination of the smallest possible cycle time and the corresponding batch time scheme complying with all constraints. The basic ideas for the solution of this scheduling problem have already been presented in (Mayer and Raisch, 2003). In this contribution, the method is generalized as the sequence of activities within the single batch is not fixed in advance.

This paper is arranged as follows: first, modeling of cyclic processes with respect to High-Throughput-Screening is discussed. Subsequently, the constraints for the scheduling problem are formulated. The scheduling problem is then cast into a mixed integer linear optimization problem. Finally, a specific application example is treated.

2. MODELING OF CYCLIC PROCESSES

Cyclic operating sequences as regarded here are characterized by the fact that all operations are repeated in a constant cyclic scheme. The cycles, called batches in chemical engineering, follow upon each other with constant time offset, called *cycle time* T. In High-Throughput-Screening, one batch consists of one or a couple of microplates passing through several work steps on several resources, e. g. incubators, liquid handling devices, transport devices etc. For the screening results to be meaningful and comparable, the time scheme for all batches has to be identical. Figure 1 gives a simple example for such a time scheme for one batch (called *single-batch time scheme*). It

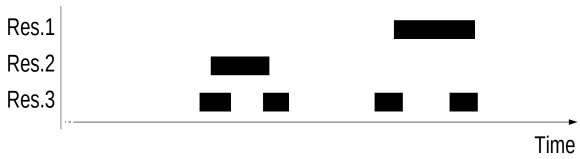

Fig. 1. Example for single-batch time scheme.

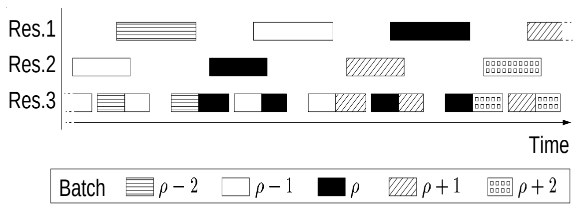

Fig. 2. Extract from cyclic schedule for Figure 1.

involves 6 activities on a total of 3 resources, pictured as a Gantt-Chart. In Figure 2, an extract of a cyclic schedule is illustrated. Different batches are displayed in different graphical patterns. Note that, due to the nature of the problem, the methods presented here do not need to account for the overall number of batches, nor do they need to identify a starting batch. Hence, batches are numbered by ρ, $\rho \in \mathbb{Z}$.

2.1 Modeling of Single-Batch Time Scheme

For the purpose of scheduling, the time scheme for a single batch is defined via the time instants at which each activity starts and ends [2]. In general, it consists of i_{max} activities, each allocating one of m resources, where J_i is the resource allocated by activity i, and n_j is the number of activities on resource j during a single-batch time scheme:

o_i	...	time, when activity i starts.
r_i	...	time, when activity i ends, $r_i > o_i$.
J_i	...	resource allocated by activity i, $J_i \in \{1 \dots m\}$.
m	...	overall number of resources.
n_j	...	number of activities on resource j.

For a cyclic schedule, the time instants for the ρ-th batch are given by:

$$
\begin{aligned}
o_i^{(\rho)} &= o_i + \rho \cdot T \\
r_i^{(\rho)} &= r_i + \rho \cdot T \ , \quad \rho \in \mathbb{Z} \ , \ \ i = 1 \dots i_{max} \ ,
\end{aligned}
\tag{1}
$$

where T is the constant cycle time.

[2] This is not necessarily identical to the moment in which the microplate enters resp. leaves the resource because there may be additional pre- or post-processing.

