Publ. Math. Debrecen
52 / 3-4 (1998), 337–342

Periodical scheduling

By JÁNOS DEMETROVICS (Budapest), KATALIN FRIEDL (Budapest)
and LAJOS RÓNYAI (Budapest)

Dedicated to Professors Zoltán Daróczy and Imre Kátai
on the occasion of their 60th birthday

Abstract. We solve a bandwidth-optimization problem for a broadcasting task in which certain messages are expected to be transmitted periodically. We determine the minimal feasible bandwidth with the aid of an integer making argument. We present a fast and simple greedy algorithm for the scheduling of the messages.

1. Description of the problem

On a communication channel an information provider broadcasts n messages $M_1, \dots , M_n$. The messages are updated from time to time and the updated versions have to be sent out periodically. One may think of stock quotations, currency exchange rates, weather reports etc., as the messages M_i.

The i-th message M_i consists of t_i blocks and it (its most recent version) has to be sent once between times kp_i and $(k+1)p_i$ for $k = 0, 1, \dots$. The number p_i is called the *period* of M_i. We assume that the length t_i of the messages does not change, and we refer to the updated versions of M_i as copies of M_i.

Mathematics Subject Classification: 68Q20, 90B35, 68M20.
Key words and phrases: communication algorithm, scheduling, integer making.
This work was supported in part by OTKA Grants 016503 and 016524, AKP Grant 96-350/24, and EC Grant ALTEC-KIT.

An optimal communication scheme tells us what the minimal bandwidth of the channel is which makes the periodical resending of messages possible. Moreover, it describes a scheduling algorithm for sending the blocks of messages. Here the bandwidth of the channel is measured by the number of blocks that can be transmitted in one unit of time.

The ith message divides the time to intervals of length p_i. Consider the case, when the period lengths p_i are integers. Let p denote the least common multiple of the p_i. Then time p is a common endpoint of some intervals for $i = 1, \dots, n$. Starting from time p the intervals look exactly as they did from time 0. Therefore it is sufficient to consider the time interval $[0, p]$, and look for a good communication scheme there. Then that scheme can be repeated as many times as it is necessary.

In a schedule for the time interval $[0, p]$ we are required to transmit p/p_i copies of the i-th message. Adding these up, we obtain that the number of blocks sent is $\sum_i t_i p/p_i$. Clearly, this requires that $m = \sum_i t_i/p_i$ blocks be transmitted during one unit of time, so the bandwidth is at least m. In this paper we consider the case when this "target bandwidth" m is also an integer. The easy (but somewhat technical) considerations needed to cover the more general settings will be treated elsewhere.

With this simplifying assumption we show that this bandwidth is always sufficient. The problem of sending the messages can be formulated as a scheduling problem. For the rudiments of scheduling terminology we refer to [3]. One can consider the periodically appearing copies of M_i as different jobs to be completed, with their own release dates and deadlines (describing the corresponding time interval). The problem is to find a feasible preemptive schedule for these tasks. With this approach the number of tasks would increase from n to $\sum_i p/p_i$. We show that this increase in the problem size can be avoided. With the aid of an "integer-making lemma" we establish first, that bandwidth m is sufficient.

2. An integer-making lemma

Let $a_1, \dots, a_m$ be real numbers and let $\mathcal{F} = \{X_1, \dots, X_s\}$ be a family of subsets of $\{1, 2, \dots, m\}$. We want to approximate the numbers a_i with integers $\tilde{a}_i$, in such a way that $|a_i - \tilde{a}_i| < 1$, and for $X \in \mathcal{F}$ the error on X, defined as $|\sum_{i \in X}(a_i - \tilde{a}_i)|$, remains small.

The version, when the numbers arranged in a rectangular matrix and $\mathcal{F}$ consists of the columns and the rows of the matrix was considered by Baranyai [1]. He showed that for this case there is a solution in which the error on every $X \in \mathcal{F}$ is smaller than 1.

The general problem was investigated by Beck and Fiala [2]. Their result states that if each a_i is covered by at most d elements of $\mathcal{F}$, then it can be achieved that the error is at most $d-1$ on every $X \in \mathcal{F}$.

When each a_i is covered by at most 2 sets ($d=2$), the error bound 1 is tight [2]. For our purposes we need that the error is strictly smaller than one in a situation when $\mathcal{F}$ consists of the blocks of two partitions of $\{1, 2, \dots, m\}$. This is slightly more general than the setting considered by Baranyai.

For simplicity, we assume that the sum $\sum_{i \in X} a_i$ is an integer for every $X \in \mathcal{F}$.

