Complexity of Minimum Length Scheduling for Precedence Constrained Messages in Distributed Systems

Piera Barcaccia
Dipartimento di Scienze dell'Informazione,
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy.
e-mail: piera@dsi.uniroma1.it

Maurizio A. Bonuccelli
Dipartimento di Informatica,
Università di Pisa,
Corso Italia 40, 56125 Pisa, Italy.
e-mail: bonucce@di.unipi.it

Miriam Di Ianni
Istituto di Elettronica,
Università di Perugia,
Via G. Duranti 1/A, 06123 Perugia.
e-mail: diianni@istel.ing.unipg.it

Complexity of Minimum Length Scheduling for Precedence Constrained Messages in Distributed Systems

Piera Barcaccia, Maurizio A. Bonuccelli, and Miriam Di Ianni

July 14, 1997

Abstract

Switching networks are the core of many communication and multiprocessor systems. In these systems, a set of entities (communication equipment or processors) communicate through the switching network, by exchanging messages. Simultaneous transmission or reception of two or more different messages through an input or output port results in the corruption of the messages (also called collision), which thus are useless and must be retransmitted later. This causes a performance degradation. Collisions can be avoided only by a proper scheduling of the messages. The same problem also arises in single-hop purely optical WDM systems, where simultaneous reception or transmission over the same wavelength channel results in a collision. In this paper, we study the problem of minimum length scheduling of a set of messages subject to precedence constraints. We prove that the decision version of the problem is NP-complete even in very restricted cases. This means that the optimization problem cannot be solved in polynomial time, unless P=NP. Since the problem cannot be optimally solved by fast algorithms, we then investigate the existence of polynomial time approximation algorithms, by first proving that approximation algorithms cannot exist with performance ratio bounded by 4/3 or smaller and successively presenting an ϵ-approximation algorithm with $\epsilon < 2$ for the case of two precedence classes of messages. Finally, we prove the existence of an asymptotically optimal schedule in the general case of an unrestricted number of precedence classes.

Index Terms:

1 Introduction

Switching networks (also called dynamic interconnection networks, or permutation networks) are the core of many communication and multiprocessor systems. For example, SS/TDMA satellites like Intelsat IV, ISDN systems, and shared and distributed memory parallel processing systems, like workstation clusters (e.g. Concurrent's MAXION), are centered around switching networks [24, 11, 8, 33, 16, 13, 37]. Optically connected parallel computers and single-hop communication networks based on passive optical star couplers and tunable transmitters and/or receivers show the same behavior when two or more messages are simultaneously transmitted or received over the same WDM channel, namely when they come from the same input or are bound to the same output ports [18, 1, 25, 31, 26, 30, 34, 6, 15, 35, 36]. In these systems, a set of entities (communication equipment or processors) called *users* communicate through the communication device (switching network or optical star coupler), by exchanging messages. Simultaneous transmission or reception of two or more different messages through an input or output port results in the corruption of the messages (usually called *collision*), which thus are useless and must be retransmitted later. This causes a performance degradation both from a system point of view (the device is used for transmitting messages that must be discarded) and from a users perspective (the messages undergo a longer delay before being successfully received), potentially leading to congestion, deadlock and starvation.

Collisions can be avoided only by a proper scheduling of the messages. However, any collision-free schedule is not sufficient for optimizing the system performance, since, for a given set of messages, several different schedules exist. Thus, it is very important to find a collision-free schedule with minimum length, namely a schedule allowing the transmission of all the messages in the shortest time. This problem has been extensively studied in the past under the name *Time Slot Assignment*, also considering particular side constraints, introduced in the basic problem to take into account specific system organizations, like intersatellite links, and special traffic features, like different priority messages (for example, see [37, 8, 22, 11, 4, 5, 10, 9, 14, 16, 13, 19, 23, 30, 25]). Very recently, the same model has been used to represent and study the scheduling of I/O operations [27].

In this paper, we study the problem of minimum length scheduling of a set of messages subject to precedence constraints. Such a problem has special relevance in multiprocessor system settings, and has not been investigated elsewhere yet. A related problem has been studied in [9], where low priority messages are scheduled in the ports left unused by higher priority ones. We mainly investigate the complexity of the problem, as well as that of approximizing it, and give a lower bound to the schedule length.