2.2 Parameterization

If the values for the variables o_i and r_i, $i = 1 \dots i_{max}$, are all predetermined, the problem can be solved by simple algorithms in polynomial time without the need of mixed integer optimization. However, in most cases the user will not determine the entire single-batch time scheme, but will only provide some of the activity start and end times and/or a number of linear constraints on the set of possible values for the variables o_i resp. r_i. The latter is to guarantee certain sequence constraints or to cope with chemical specifications (e. g. time windows for incubation times). The degrees of freedom that remain for the variables o_i and r_i are represented by K time variables $\theta_k \in \mathbb{R}_0^+$. The time instants o_i and r_i are then expressed as linear combinations of the θ_k:

$$\begin{aligned} o_i &= \underset{\sim}{o}_i + \sum_{k=1}^{K} (\chi_{i,k} \cdot \theta_k) \dots \text{Start of activity} \\ r_i &= \underset{\sim}{r}_i + \sum_{k=1}^{K} (\psi_{i,k} \cdot \theta_k) \dots \text{End of activity}\ . \end{aligned} \tag{2}$$

This means that the single-batch time scheme is described by fixed parameters $\underset{\sim}{o}_i$, $\underset{\sim}{r}_i$, $\chi_{i,k}$, and $\psi_{i,k}$, $i = 1 \dots i_{max}$, and yet unknown variables θ_k, $k = 1 \dots K$. The latter could, for example, be interpreted as delays which are inserted into the sequence of activities for a microplate.

The sequence and time window constraints on the variables o_i and r_i are represented by upper bounds for the time variables θ_k,

$$\theta_k \leq \theta_{k,max}\ ,\ \ k = 1 \dots K\ \ , \tag{3}$$

and, if necessary, by P additional linear constraints of the form

$$\sum_{k=1}^{K} (\kappa_{p,k} \theta_k) \leq \vartheta_p\ ,\ \ p = 1 \dots P\ \ . \tag{4}$$

The representation of the degrees of freedom in the single-batch time scheme by use of time variables θ_k leads to a significantly reduced problem size, because the number of variables o_i, r_i ($2i_{max}$ variables) is reduced to K variables θ_k, and usually $K << 2i_{max}$. Based on this description of the single-batch time scheme, the scheduling problem can now be formulated.

3. THE SCHEDULING PROBLEM

The task of finding a batch time scheme for a strictly cyclic schedule which allows for the smallest possible cycle time T, thus getting maximum possible throughput, will be called scheduling problem. As described in Section 2, two subsequent sample batches enter the plant with the fixed time offset T and are processed under the same basic time scheme. This can be formulated as an optimization problem.

First, some bounds for the cycle time T are formulated. Obviously, T is a strictly positive number, but a tighter bound can be deduced from the fact that if each single activity is finished as soon as possible and the busiest resource is allocated non-stop, no further speed increase is possible:

$$T \geq T_{min} = \max_j \Big(\min_{\theta_1 \dots \theta_K} \sum_{i=1}^{i_{max}} (r_i - o_i) \delta_{J_i j} \Big), \tag{5}$$
$$\text{where } \delta_{J_i j} = \begin{cases} 0 & \text{for } J_i \neq j \\ 1 & \text{for } J_i = j\ . \end{cases}$$

An *upper bound* for T can be prescribed by the user, or can be derived from the trivial case, in which no batch is started before the previous batch is completely finished, i. e.

$$T \leq T_{max} = \max_{\theta_1 \dots \theta_K} (\max_i r_i - \min_i o_i)\ \ . \tag{6}$$

Further constraints for T result from the fact that the cycle time can never be faster than the sum of allocation times (one batch) for any resource:

$$T \geq \sum_{i=1}^{i_{max}} (r_i - o_i) \delta_{J_i j}\ ,\ \ j = 1 \dots m\ , \tag{7}$$
$$\text{where } \delta_{J_i j} = \begin{cases} 0 & \text{for } J_i \neq j \\ 1 & \text{for } J_i = j\ . \end{cases}$$

3.1 Disjunctive Constraints

The solution to the scheduling problem has to meet the requirement that no two different activities are allowed to allocate the same resource simultaneously. These constraints will be called *disjunctive constraints*. Before deducing their formulation for the optimization problem, the term *nesting level* is introduced:

The nesting level $z_{(i1,i2)}$ for each combination of two activities $(i1, i2)$, $i2 > i1$, of the same resource is defined as

$$z_{(i1,i2)} = \lceil \frac{r_{i2} - o_{i1}}{T} \rceil - 1\ ,\ i2 > i1\ ,\ J_{i1} = J_{i2}\ , \tag{8}$$

where $\lceil x \rceil$ denotes the ceil-function, i.e. the smallest integer number that is greater or equal to x.