Lemma 2.1. *Let $a_1, \dots, a_m$ be real numbers, and $\bigcup_{i=1}^{k_1} X_{1i} = \bigcup_{i=1}^{k_2} X_{2i} = \{1, \dots, m\}$ be two partitions of $\{1, \dots, m\}$. Assume, that for each of the sets X_{tj}, the numbers $\sum_{i \in X_{tj}} a_i$ are integers. Then there are integers $\tilde{a}_i$, such that*

$$|a_i - \tilde{a}_i| < 1, \quad 1 \le i \le m$$

and

$$\sum_{i \in X_{tj}} a_i = \sum_{i \in X_{tj}} \tilde{a}_i, \quad 1 \le j \le k_t, \ t = 1, 2$$

Proof. Set first $\tilde{a}_i = a_i$. Consider a graph on nodes $\{1, \dots, m\}$. If $\tilde{a}_i$ is an integer, then node i will be isolated. Between the rest of the nodes there will be blue and red edges, corresponding to the two partitions: i and j (when $\tilde{a}_i$ and $\tilde{a}_j$ are not integers) are connected by a blue edge, if there is a block $X_{1\ell}$ of the first partition which contains both i and j. Similarly, there is a red edge between i and j, if there is an $X_{2\ell}$ which contains both. Since the numbers corresponding to a set $X_{t\ell}$ add up to an integer, a non-isolated node is incident to at least one blue and at least one red edge. Hence if not all the nodes are isolated, then there is an alternating blue–red cycle in the graph. Along this cycle the values of the corresponding numbers can be increased and decreased alternately with the same amount ε. This way the set-sums do not change. The value of ε is chosen in such a way, that $|a_i - \tilde{a}_i| < 1$ hold with the modified values of $\tilde{a}_i$, and $\tilde{a}_j$ is an integer for at least one j along the cycle.

We repeat this process until all the nodes become isolated, i.e. the values $\tilde{a}_i$ are integers.

Remark. The proof above is an adaptation of Baranyai's argument. We could have used the Beck–Fiala approach [2] as well.

3. The minimal feasible bandwidth

We have already seen that a bandwidth of at least $m = \sum_i t_i/p_i$ *blocks/time unit* is necessary to have a feasible schedule. The next theorem amounts to stating that this bandwidth is also sufficient, hence optimal.

Theorem 3.1. *Assume that p_i, t_i $(1 \leq i \leq n)$, and $m = \sum t_i/p_i$ are positive integers. The periodical broadcasting problem where message M_i has length t_i and period p_i admits a feasible schedule which sends out m blocks of information in a unit of time.*

Proof. We shall use the lemma. We define a matrix A with n rows and p columns, where p is the least common multiple of the p_i. Row i will correspond to message M_i and column t to the t-th time unit from the start. Let every entry in the i-th row be t_i/p_i. Note that the sum of the elements in a column is equal to m. The sum of the elements in the i-th row is equal to pt_i/p_i.

Now we can apply Lemma 2.1. The numbers are the entries of A. The columns of A give one of the partitions. The other partition is obtained from the rows, dividing the i-th row into the sets $X^{ik} := \{a_{i,kp_i+1}, \dots, a_{i,(k+1)p_i}\}$. The sum of the values in this set is t_i, which, by assumption, is an integer. The lemma guarantees the existence of a matrix $\tilde{A}$ with integer entries, where the set sums are the same as in A. This $\tilde{A}$ provides a preemptive periodical scheduling: in time unit t we broadcast (the next) $\tilde{a}_{i,t}$ blocks from message M_i. This way we send exactly m blocks in every time unit. Moreover, the constraint on X^{ik} forces that we send a complete copy (t_i blocks) of M_i in the interval $[kp_i, (k+1)p_i]$ for $k = 0, 1, \dots, p/p_i - 1$. This finishes the proof.

4. The algorithm

For the periodical scheduling problem Theorem 3.1 gives more than a necessary and sufficient condition. One can also find a feasible schedule based on that proof. For this purpose, one has to consider the possibly huge matrix A, and find a good way of rounding its elements, for example following the proof of the lemma. A disadvantage of this approach is apparent when p turns out to be large (perhaps much larger than the time interval in which we want to use our communication scheme).

Instead of that approach, we show that a simple and fast greedy method produces a good schedule. It will be more convenient to describe this method with the time re-scaled. We select as unit time the amount needed to transmit one block of information. In this new scale the period of M_i (at bandwidth m) will be $q_i := mp_i$.