In Section 2, we formally state the problem. In Section 3, we prove that the decision version of the problem is NP-complete even in two very restricted cases. This means that the optimization problem cannot be solved in polynomial time, unless P=NP. Since the optimum length of a schedule for k precedence constrained matrices can not be discovered in polynomial time, in Section 4 we estimate such a quantity by proving a purely existential upper bound to the length of an optimal schedule for k precedence constrained matrices. In particular, we consider the general case of k matrices and we prove the existence of a solution whose schedule length is O(lower bound). Since the problem cannot be optimally solved by fast algorithms, we also investigate the existence of polynomial time approximation algorithms (Section 5). We present an ϵ-approximation algorithm with $\epsilon < 2$ for the two-matrices case which can be trivially extended to a k-approximation algorithm for the k matrices and prove that approximation algorithms cannot exist with performance ratio bounded by 4/3 or smaller. This means that the two matrices case problem is in the complexity class APX of problems polynomial-time approximable within a constant error but not in the complexity class PTAS of problems polynomial-time approximable within *any* constant error, while it is still unknown whether the k matrices problem is included in APX. Conclusions and open problems terminate the paper.

2 The problem

In this paper, we consider the packet scheduling paradigm in single hop distributed systems. Messages are fixed length packets that do not tolerate preemption: if the transmission of a packet is interrupted, it must be entirely retransmitted later, thus wasting resources in transmitting useless information. Packetized communication is a standard *de facto*, since it has good efficiency and fairness properties. The system under consideration is a synchronous one, and is made of users that can directly communicate via single hop parallel equipment allowing the simultaneous transmission of several packets. Systems of this kind are those based on switching networks like crossbar switches or Beneš networks, or optical passive stars with tunable receivers and/or transmitters.

Assume there are n input users and n output users. We represent all the packets that must be sent by an $n \times n$ matrix $D = (d_{ij})$, the *traffic matrix*, with non negative integer entries. Entry d_{ij} represents the number of packets that input user i wants to send to output user j[1]. Due to packetized communication, non preemption, and synchronicity the schedule is divided in equal length *time slots* each representing the simultaneous transmission of a set of packets according to a given input-output mapping. When the traffic is represented by a single matrix (the *basic problem*), a *conflict* occurs if an input user sends more than one packet in a time slot or an output user receives more than one packet in a time slot. In this case, the problem of transmitting as fast as possible all the traffic without conflicts (in short, ML-SN SCHEDULING problem) can be formulated as decomposition of the traffic matrix into switching matrices. A *switching matrix* is a matrix with non negative integer entries and with at most one non zero entry in each row and in each column. Such a matrix represents the traffic that can be transmitted without conflicts in more than one time slot, but with the same input-output association. The *length* L_S of a switching matrix S is the value of the largest entry in the matrix and equals the number of time slots needed to transmit the packets depicted in S. A *switching schedule* $\mathbf{S} = (S_1, S_2, \ldots, S_t)$ is a sequence of switching matrices S_i $(i = 1, \ldots, t)$. The *length of the schedule* is the sum of the number of time slots L_{S_i} necessary to transmit every switching matrix S_i, $i = 1, \ldots, t$, i.e. $L(D, \mathbf{S}) = \sum_{i=1}^{h} L_{S_i}$. The ML-SN SCHEDULING problem (finding a schedule of minimum length) can be optimally solved in polynomial time [8, 12]. In the remaining of this paper we call *basic algorithm* such an optimal algorithm.

Given an $n \times n$ matrix D, r_i denotes the sum of the entries in the i-th row $r_i = \sum_{j=1}^{n} d_{ij}$ and c_j denotes the sum of the entries in the j-th column $c_j = \sum_{i=1}^{n} d_{ij}$. The term *line* is used to refer to either a row or a column. A

[1]Of course, such definition can be trivially extended to the more general case of m input and n output users.

line is said *exposed* or *empty* if all entries in that line are zero and *covered* if there is at least one non-zero entry in it. A trivial lower bound on the length of any schedule is $L = \max\{r_i, c_j : i, j = 1, \ldots, n\}$ representing the amount of packets that must be necessarily scheduled sequentially. A line matching the value L is called *critical* .