The nesting level is always an integer number i. e. $z_{(i1,i2)} \in \mathbb{Z}$. For $z_{(i1,i2)} \geq 0$, the nesting level can be interpreted as follows: when considering two activities $i1$ and $i2$ of the same resource

(i. e. $J_{i1} = J_{i2}$) belonging to the same batch, the nesting level $z_{(i1,i2)}$ indicates the number of activities $i1$ belonging to other batches which take place in between. Each set of values for the variables $z_{(i1,i2)}$ represents one possible *sequence* of activities in the overall schedule.

The requirement of ruling out the overlapping of two activities can be formulated as an exclusive OR term: for two activities $(i1, i2)$ of the same resource $J_{i1} = J_{i2}$ belonging to batch $\rho1$ and $\rho2$, respectively, the following condition ensures exclusion of overlapping:

$$o_{i1}^{(\rho1)} \geq r_{i2}^{(\rho2)} \ \text{ XOR } \ o_{i2}^{(\rho2)} \geq r_{i1}^{(\rho1)}. \tag{9}$$

This condition has to be met for all pairs of activities on the same resource and for all pairs of batch numbers, including activities belonging to the same batch and including the same activity in different batches ($i1 = i2,\ \rho_1 \neq \rho2$). Nevertheless, due to symmetry, it is sufficient to consider only cases $i2 > i1$, as well as the special case $\{i2 = i1 = i,\ \rho1 \neq \rho2\}$. Let us consider the latter case first. From (9), (1) and $r_i > o_i$, we get

$$\begin{aligned} r_i - o_i &\leq (\rho1 - \rho2) \cdot T \quad \forall(\rho1, \rho2)\ ,\ \rho1 > \rho2 \\ r_i - o_i &\leq (\rho2 - \rho1) \cdot T \quad \forall(\rho1, \rho2)\ ,\ \rho2 > \rho1\ , \end{aligned}$$

hence,

$$r_i - o_i \leq T \tag{10}$$

for all activities i, which is already guaranteed by Equation (7).

We now investigate case 2, where (9) has to hold for all $(\rho1, \rho2)$ and for all $(i1, i2)$, $J_{i1} = J_{i2}$, $i2 > i1$. As only cyclic sequences are considered, it is sufficient to ensure condition (9) for $\rho2 = 0$ and $\rho1 =: \rho$, $(\rho \in \mathbb{Z})$.

Equation (9) can then be reformulated using (1):

$$\begin{aligned} & o_{i1} + \rho T \geq r_{i2} && \text{(11a)} \\ \text{XOR} \quad & o_{i2} \quad\quad\ \geq r_{i1} + \rho T\ , \quad \rho \in \mathbb{Z}. && \text{(11b)} \end{aligned}$$

We now consider all possible values for ρ, distinguishing four cases.

(1) For $\rho = z_{(i1,i2)}$, condition (11a) will never be met. This means we need to ensure condition (11b):

$$z_{(i1,i2)} \cdot T - (o_{i2} - r_{i1}) \leq 0\ .$$

(2) For $\rho = z_{(i1,i2)} + 1$, condition (11b) will never be met. This means we need to ensure condition (11a):

$$\left(z_{(i1,i2)} + 1 \right) \cdot T - (r_{i2} - o_{i1}) \geq 0\ .$$

(3) For $\rho > z_{(i1,i2)} + 1$, condition (11a) is always met: from Definition (8), it immediately follows that

$$z_{(i1,i2)} + 1 \geq \frac{r_{i2} - o_{i1}}{T}\ .$$

Therefore

$$\rho > \frac{r_{i2} - o_{i1}}{T}\ ,$$

which, in turn, implies (11a). As condition (9) is satisfied, we don't need to introduce any further conditions into the optimization problem.

