On Cyclic Plans for Scheduling a Smart Card Personalisation System

Tim Nieberg

Universiteit Twente
Faculteit Elektrotechniek, Wiskunde en Informatica
Postbus 217
NL-7500 AE Enschede
e-mail: T.Nieberg@utwente.nl

Abstract. An industrial case study for scheduling the personalisation of smart cards is presented and analysed. Smart cards are personalised in several machines that are served by an underlying conveyor belt connecting these.
As there are usually a very high number of smart cards to be personalised, the focus is on cyclic schedules and the goal is to obtain a plan with high throughput. By characterizing certain schedules by the number of cards and free slots on the conveyor belt per cycle, non-trivial bounds on the cycle-time, and thus on the throughput, can be provided. This is done by looking at certain special scheduling policies, and using techniques known as destructive bounding. Additionally, with respect to the case study, optimality is proven for the input instances provided.

Keywords. AMETIST (Advanced Methods for Timed Systems), cyclic scheduling

1 Introduction

This article deals with an industrial case study provided by CYBERNETIX [1]. In this study, a machine environment for the personalisation process of smart cards is described. The main goal is to obtain a feasible schedule that maximizes the throughput of the system.

The smart cards that are to be personalised are placed on a conveyor belt. This belt is used to transport the cards to the different machines on which a card's embedded chip is processed, or a card is being printed. The specific machines for the different steps in the personalisation are either fed with cards from the belt or work directly on it.

Usually there is a high number of smart cards to be personalized, so that we focus on obtaining a periodic schedule that is repeated in the same fashion several times. This yields a cyclic scheduling problem, where maximizing the throughput is eqivalent to minimizing the cycle time for a fixed number of smart cards per cycle (cycle length).

For periodic schedules, we introduce an easy characterization by free slots on the conveyor belt per cycle. We prove several lower bounds on cycle times and use these to show optimality for several instances given by the industrial case study.

In [3], a model checking approach to create solutions to the problem is presented. There, the question wether the solutions given could be improved is posed. This article gives an answer to this question, by showing that the so-called single-mode schedule is optimal for the instances provided with the case study.

This article is organized as follows. In Section 2, we present a model of the machine environment. Periodic and feasible schedules for this model are characterized. Also, some further details of the case study are given. Section 3 provides several lower bounds on the cycle time, and presents some important scheduling policies. For the case study, optimal cyclic schedules for the personalisation system are discussed in Section 4. The paper concludes with on outlook on possible extensions and other solution approaches.

2 Problem Formulation

Given are n smart cards $J_1, \ldots, J_n$ that have to be personalized by going through the personalisation process described in this section. Furtheron, we refer to the smart cards as identical jobs, and we assume that the *blanc* smart cards are numbered according to the order in which they enter the system, i.e. from 1 to n.

Basically, the personalisation of each smart card is done in two parts: (1) personalisation of the chip on the card, and (2) graphical treatment of the card itself. Underneath the entire process, a conveyor belt is used to carry each card from one machine to another, and the machines are placed in a row at equal distance along this belt. A schematic view of the whole system is given in Figure 1. The conveyor belt can only move forward, and each movement one step further takes one time unit[1]. The distance at which the machines are placed corresponds to the distance the belt moves in a step forward. The cards are also placed at this distance onto the conveyor belt, so that we may refer to each place on the belt as a *slot*. One slot can carry at most one smart card.

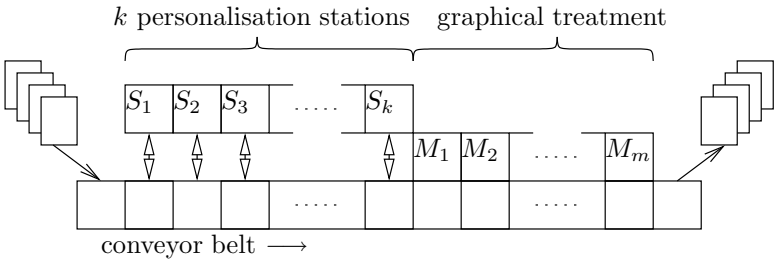

Fig. 1. The machine environment

The personalisation of a card's chip is done in a so-called personalisation station, of which there are k placed next to each other at the beginning of the system. These parallel machines work as follows. An unpersonalised card that is transported to the