Sometimes, more complex situations need to be modeled because of precedence constraints that must be met in scheduling the messages, typically arising when we consider distributed memory multiprocessor systems. For instance, in a parallel FFT computation, a processor must wait for some data coming from other processors before processing them, and send the result to other processors. We can think of three kinds of precedences: *dependencies among data* (a message must be received before sending another one), *higher priority of some data* (some data must be sent or received before some other) and *limited resource availability* (necessity to empty a buffer with limited size to make room for new data). In order to model such constraints, we introduce *precedence relations* among the packets. We represent the messages by k different $n \times n$ matrices $D^{(1)}, D^{(2)}, \ldots, D^{(k)}$, such that messages in $D^{(s)}$ must precede messages in $D^{(s+1)}$, as specified in the next definition:

Definition 1 Minimum Precedence Constrained Message Scheduling (MIN PCMS) *Given k $n \times n$ matrices $D^{(1)}, D^{(2)}, \ldots, D^{(k)}$ and a precedence relation $\mathcal{R}$ from elements of $D^{(s)}$ to elements of $D^{(s+1)}$, find a switching schedule $\mathbf{S} = (S_1, S_2, \ldots, S_t)$ for $D^{(1)} + D^{(2)} + \cdots + D^{(k)}$, such that if $d_{ij}^{(s+1)}$ is scheduled in the switching matrix S_q, then $d_{hl}^{(s)}$ must have been scheduled in a switching matrix S_p with $p < q$, for each $i, j, h, l = 1, \ldots, n$, whenever $(d_{ij}, d_{hl}) \in \mathcal{R}$ and such that the length of the schedule is minimized.*

Observe that an obvious lower bound L to the schedule length is obtained by considering the basic problem on $D = D^{(1)} + D^{(2)} + \cdots + D^{(k)}$, namely by assuming that $\mathcal{R}$ is empty. So, $L = \max_{1 \le i,j \le n}\{\sum_{h=1}^{k} r_i^{(h)}, \sum_{h=1}^{k} c_j^{(h)}\}$, where $r_i^{(h)}$ and $c_j^{(h)}$ denote respectively the sum of entries on the i-th row and on the j-th column of matrix $D^{(h)}$.

By properly defining the relation $\mathcal{R}$, all the above introduced precedence relations can be modeled. For instance, if processor i needs to send some data to other processors before sending some message to processor j, then this last message can only be put in a matrix having a higher index than the matrix containing entries that represent the messages to be sent first. In this case, a schedule $\mathbf{S}$ is a sequence of switching matrices $(S_1, \ldots, S_t)$ such that the entries of the i-th row of $D^{(s)}$ can be scheduled in S_r only if the entries of the i-th row of $D^{(s-1)}$ have already been scheduled in $S_1, \ldots, S_{r-1}$. In other words, in this case $\mathcal{R}$ (that we call a *precedence relation on the rows*) contains all pairs (d^p_{ij}, d^q_{il}) such that $p < q$. It is easy to see that the other cases can be defined similarly with simple role interchanges between rows and columns, namely matrix transposition. Without loss of generality, in the following we only consider precedence relations of the kind just defined, that is, scheduling problems with precedence constraints on the rows.The corresponding problem is denoted MIN PCRMS.

3 Intractability result

In this section, we show that the decision problem associated to MIN PCRMS is NP-complete, even under some strong restrictions. This implies that the MIN PCRMS problem cannot be solved in polynomial time (if P $\neq$ NP). Furthermore, the result holds also for all the problems that can be formulated considering the other relations introduced in the previous section.

Before proving such a result, we need a preliminary definition. A *Quasi Doubly Stochastic matrix* QDS(s) is a matrix with integer entries such that each row and each column have line sum s.

The decision version of our problem can be formulated as follows:

Definition 2 Feasible Precedence Constrained on Rows Message Scheduling (PCRMS) *Given k $n \times n$ matrices $D^{(1)}, D^{(2)}, \ldots, D^{(k)}$, a precedence relation on the rows $\mathcal{R}$, and an integer l, does there exist a switching schedule $(S_1, S_2, \ldots, S_t)$ of length at most l for $D^{(1)}, D^{(2)}, \ldots, D^{(k)}$, such that the precedence relation on the rows is met?*

In order to prove the next theorem, we need to introduce the following problem:

Definition 3 (DOUBLE PERFECT MATCHING (DPM)) *Given a bipartite graph $G = (X, Y, E)$ with an even number of vertices, does there exist a perfect matching M_1 for G, such that the graph $G' = (X', Y', E \setminus M_1)$ obtained from G by removing the edges in M_1, contains a perfect matching M_2 ?*

Here X' and Y' denote the sets of non isolated nodes in X and Y (respectively), after the removal of the edges in M_1. Gabow et al. proved in [21] that this problem is NP-complete even if $|X| = |Y|$ and the maximum degree Δ of a node in the graph is three, by reducing the well known NP-complete problem Restricted Time Table Design to it. Since such an NP-completeness result is not easily accessible, and it was given in a rather sketchy way, we report a complete version of it in the appendix. As you can see from the appendix, the DPM problem is NP-complete even if $|X^i| = |Y^i|, i = 1, 2, 3$, where X^i and Y^i are the subsets of, respectively, X and Y containing nodes of degree i.