(4) For $\rho \leq z_{(i1,i2)} - 1$, condition (11b) is always met: from definition (8), it follows that

$$z_{(i1,i2)} < \frac{r_{i2} - o_{i1}}{T}\ . \tag{12}$$

Equation (7), ensures that $T \geq r_{i2} - o_{i2} + r_{i1} - o_{i1}$. Substituting (7) into (12) gives

$$z_{(i1,i2)} - 1 < \frac{o_{i2} - r_{i1}}{T}\ .$$

This implies

$$\rho < \frac{o_{i2} - r_{i1}}{T}\ ,$$

and therefore (11b). Again, condition (9) is satisfied. We don't need to introduce any further conditions into the optimization problem.

In summary, if (7) is satisfied, a necessary and sufficient condition for (9) to hold for all $(\rho1, \rho2)$ and for all $(i1, i2)$, $i2 > i1$, $J_{i1} = J_{i2}$), is:

$$z_{(i1,i2)} \cdot T - (o_{i2} - r_{i1}) \leq 0 \tag{13}$$

$$\left(z_{(i1,i2)} + 1 \right) \cdot T - (r_{i2} - o_{i1}) \geq 0\ . \tag{14}$$

Equations (13) and (14), together with $r_{i1} > o_{i1}$ and $r_{i2} > o_{i2}$, at the same time ensure that definition (8) is met.

3.2 MILP Formulation

We can now formulate our scheduling problem as an optimization problem.

In order to simplify notation, each possible pair of values for indices $(i1, i2)$, $i2 > i1$, $J_{i1} = J_{i2}$, is mapped to one value for a single index ι

$$\iota = 1 \ldots \iota_{max}\ ,\quad \iota_{max} = \sum_{j=1}^{m} \frac{n_j(n_j - 1)}{2}\ . \tag{15}$$

Hence, each value for ι signifies a pair of activities using the same resource.

The following abbreviations are introduced:

$$v_{\iota,k} = \chi_{i2,k} - \psi_{i1,k} \tag{16}$$

$$w_{\iota,k} = \psi_{i2,k} - \chi_{i1,k} \tag{17}$$

$$v_{\iota,0} = \underline{\varrho}_{i2} - \underline{r}_{i1} \tag{18}$$

$$w_{\iota,0} = \underline{r}_{i2} - \underline{\varrho}_{i1} \; . \tag{19}$$

Substituting (2) into (7) results in

$$b_{j,0} + \sum_{k=1}^{K} (b_{j,k} \cdot \theta_k) - T \le 0 \; , \quad j = 1 \ldots m \; , \tag{20}$$

where

$$b_{j,0} = \sum_{i=1}^{i_{max}} (\underline{r}_i - \underline{\varrho}_i) \delta_{J_i j}$$

$$b_{j,k} = \sum_{i=1}^{i_{max}} \sum_{k=1}^{K} (\psi_{i,k} - \chi_{i,k}) \delta_{J_i j}$$

$$\delta_{J_i j} = \begin{cases} 0 & \text{for } J_i \neq j \\ 1 & \text{for } J_i = j \; . \end{cases}$$

In order to formulate the scheduling problem as an optimization problem, we have to take the cycle time T as the objective function to be minimized under the constraints given by equations (13) and (14) as well as (3), (4), (5), (6) and (20). The search space for the optimization problem is defined by the following variables:

- cycle time $T \in \mathbb{R}^+$,
- time variables $\theta_k \in \mathbb{R}_0^+$,
- nesting levels $z_\iota \in \mathbb{Z}$.

Hence, using the abbreviations (16) to (19) and (2), the HTS scheduling problem can be written as the following optimization problem:

Min T over ($T \in \mathbb{R}^+$, $\theta_k \in \mathbb{R}_0^+$, $z_\iota \in \mathbb{Z}$) (NP1)
s. th.