[1] By proper scaling, the movement time of one unit can always be achieved

slot under a free personalisation machine is loaded from the belt into the machine. There, the data is transferred to the smart chip, and then the card is loaded back onto a free slot of the the conveyor belt. Involved in this are three operations, the personalisation itself, taking P_{pers} time units, and the loading and unloading of the card to the machine, taking P_{in} and P_{out} time units respectively. During the loading and unloading, the conveyor belt may not move. Let the i^{th} personalisation station in the row be denoted by S_i.

After a card is personalised in one of the personalisation machines, it proceeds to the graphical treatment. Here, m machines $M_1, \ldots, M_m$ are placed at equal distance along the belt. Each card has to be processed by each machine M_i for p_i time units, and the cards are processed on the conveyer belt. This forces the belt not to move during any processing of a card during the graphical treatment. This setup is somewhat similar to a flow-shop setup, however, the underlying conveyor belt forces the slowest machine to be waited for before the belt can move further. Let $p_{\max} = \max_{\{j=1,\ldots,m\}} p_j$ denote the longest processing time of the machines of the graphical treatment. Of course, if there is a free slot under the bottleneck machine(s), we may not have to wait $p_{\max}$ before advancing the conveyor belt.

In total, the conveyor belt thus has $k+m+2$ slots, i.e. one slot under each of the k personalisation stations, one slot on which each of the m machines of the graphical treatment may process a card, and a slot each for placing and removing a card at the beginning and the end of the system respectively.

For the whole personalisation process, there is an additional constraint which has to be satisfied. The order of the smart cards that go through the personalisation process has to be fixed for the process. This means that the jobs have to leave the systems in the same order as they entered it. As the cards cannot change places on the conveyor belt itself, this reduces to the constraint that the jobs have to leave the personalisation part of the system in the order in which they enter.

The objective is to obtain a feasible schedule with high throughput.

2.1 Characterization of Schedules

A schedule for the personalisation process of n smart cards can now be characterized as follows. For each smart card $J_i, i = 1, \ldots, n$, let

- σ_i^0 be the time at which the card J_i is placed onto the conveyor belt,
- $\pi_i \in \{S_1, \ldots, S_k\}$ denote personalisation station in which the card is personalised,
- $L_i^{IN}(L_i^{OUT})$ denote the time the card J_i is loaded into (removed from) the personalisation station π_i,
- $\sigma_i^{M_j}, j = 1, \ldots, m$, be the time at which the machine M_j is starting to process card J_i in the graphical treatment, and
- c_i denote the time at which the card J_i is removed from the personalisation system at the end of the conveyor belt.

Additionally, in order to account for the conveyor belt, let T be some ordered index set and let $t_1 > t_2 > \ldots > t_{|T|}$ be the times at which the belt advances one slot. For these advancement times, be $t^{-1}(x) \in T$ the index of the last advance of the belt before time x.

With the above values, it becomes possible to give the smart cards that are occupying a slot on the conveyor belt at any given time during the scheduling period. For this, denote by B^x_λ the occupancy of the slot λ (counted from the beginning of the belt), at time x, it is $\lambda \in \{1, \ldots, k+m+2\}$. Note that this occupancy is computed taking into account preceeding slots during earlier times, and the (un-)loading at the beginning, the personalisation stations and at the end of the belt.