Theorem 1 *The* PCRMS *problem is NP-complete even under the following restrictions:*

- *only two matrices* $D^{(1)}$ *and* $D^{(2)}$ *must be scheduled;*
- *the entries of matrix* $D^{(1)}$ *are in* $\{0, 1\}$ *and the entries of matrix* $D^{(2)}$ *are in* $\{0, 1, 2\}$*;*
- *the length* l *of the required schedule is* 3.

Proof. This theorem is proved by showing a polynomial time reduction from the DPM problem to the PCRMS one. The graph $G = (X, Y, E)$ of a generic instance of DPM is transformed into a pair of matrices $D^{(1)}$, $D^{(2)}$ such that

$$G \in \text{DPM} \Longleftrightarrow (D^{(1)}, D^{(2)}, 3) \in \text{PCRMS}.$$

The first matrix $D^{(1)}$ corresponds to the bipartite graph G, namely $D^{(1)}$ has a row for each node $x \in X$, and a column for each node $y \in Y$. Furthermore, the entry in the row corresponding to x and in the column corresponding to y is 1 if $(x, y) \in E$, and is 0 otherwise. So, matrix $D^{(1)}$ is an $n \times n$ matrix with entries in $\{0, 1\}$. The second matrix $D^{(2)}$ is built according to the following algorithm.

Algorithm "Build $D^{(2)}$"

1. *Compute the row and column sums of matrix* $D^{(1)}$,

$$\begin{aligned} r_i^{(1)} &= \textstyle\sum_{j=1}^{n} d_{ij}^{(1)} \\ c_j^{(1)} &= \textstyle\sum_{i=1}^{n} d_{ij}^{(1)} \end{aligned}$$

2. *Set the row and column sums of matrix* $D^{(2)}$ *to*

$$\begin{aligned} r_i^{(2)} &= 3 - r_i^{(1)} \\ c_j^{(2)} &= 3 - c_j^{(1)} \end{aligned}$$

3. *initialize all the entries of matrix* $D^{(2)}$ *to 0.*

4. **for** $h = 1$ **to** *the number of lines in* $D^{(2)}$ *with line sum 1* **do**

   $d_{ij}^{(2)} := 1$ *if row* i *and column* j *are the* h*-th row and column with line sum equal to 1.*

5. **for** $h = 1$ **to** *the number of lines in* $D^{(2)}$ *with line sum 2* **do**

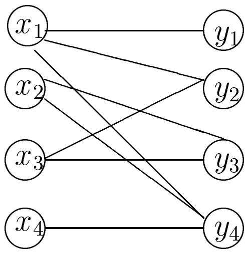
   $d_{ij}^{(2)} := 2$ *if row* i *and column* j *are the* h*-th row and column with line sum equal to 2.*

Observe that $D^{(1)} + D^{(2)}$ is QDS(3), and the possible values for $r_i^{(2)}$ and $c_j^{(2)}$ $(i, j = 1, \ldots, n)$ are 0,1, and 2. Furthermore, as we chose an instance of DPM such that $|X^i| = |Y^i|, \ \ \forall i = 1, 2, 3$, then in $D^{(1)}$, and therefore also in $D^{(2)}$, the number of rows with row sum k is equal to the number of columns with column sum k ($k = 1, 2, 3$ for $D^{(1)}$, $k = 0, 1, 2$ for $D^{(2)}$). The correctness of the algorithm follows from the above considerations. An example of this transformation is given in fig. 1.