$$z_\iota \cdot T - v_{\iota,0} - \sum_{k=1}^{K} (v_{\iota,k} \cdot \theta_k) \le 0 \tag{NP2}$$

$$\Big(z_\iota + 1 \Big) \cdot T - w_{\iota,0} - \sum_{k=1}^{K} (w_{\iota,k} \cdot \theta_k) \ge 0 \tag{NP3}$$

for $\iota = 1 \ldots \iota_{max}$

$$\theta_k \le \theta_{k,max} \; , \; k = 1 \ldots K \tag{NP4}$$

$$\sum_{k=1}^{K} (\kappa_{p,k} \cdot \theta_k) \le \vartheta_p \; , \; p = 1 \ldots P \tag{NP5}$$

$$T_{min} \le T \le T_{max} \tag{NP6}$$

$$b_{j,0} + \sum_{k=1}^{K} (b_{j,k} \cdot \theta_k) - T \le 0 \; , \; j = 1 \ldots m \tag{NP7}$$

Clearly, (NP1) to (NP7) constitutes a mixed integer nonlinear program (MINLP), i. e. a nonlinear optimization problem consisting of real and integer variables. There exist several solvers for MINLP optimization problems with different advantages and disadvantages. However, this task is very complex and can result in long computing times, even for small systems. Fortunately, it is possible to transform the MINLP into a linear formulation using

$$\bar{T} := \frac{1}{T} \; , \quad \bar{\theta}_k := \frac{\theta_k}{T} \; , \quad k = 1 \ldots K \; . \tag{21}$$

This leads to the following mixed integer linear program (MILP):

Max $\bar{T}$ over ($\bar{T} \in \mathbb{R}^+$, $\bar{\theta}_k \in \mathbb{R}_0^+$, $z_\iota \in \mathbb{Z}$) (LP1)
s. th.

$$z_\iota - v_{\iota,0} \cdot \bar{T} - \sum_{k=1}^{K} (v_{\iota,k} \cdot \bar{\theta}_k) \le 0 \tag{LP2}$$

$$z_\iota + 1 - w_{\iota,0} \cdot \bar{T} - \sum_{k=1}^{K} (w_{\iota,k} \cdot \bar{\theta}_k) \ge 0 \tag{LP3}$$

for $\iota = 1 \ldots \iota_{max}$

$$\bar{\theta}_k - \theta_{k,max} \, \bar{T} \le 0 \; , \; k = 1 \ldots K \tag{LP4}$$

$$\sum_{k=1}^{K} (\kappa_{p,k} \cdot \bar{\theta}_k) - \vartheta_p \, \bar{T} \le 0 \; , \; p = 1 \ldots P \tag{LP5}$$

$$\frac{1}{T_{max}} \le \bar{T} \le \frac{1}{T_{min}} \tag{LP6}$$

$$b_{j,0} \cdot \bar{T} + \sum_{k=1}^{K} (b_{j,k} \cdot \bar{\theta}_k) - 1 \le 0 \; , \; j = 1 \ldots m \tag{LP7}$$

Such an optimization problem can be solved [3] by advanced branch-and-bound techniques, for example by using the CPLEX library (http://www.ilog.com/products/cplex).

3.3 Adding Bounds for Integer Variables

For assays with a large number of activities, the optimization problem may become rather complex [4] which can result in very long computation times. Therefore it can be helpful to add additional bounds for the integer variables z_ι to the problem (LP1) to (LP7). This can be done as follows:

$$z_{\iota,min} \le z_\iota \le z_{\iota,max} \; , \tag{22}$$

where [5]

---

[3] Attention has to be paid to numerical aspects during optimization runs due to the fact that now the reciprocal of the original objective function is used.

[4] For n_j activities on a resource j, the number of pairs is $\frac{n_j (n_j - 1)}{2}$.

[5] $\lfloor x \rfloor$ denotes the floor-function, i.e. the largest integer number that is less or equal to x.