A schedule as given above is then *feasible* if the following properties hold:

- The smart cards enter and leave the system in the same order, i.e. for all $i = 1, \ldots, n-1$ it is $\sigma^0_i < \sigma^0_{i+1}$ and $c_i < c_{i+1}$, and no two cards are placed onto the same slot of the belt, i.e. $t^{-1}(\sigma^0_i) < t^{-1}(\sigma^0_{i+1})$.
- Each card $J_i, i \in \{1, \ldots, n\}$, is personalised, i.e. $L^{IN}_i + P_{in} + P_{pers} \leq L^{OUT}_i$.
- Two smartcards $J_i, J_j \in \{1, \ldots, n\}$ with $i \neq j$ and $\pi_i = \pi_j$ are not personalized at the same time, i.e. $[L^{IN}_i, L^{OUT}_i] \cap [L^{IN}_j, L^{OUT}_j] = \varnothing$.
- At the time a (blanc) smart card $J_i, i \in \{1, \ldots, n\}$, is loaded into the personalisation station S_{π_i}, it is currently in the slot underneath S_{π_i}, i.e. $B^{L^{IN}_i}_{\pi_i+1} = J_i$.
- At the time a (personalised) smart card $J_i, i \in \{1, \ldots, n\}$, is unloaded again, the slot underneath the personalisation station S_{π_i} is not occupied by another card, i.e. $B^{L^{OUT}_i}_{\pi_i+1} = \varnothing$.
- At the time a smart card $J_i, i \in \{1, \ldots, n\}$, is being processed on a machine $M_j, j \in \{1, \ldots, m\}$ of the graphical treatment, it occupies the slot of M_j, i.e. $B^{\sigma^{M_j}_i}_{k+1+j} = J_i$.
- During the movement of the conveyor belt, no loading/unloading of personalisation stations, and no graphical processing may take place, i.e. for all $\varepsilon \in T, i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}$ we have
  - $[t_\varepsilon, t_\varepsilon + 1) \cap [L^{IN}_i, L^{IN}_i + P_{in}) = \varnothing$,
  - $[t_\varepsilon, t_\varepsilon + 1) \cap [L^{OUT}_i, L^{OUT}_i + P_{out}) = \varnothing$, and
  - $[t_\varepsilon, t_\varepsilon + 1) \cap [\sigma^{M_j}_i, \sigma^{M_j}_i + p_j) = \varnothing$.

The above definition of feasibility for a schedule represents the constraints given for the personalisation system. Note that due to the constraint of not placing a smart card onto a slot that is already occupied, both at the beginning and after the personalisation stations, each slot of the conveyor belt is occupied by at most one smart card during the entire planning period.

2.2 Cyclic Schedules

A schedule as defined above is called *periodic*, and thus defines a cyclic plan, if it can be repeated an arbitrary number of times, each time yielding the same sub-schedule.

Clearly, this is the case if there exist an $L \leq n$ and $\alpha > 0$ such that for each $l = 1, \ldots, n - L$

- $\sigma^0_l = \sigma^0_{l+L} + \alpha$,
- $\pi_l = \pi_{l+L}$,
- $L^{IN}_l = L^{IN}_{l+L} + \alpha, L^{OUT}_l = L^{OUT}_{l+L} + \alpha$,
- $\sigma^{M_j}_l = \sigma^{M_j}_{l+L} + \alpha$, for each $j = 1, \ldots, m$, and
- $c_l = c_{l+L} + \alpha$

holds. Then, we denote by α the *cycle time*, and by L the *cycle length* (in terms of smart cards per cycle).

For the smart card personalisation problem, any schedule involves placing the cards onto the conveyor belt, possibly with free slots in between. We therefore characterize cyclic schedules not only by the cycle length L for the number of smart cards per cycle, but additionally by the number f of free slots per cycle.

Definition 1. A cyclic schedule as characterized above that involves placing L smart cards onto the conveyor belt, and that uses f free slots, is called (L, f)-*cyclic schedule*.

In each cycle, the conveyor belt thus advances exactly $L+f$ times in an (L, f)-cyclic schedule.

Usually, we are interested in periodic schedules where the number of smart cards L per cycle is rather low compared to the total number of smart cards to be personalized. In fact, the above characterization does not depend on n and can be used to give a cyclic plan that might be executed an infinite number of times. For one cycle of this schedule, minimizing the cycle time α results in maximizing the throughput. The resulting throughput of the system is then L/α smart cards per time unit.