We must now prove that the generic instance of DPM has answers yes if and only if the corresponding instance of PCRMS has answers yes, namely:

$$G \in \text{DPM} \Longleftrightarrow (D^{(1)}, D^{(2)}, 3) \in \text{PCRMS}.$$

($\Rightarrow$) Suppose that we have the two perfect matchings M_1 and M_2 in G. It can be easily shown that the edges of G remaining after the deletion of the edges in M_1 and in M_2 form a third matching M_3, as all the nodes with positive degree in the graph induced by the above deletion have degree one. Consider the three matrices S_1, S_2, and S_3 corresponding to the three matchings M_1, M_2, M_3. These matrices are a switching schedule for $D^{(1)}$. We now prove that all the entries of $D^{(2)}$ can be inserted in the free locations (i.e. positions in the matrix that correspond to zero rows and columns) of this schedule, meeting the precedence relations on the rows. There are two possible cases for a non zero entry $d_{ij}^{(2)} \in D^{(2)}$:



$G = (X, Y, E)$

$$\begin{array}{c|c|c|c|c|c} & y_1 & y_2 & y_3 & y_4 & \\ \hline x_1 & 1 & 1 & 0 & 1 & 3 \\ \hline x_2 & 0 & 0 & 1 & 1 & 2 \\ \hline x_3 & 0 & 1 & 1 & 0 & 2 \\ \hline x_4 & 0 & 0 & 0 & 1 & 1 \\ \hline & 1 & 2 & 2 & 3 & \end{array}$$

$D^{(1)}$

$$\begin{array}{|c|c|c|c|c} \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 & 1 \\ \hline 2 & 0 & 0 & 0 & 2 \\ \hline 2 & 1 & 1 & 0 & \end{array}$$

$D^{(2)}$

Figure 1: Example of transformation of the bipartite graph $G = (X, Y, E)$ into matrices $D^{(1)}$ and $D^{(2)}$. Matrix $D^{(1)}$ corresponds to the graph G (rows corresponding to nodes in $X = \{x_1, x_2, x_3, x_4\}$, columns corresponding to nodes in $Y = \{y_1, y_2, y_3, y_4\}$, and non-zero entries corresponding to edges). Matrix $D^{(2)}$ is built according to algorithm "Build $D^{(2)}$". The numbers on the right and on the bottom of the matrices represent, respectively, the row and column sums.

1. $d_{ij}^{(2)} = 1$. In this case row i and column j have line sum 1 in $D^{(2)}$ and therefore line sum 2 in $D^{(1)}$. As the two matchings are both perfect, the two entries in row i and in column j of $D^{(1)}$ must have been scheduled in the first two matrices of the schedule S_1 and S_2. So, in the third matrix S_3, row i and column j are both exposed, and entry $d_{ij}^{(2)}$ can be scheduled in S_3, after the entries of $D^{(1)}$ in the same row.

2. $d_{ij}^{(2)} = 2$. In the same way, we can see that in this case there is only one entry in row i and in column j of $D^{(1)}$ and that this entry must have been scheduled in S_1. So, row i and column j are both exposed in S_2 and S_3, and $d_{ij}^{(2)}$ can be scheduled in these two matrices after the $D^{(1)}$ entry in the same row.

($\Leftarrow$) Assume now that we have a switching schedule (S_1, S_2, S_3) for $D^{(1)}$ and $D^{(2)}$. Since $D^{(1)} + D^{(2)}$ is QDS(3), every switching matrix S_i is a permutation matrix (that is, it has exactly one entry equal to 1 in each line). So, the matchings corresponding to $D^{(1)}$ entries in S_1 and in S_2 must be both perfect. Otherwise, there should be at least one entry in $D^{(1)}$ that was not scheduled when one of its lines was critical (but this situation would lead to a schedule of length at least 4), or some precedence constraint is violated. In both cases, the schedule is not feasible, a contradiction.

Finally, it is easy to see that the above transformation can be performed in polynomial time, and a polynomial time certificate for the problem can be readily obtained, thus assessing that the problem is in the complexity class NP [20]. □

The NP-completeness of the PCRMS problem, even under the restriction that each entry of $D^{(2)}$ can be scheduled only after all the entries in *both* the corresponding row and column of $D^{(1)}$ are scheduled, follows from the above proof. That is, even if we consider a precedence relation *both* on the rows *and* on the columns the problem is NP-complete. So, we established the NP-completeness of all the more general cases presented in the previous section.

Furthermore, the above proof shows that MIN PCMS is NP-complete in the strong sense [20], and thus no pseudopolynomial time optimal algorithm for it exists (unless P=NP).