The *algorithm* is quite simple: at time unit $t = 1, 2, \dots$ we decide which block of information is to be transmitted. For each i there is a current copy of the message M_i. Let B_i be the first block of this copy which has not been sent yet. We shall select for transmission one of the blocks B_i. In order to make the decision, compute for each block B_i how long it can wait: if there are ℓ_i as yet unsent blocks of the current copy of M_i, then B_i can wait $w_i = \lceil t/q_i \rceil q_i - \ell_i - t + 1$ time units and M_i still be finished by the end of its time period. We set $w_i = \infty$ if we are not allowed to send B_i yet (in this case B_i is necessarily the first block of M_i). Now take a block B_i with the smallest $w_i < \infty$, and send this one in time unit t (and update the quantities w_j for the next round).

The following result states that the algorithm provides a schedule at bandwidth m, if there exist a feasible schedule with bandwidth m at all. Together with Theorem 3.1 this implies that the algorithm provides an optimal schedule.

Theorem 4.1. *If there is a feasible schedule, then the preceding algorithm finds one.*

PROOF. A schedule can be described by a sequence $(s_1, s_2, \dots)$ where s_t describes what happens at time interval $[t-1, t]$, so s_t can be either an $i > 0$, with the meaning that the next block of M_i is sent, or 0 when nothing is scheduled for that time interval.

Let $\mathcal{A} = (s_1, s_2, \dots)$ denote the schedule that our algorithm produces and let $\mathcal{S} = (s'_1, s'_2, \dots)$ be a feasible schedule which agrees with $\mathcal{A}$ on the longest prefix. We show, that $\mathcal{S} = \mathcal{A}$. Assume, that $s_i = s'_i$ for $i < t$ and $s_t \neq s'_t$. There are different cases to consider:

$s'_t = 0$, $s_t = j$, i.e. $\mathcal{S}$ schedules nothing and $\mathcal{A}$ schedules M_j for the tth time unit. Then j also appears in schedule $\mathcal{S}$, and its first occurrence after t can be moved to time t: let $\ell > t$ be the smallest index such that

$s'_\ell = j$. In the modified schedule $s'_t = j$ and $s'_\ell = 0$. The schedule obtained in this way from $\mathcal{S}$ is feasible and agrees with $\mathcal{A}$ on a longer prefix than $\mathcal{S}$ did.

$\mathcal{S}$ and $\mathcal{A}$ schedule different blocks for time t, say $s'_t = i$ and $s_t = j$ with $i \neq j$, and $i, j > 0$. Then look for the first j after time t in $\mathcal{S}$; let it be $s'_\ell = j$ ($\ell > t$) and swap the values of s'_t and s'_ℓ. It is immediate at once that the new $\mathcal{S}$ is feasible for all messages M_k with $k \neq i$. To see that M_i is also handled properly, it suffices to verify that $t^* \geq \ell$, where $t^* = \lceil t/q_i \rceil q_i$ is the deadline for the current (at time t) copy of M_i. This is because we send then as many blocks of M_i in $[t-1, t^*]$ as we did before, hence we complete the copy in time. We have on one hand $w_j \geq \ell - t$, because $\mathcal{S}$ is feasible. From the selection rule of the algorithm we infer that $w_j \leq w_i$. By putting these together we obtain that

$$\begin{aligned} t^* &= \lceil t/q_i \rceil q_i = w_i + \ell_i + t - 1 \\ &\geq w_j + \ell_i + t - 1 \geq \ell - t + \ell_i + t - 1 \geq \ell. \end{aligned}$$

At the last inequality we used $\ell_i \geq 1$. This is true because $\mathcal{S}$ is a correct schedule, hence after $t-1$ units of time the current copy of M_i was not transmitted completely. As in the previous case, $\mathcal{A}$ and the modified $\mathcal{S}$ agree up to time unit t.

The situation that $\mathcal{S}$ schedules a block B_i and $\mathcal{A}$ schedules nothing for the interval $[t-1, t]$ cannot happen, for if there is an eligible block, then $\mathcal{A}$ always selects one of them.

References

[1] Zs. Baranyai, On the factorization of the complete uniform hypergraph, in: Infinite and finite sets (A. Hajnal, R. Rado, V. T. Sós, eds.), Colloquia Math. Soc. János Bolyai, vol. 10, *North-Holland*, 1975, 91–107.

[2] J. Beck and T. Fiala, "Integer-making" theorems, *Discrete Applied Mathematics* **3** (1981), 1–8.

[3] V. S. Tanaev, V. S. Gordon and Y. M. Shafransky, Scheduling Theory. Single-Stage Systems, *Kluwer*, 1994.

COMPUTER AND AUTOMATION INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES
H–1111 BUDAPEST, LÁGYMÁNYOSI U. 11
HUNGARY

E-mail: lajos@nyest.ilab.sztaki.hu

(Received December 8, 1997)