For the remainder of this article, we are interested in finding cyclic schedules with minimal cycle time.

2.3 The personalisation as given by CYBERNETIX

In the following, we briefly explain the practical example of the machine environment. For this, the case study provided by CYBERNETIX is fitted into the above framework of a smart card personalisation system, and real-world data is given to the processing times etc.

Additionally, the data from the case study is used to motivate some of the assumptions we make further on when describing optimal schedules for the personalisation system.

In the case study, the personalisation of the smart chips is done in k parallel personalisation stations, where $k \in \{4, 8, 16, 32\}$. The graphical treatment consists of 5 machines: two printers (PR), two flip-overs (FO), and a laser engraver (L). These are ordered PR→FO→PR→FO→L. All the processing times are given in Table 1.

Additionally, there are a few notes to be made with respect to the original case study provided by CYBERNETIX:

- At the beginning and at the end of the conveyor belt, there are two additional machines to place the cards onto the belt, and to remove them again. The processing times of these two machines, even though they are higher than e.g. loading the personalisation machines, are for now considered neglectible. Later on, in Section 4, we discuss these additional machines, and the resulting schedules, further.

action	index	time
belt movement	–	1
loading/unloading of personalisation station	P_{in}, P_{out}	1/2
personalisation	P_{pers}	10–50
printing	p_{PR}	3
flip-over	p_{FO}	3/2
laser engraving	p_L	4

Table 1. Processing times in the case study

- In between the personalisation stations and the graphical treatment, there exists a gap so that, for a single card to be transported from the personalisation part to the graphical treatment, there are additional steps needed. However, as we are interested in cyclic schedules, we may omit these additional belt slots. Note that later on, our arguements do not depend on the exact order of free slots, but on their number only.
- The *flip-over* machines in the real system occupy two slots on the belt: a card is flipped by being transported from the slot under the machine to the succeeding slot. However, in our model we assume only one slot under the machine and a processing time of p_{FO} for the turn-over process. It is easy to see that the two characterizations are equivalent.

3 Special Schedules and Bounds on the Cycle Time

In this section, we present some background on cyclic schedules and their length. By looking at the personalisation and the graphical treatment separately, we obtain two lower bounds on the cycle time. After that, we give two cyclic schedules that play an important role in the further discussion, especially with respect to the case study.

3.1 Lower Bounds on the Cycle Time

Consider the first part of the system consisting of the k personalisation stations $S_1, \ldots, S_k$. For a single personalisation station, the time needed to personalise a blanc smart card is $P_{pers}+P_{in}+P_{out}+1$. This can be seen by looking at the process: a card has to be loaded, personalised, and then unloaded again. For the next card to be personalised, at least one advancement of the conveyor belt is needed to remove the personalised card from underneath the respective personalisation station. We therefore obtain the following lower bound on the cycle time.

Claim 1. Any cyclic schedule has a cycle time of at least $P_{pers} + P_{in} + P_{out} + 1$.

Now consider the graphical treatment. Suppose we have a feasible (L, f)-cyclic schedule given. We then know the following.

- Obviously, the conveyor belt has to be moved $L + f$-times during the cycle, and during movement, no machine of the graphical treatment may process a smart card.

- Each card has to be processed on each machine, thus also on the machine with the longest processing time, say $M_i, i \in \{1, \ldots, m\}$. This gives an additional $L \cdot p_{\max}$ of processing in the cycle.

Overall, $L + f$ slots are presented to the bottleneck machine M_i. During the f slots where M_i is not processing a smart card, unless $f \geq m$, some other machine(s) have to process a smart card in a feasible schedule before the belt advances further. For all m machines of the graphical treatment, let F denote the maximal number of free slots under these m machines at any time. For example, consider a $(4, 1)$-cyclic schedule with $m = 5$ graphical machines, then at any time, at most one free slot is under a machine.

Now, consider the case that the F free slots are positioned under the F machines with the largest processing time. Clearly, this is best possible. Let $\max^F = \min \binom{\{p_1,\ldots,p_m\}}{m-F}$ denote the F^{th} largest processing time, then in the case that the F free slots are positioned under the F machines with the largest processing time, the belt has to wait $\max^F$ time units before advancing.