$$z_{\iota,min} := \begin{cases} \lceil \frac{\underline{W}_\iota}{T_{min}} \rceil - 1 & \text{for} \quad \underline{W}_\iota < 0 \\ \lceil \frac{\underline{W}_\iota}{T_{max}} \rceil - 1 & \text{for} \quad \underline{W}_\iota \geq 0 \end{cases}$$

$$z_{\iota,max} := \begin{cases} \lfloor \frac{\bar{V}_\iota}{T_{max}} \rfloor & \text{for} \quad \bar{V}_\iota \leq 0 \\ \lfloor \frac{\bar{V}_\iota}{T_{min}} \rfloor & \text{for} \quad \bar{V}_\iota > 0 \end{cases}$$

$$\underline{W}_\iota = \min_{\theta_1 \dots \theta_K} \left(w_{\iota,0} + \sum_{k=1}^{K} w_{\iota,k} \cdot \theta_k \right)$$

$$\bar{V}_\iota = \max_{\theta_1 \dots \theta_K} \left(v_{\iota,0} + \sum_{k=1}^{K} v_{\iota,k} \cdot \theta_k \right) .$$

Note that introducing these bounds only removes parts of the search space where at least one of conditions (LP1) to (LP7) is not satisfied, i. e. the feasible region of the optimization problem is not reduced.

4. APPLICATION

The proposed method has been applied to a number of HTS tasks (assays) in the pharmaceutical industry. An example for such an assay is given in the following: it consists of $m = 12$ resources with

$$\begin{aligned} n_j &= 8 \,, \quad j = 1, 12 \\ n_j &= 5 \,, \quad j = 5, 11 \\ n_j &= 2 \,, \quad j = 2 \dots 4 \\ n_j &= 1 \,, \quad j = 6 \dots 10 \end{aligned}$$

i. e. there are eight activities on the first resource, two activities on the second resource etc. The total number of pairs of activities on the same resource, i. e. the number of integer variables z_ι, is $\iota_{max} = 79$ (see Equation (15)). For this example the number of real variables θ_k is $K = 30$. This number follows from the time window constraints specified by the user. Empiric approaches (e. g. shifting of activities or insertion of delays in order to find solutions with small cycle times) will normally only provide suboptimal results for such problems. The mixed integer linear program for this example consists of 30+1 real valued variables and 79 integer variables as well as 204 linear inequality constraints. A globally optimal solution has been found using GAMS/CPLEX (http://www.gams.com) requiring a calculation time of only a few seconds.

In general, it is not possible to provide guaranteed computation time bounds for solving mixed integer optimization problems, since slight changes in the problem structure can have significant impact on overall computing times of solver algorithms. Nevertheless, for all HTS scheduling problems we considered, the computing times have found to be highly acceptable for the user.

5. CONCLUSION

The problem of finding a time optimal schedule for cyclically repeated batch processes with revisited resources and time window constraints has been treated. It has been shown that this scheduling problem can be modeled as a mixed integer linear optimization program (MILP). Problem instances (assays) for real High-Throughput-Screening (HTS) plants result in optimization programs of reasonable size and structure for which a globally optimal solution can be found within short calculation time.

With respect to HTS, two aspects have not been treated in this paper. The first is how to treat multiple assays or resources with a capacity greater than one. The second is the description of methods for casting the user defined time window constraints into the linear representation (2) using time variables θ_k such that a minimum number of variables is needed.

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding by CyBio AG, Jena, Germany and the German Ministry of Economics and Labour via the PRO INNO project 'Just-In-Time Optimierung verteilter Prozessabläufe im HTS-Labor'.

REFERENCES

Donzel, A., J. Carmona and L.A. Corkan (1997). Perspectives on scheduling. In: *High throughput screening* (J.P. Devlin, Ed.). Chap. 33, pp. 525–544. Marcel Dekker Inc. New York.

Löhl, T., C. Schulz and S. Engell (1998). Sequencing of batch operations for a highly coupled production process: Genetic algorithms versus mathematical programming. *Computers and Chemical Engineering* **22, Suppl.**, 579–585.

Mayer, E. and J. Raisch (2003). Time-optimal scheduling for high throughput screening processes using cyclic discrete event models. In: *Proc. 4th Mathmod, Vienna, 2003*. pp. 1349–1356.

Murray, C. and C. Anderson (1996). Scheduling software for high-throughput screening. *Laboratory Robotics & Automation* **8**(5), 295–305.

Schilling, G. and C.C. Pantelides (1999). Optimal periodic scheduling of multipurpose plants. *Computers and Chemical Engineering* **23**, 635–655.