Summing up, we obtain the following bound.

Claim 2. An (L, f)-cyclic schedule has a cycle time of at least

$$(L + f) + (L \cdot p_{\max}) + (f \cdot \max^F),$$

with F and $\max^F$ as defined above.

For some instances, these bounds can be met and we thus obtain optimal cyclic schedules. Next, we present two such schedules and explain which additional constraints on the input instance are sufficient for their optimality.

3.2 Tight Loading

At a first glance, it seems reasonable to exploit every slot of the conveyor belt by placing a card on it. A possible schedule evolving from this strategy, which we call *tight loading*, is given as follows.

- After each advancement of the belt, each machine of the graphical treatment immediately starts processing the card underneath. Also, a new smart card is put onto the first slot at the beginning of the conveyor belt.
- If after an advancement of the belt, underneath the last personalisation station S_k, there is a smart card which has already been in one of the personalisation stations, advance the belt as soon as possible (i.e. after $p_{\max}$).
- If after an davancement of the belt, underneath S_k, there is a yet blanc card, load this card and all preceeding ones (also blanc) into the respective personalisation station and begin processing these. These cards are then unloaded as soon as the process in the personalisation stations is finished. Advance the belt as soon as possible, i.e. when both all graphical treatment machines are finished processing and the now processed cards are again placed onto the belt (i.e. after $\max\{p_{\max}, P_{pers} + P_{in} + P_{out}\}$).

An example for a personalisation system with 4 personalisation stations and 5 graphical machines is given in Figure 2. The cards are numbered according to their position in the cycle and are subscripted by the relative cycle number in which they entered the system, e.g. 2_1 denotes the second smart card of the current run of the cycle. We denote a blanc smart card by a superscript $\circ$. In (a), the situation at the beginning of the cycle is presented, (b) shows the machines just after the 4^{th} advancement of the belt after which the four new smart cards are loaded into the personalisation station. In (c), the end of the cycle is shown.

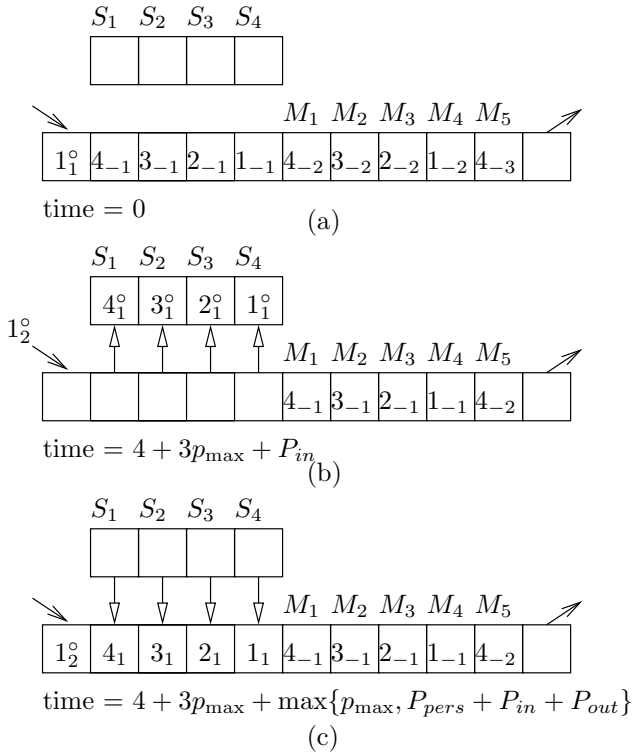

Fig. 2. Example of a cycle during *Tight Loading* for $k = 4$ and $m = 5$.

This tight loading schedule is a periodic schedule consisting of $L = k$ smart cards for each cycle. The number of free slots is 0, it is thus a $(k, 0)$-cyclic schedule. Obviously, for $L < k$, a $(L, 0)$-cyclic schedule is dominated by the above tight loading in the sense that a tight loading performes equally well or better. Also, any cyclic schedule of greater length ($L > k$) can be split up, resulting in a $(k, 0)$-cyclic schedule, as well.

We now show that it is unique, i.e. there does not exist another $(k, 0)$-cyclic schedule.

Theorem 1. *Suppose that the loading of the personalisation stations occurs directly after the respective advance of the conveyor belt. Any* $(k, 0)$*-cyclic schedule only*

loads (and unloads) the personalisation stations once during each cycle. Loading and unloading occurs from and to the same slot on the conveyor belt.

Proof. First note that loading the personalisation stations directly after the advance of the belt results in the same schedule[2] than waiting after the (blanc) smart cards have been transported underneath their respective personalisation station.

We show that the only feasible schedule is given by the tight loading. Suppose the opposite, and consider the first smart card J_0 of a cycle, w.l.o.g. this is a smart card that is not loaded onto the same slot after processing in the respective personalisation station. If at time σ_0^0, all personalisation stations are loaded, and each slot of the belt is also occupied, the card J_0 cannot enter any of these machines as these cannot unload any card onto the full belt. The schedule is thus not feasible.

Therefore, a belt slot or a personalisation station has to be free. Suppose there is a free slot on the belt. As there are no free slots entering the conveyor belt, no free slot can leave from underneath the personalisation stations. These free slots are eventually filled by a personalised smart card.

There is a free station, say S_i, in which J_0 is loaded. The resulting free belt slot then advances before J_0 is unloaded again. However, J_0 has to be unloaded onto a free slot, and as no free slots are entering the system, such a free slot has to be created by loading another (blanc) smart card into a preceeding personalisation station, say $S_j, j < i$, and moving the belt further. Repeating this arguement, now in order to move a free slot to the card now in station S_j, we obtain after at most k times a situation, where there is no preceeding personalisation station in order to create a free slot on the belt. □

For the tight loading schedule, we now have the following property.

Claim 3. If $P_{pers} + P_{in} + P_{out} \leq p_{\max}$, the tight loading schedule is optimal.

Proof. The tight loading schedule meets the bound given in Claim 2 ($f = 0$). □

From the above discussion, it is clear that if we want to improve the throughput, we have to utilize free slots into the cyclic schedule. It can be seen that, unless a tight loading of the belt is desired, at least one free slot per k smart cards has to be given in any schedule.

3.3 Single Mode

Here, we give a schedule that uses one free slot per k smart cards called *single mode*. This schedule is derived from a policy called *Super Single Mode* which was originally suggested by CYBERNETIX.

The single mode now begins with a free slot that is advanced under the first personalisation station at the beginning of each $(k, 1)$-cycle, followed by the k smart

[2] The schedule may be better in terms of throughput.

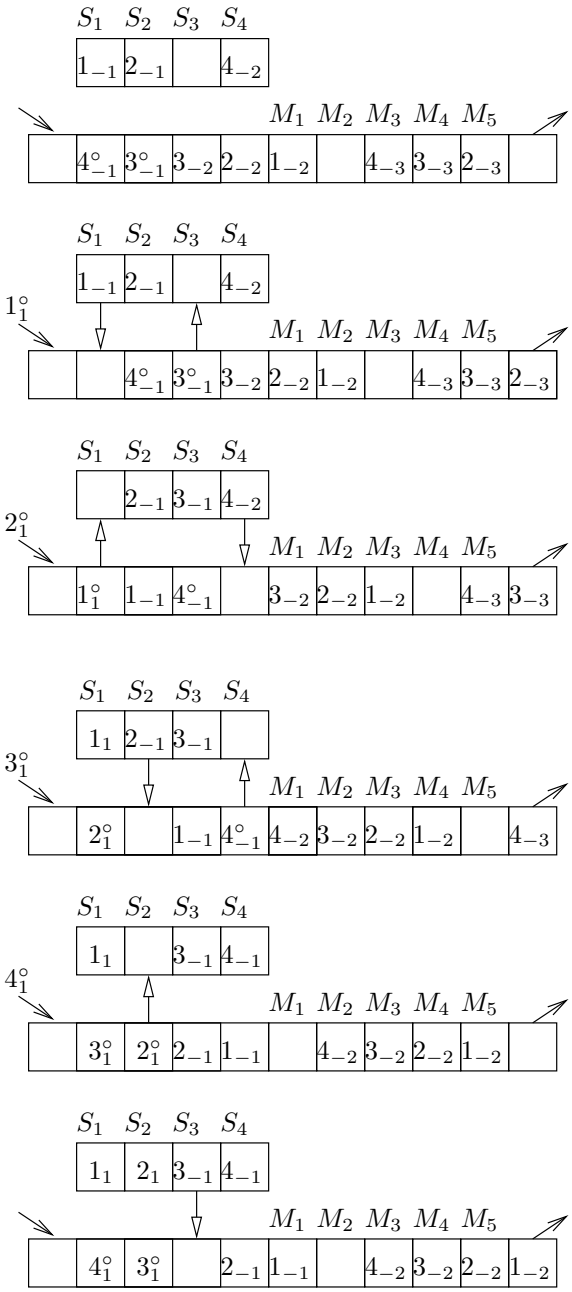

Fig. 3. Example of a cycle during *Single Mode* for $k = 4$ and $m = 5$.

cards without free slots in between. The working from there is more or less event-driven in the sense that the belt advances as soon as possible, i.e. when all tasks to be performed in between have completed.

These tasks are, of course, all the processes of the graphical treatment, furthermore some loading and unloading of personalisation stations. Each time a free slot advances underneath a personalisation station, the card is placed on the conveyor belt. Then, after the following advance, the new card placed on that slot is inserted into the personalisation station. By an inductive arguement, it is easy to see that the order of the cards is not affected by this procedure. In Figure 3, an example of a cycle in single mode is given.

By looking at a single personalisation station during one complete cycle, it can be seen that each personalisation station is utilized at all times except for one (required) advancement step of the conveyor belt. This yields the following sufficient condition for optimality of the single mode.

Claim 4. If $P_{pers} + P_{in} + P_{out} \geq k + k \cdot p_{\max} + \max^1$, the schedule given by the single mode is optimal.

Proof. In this case, the single mode has a cycle time of $P_{pers} + P_{in} + P_{out} + 1$, which is a lower bound on the cycle time. □

However, in other cases, the single mode may also provide optimal throughput of the system. In the next Section, we show that it is optimal for the instances corresponding to the case study.

4 Optimal Schedules for the Case Study

In this section, we focus on the case study provided by CYBERNETIX, and show that with the characterization done in the previous section, we are able to derive optimal cyclic schedules for the data given in the study. For the data already presented in Section 2, we show that the *single mode* always provides an optimal cyclic schedule, i.e. the overall best throughput of the system. In the case study, there are 5 graphical machines, and at least 4 personalisation stations.

4.1 Bounds

For the data of the case study, Table 2 gives the values for $\max^F$ and the resulting lower bounds (LB) on the possible (k, f)-cyclic schedules.

As shown in the preceeding section, the only possible periodic schedule without any free slots is given by tight loading. For the case study, tight loading results in $(k, 0)$-cyclic plans with cycle time as given in Table 3, note that $P_{pers} \geq 10$.

It can be seen that a tight loading schedule does not meet the lower bound on the cycle time. Therefore, we may only obtain a better schedule by inserting free slots into a cyclic plan, as in this case we have better bounds. For a cycle of k smart cards, with free slots, the best lower bounds for the graphical treatment part of

free slots per cycle	$\max^P$	LB	$k=4$	$k=8$	$k=16$	$k=32$
0	4	$k+4k=5k$	20	40	80	160
1	3	$(k+1)+4k+3=5k+4$	24	44	84	164
2	3	$(k+2)+4k+3\cdot 2=5k+8$	28	48	88	168
3	1.5	$(k+3)+4k+1.5\cdot 3=5k+7.5$	27.5	47.5	87.5	167.5
4	1.5	$(k+4)+4k+1.5\cdot 4=5k+10$	30	50	90	170
>5	0	$\geq (k+5)+4k=5k+5$	≥ 25	≥ 45	≥ 85	≥ 165

Table 2. Bounds on cycle time for the case study (cycle length = k).

k	4	8	16	32
Cycle time	$17+P_{pers}\geq 27$	$37+P_{pers}\geq 47$	$77+P_{pers}\geq 87$	$157+P_{pers}\geq 167$

Table 3. Cycle time for tight loading.

the system are given in Table 4. Resulting from previous arguments, the bounds presented hold true for the whole system given by the case study. Note that these bounds are obtained from a $(k,1)$-cyclic schedule as tight loading yields a plan with higher cycle time.

k	4	8	16	32
Cycle time	24	44	84	164

Table 4. Lower bounds for the case study (graphical treatment).

For the personalisation stations, we obtain from Claim 1 different lower bounds for different processing times of these machines given in Table 5.

P_{pers}	10	20	30	40	50
Cycle time	12	22	32	42	52

Table 5. Lower bounds for the case study (personalisation stations).

4.2 Optimal Plans

Consider the plan obtained by the single mode scheduling policy. Table 6 gives the cycle times for different values of P_{pers}. For each combination of personalisation time and number of personalisation stations, the values are given. All these plans meet a lower bound, denoted by a superscript P when coming from the personalisation stations as bottleneck (Claim 1), with a superscript G when coming from the bounds given in Table 4. Thus, these cycle times are optimal. In Table 7, the corresponding throughput is given.

P_{pers}	10	20	30	40	50
$k = 4$	24^G	24^G	32^P	42^P	52^P
$k = 8$	44^G	44^G	44^G	44^G	52^P
$k = 16$	84^G	84^G	84^G	84^G	84^G
$k = 32$	164^G	164^G	164^G	164^G	164^G

Table 6. Optimal cycle times.

P_{pers}	10	20	30	40	50
$k = 4$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{8}$	0.095	.077
$k = 8$	0.182	0.182	0.182	0.182	0.154
$k = 16$	0.19	0.19	0.19	0.19	0.19
$k = 32$	0.195	0.195	0.195	0.195	0.195

Table 7. Optimal throughput of personalisation process.

4.3 Loading and Unloading of the Conveyor belt

At the beginning and the end of the conveyor belt, there are two machines placed in the case study for loading blanc smart cards onto the belt, and removing them again at the end. These two machines take 2 time units each to perform their duties. In all the schedules persented above, in between two advances of the conveyor belt, there are at least 3 time units of time needed for processing during the graphical treatment. Thus, there is always enough time to place and remove the cards.

5 Conclusions and Future Work

In this article, we present a scheduling model for the personalisation of smart cards which comes from an industrial case study, and characterize cyclic schedules for this model. We derive several bounds on the cycle time, which is related to the throughput of the entire process. For the given scenarios of the case study, optimal cyclic schedules are given. Note that we have not used the order in which the machines are placed alongside the conveyor belt in the graphical treatment part. This order does thus not play a significant role in the arguments concerning the bounds or optimal plans given in this article.

However, the case study contains more, related scenarios like single-sided printing of the smart cards which are not discussed in this article. During single-sided printing, each card only needs to be printed in one of the two printers of the graphical treatment. Nevertheless, the arguements can be adjusted accordingly so that similar bounds as presented here can be derived. Also, the influence of damaged cards mentioned in the case study is not considered. Future work may focus on these scenarios.

Acknowledgement The author would like to thank Angelika Mader for presenting this problem and Johann Hurink for the discussions and useful hints in preparing this document.

References

1.ALBERT, S. (2002): CYBERNETIX Case Study Informal Describtion, Cybernetix Recherche, Marseille, France.
2.AMETIST (Advanced Methods for Timed Systems), IST project 2001-35304. http://ametist.cs.utwente.nl/.
3.
3.MADER, A. (2003): Deriving Schedules for a Smart Card Personalisation System, preprint.