The above NP-completeness result states that MIN PCMS problem is intractable when the entries of a line in a matrix $D^{(i+1)}$ must be preceded by many (namely, all) entries in the same line of the previous matrix $D^{(i)}$. It is natural to address the complexity of the same problem when few entries of a matrix must precede a given entry of the following matrix. Here we show that the *density* of $\mathcal{R}$ is not important from a complexity viewpoint since the MIN PCMS problem remains NP-complete even if *only one* entry of $D^{(i)}$ must precede a given entry of $D^{(i+1)}$, $i = 1, 2, \ldots, k-1$. Observe that neither the result proved in the next theorem nor the one proved in the previous one can be derived one from the other.

Theorem 2 *The* MIN PCMS *problem remains NP-complete even under the following restrictions:*

- *only two matrices $D^{(1)}$ and $D^{(2)}$ must be scheduled;*
- *the entries of matrix $D^{(1)}$ are in $\{0, 1\}$ and the entries of matrix $D^{(2)}$ are in $\{0, 1, 2\}$;*
- *the length l of the required schedule is 3;*
- *each non zero entry $d^{(2)}_{ij}$ must be preceded by only one $D^{(1)}$ entry, and this last entry is in column j.*

Proof. Let us first notice that the DPM problem is NP-complete even if the following property holds: let y_j be any degree 2 node in the Y set, and let (x_i, y_j), and (x_h, y_j) be the two edges incident to y_j. Then, no degree 2 Y node is adjacent to a degree 2 X node, and so the degree of either x_i or x_h is 3, and the degree of the other X node is 1.

The polynomial time reduction used here is similar to that given in theorem 1, the only difference being the precedence relation $\mathcal{R}'$. Here we have:

$$\begin{aligned}\mathcal{R}' &= \{(d_{ij}^{(1)}, d_{hj}^{(2)}) : d_{ij}^{(1)} > 0, d_{hj}^{(2)} > 0, c_j^{(1)} = 1\} \cup \\ &\quad \{(d_{ij}^{(1)}, d_{hj}^{(2)}) : d_{ij}^{(1)} > 0, d_{hj}^{(2)} > 0, r_i^{(1)} = 3, c_j^{(1)} = 2,\ \exists i' : d_{i'j}^{(1)} > 0, r_{i'}^{(1)} = 1\}.\end{aligned}$$

Again, the proof consists of two parts. The $\Longrightarrow$ part is identical to that given in theorem 1, and will not be repeated here.

($\Longleftarrow$) A schedule $\mathbf{S} = (S_1, S_2, S_3)$ for $D^{(1)}$ and $D^{(2)}$ meeting $\mathcal{R}$ exists, and we must show that the DPM problem has a solution. We can consider two cases (see figure 2):

Case 1 : $d_{hj}^{(2)} = 2$. Thus, $r_h^{(1)} = 1$ and $c_j^{(1)} = 1$. Let $d_{ij}^{(1)} = 1$ and $d_{hj'}^{(1)} = 1$ be the only non-zero entries in column j and row h, respectively, of $D^{(1)}$. Since $\mathbf{S}$ is a 3 slots schedule, then $d_{ij}^{(1)} \in S_1$, $d_{hj'}^{(1)} \in S_1$, and $d_{hj}^{(2)} \in S_2, S_3$. So, $(x_i, y_j) \in M_1$, $(x_h, y_{j'}) \in M_1$, and the degree 1 X nodes are covered in the first matching.

Case 2 : $d_{hj}^{(2)} = 1$. Hence, $r_h^{(1)} = c_j^{(1)} = 2$. According to the NP-completeness proof for DPM problem given in the appendix, there exist i and i' such that $d_{ij}^{(1)} = 1$, $d_{i'j}^{(1)} = 1$, $r_i^{(1)} = 3$ and $r_{i'}^{(1)} = 1$. So, there exists an index j' such that $d_{i'j'}^{(2)} = 2$, and so $d_{i'j}^{(1)} \in S_1$ and $(x_{i'}, y_j) \in M_1$. Besides, $d_{ij}^{(1)} \in S_2$ (so that $(x_i, y_j) \in M_2$) and $d_{hj}^{(2)} \in S_3$, since $(d_{ij}^{(1)}, d_{hj}^{(2)}) \in \mathcal{R}$. Consequently, the non-zero entries in row h of $D^{(1)}$ must be in S_1 and in S_2. It follows that if a $D^{(1)}$ entry is scheduled in S_2, then an entry in the same row or column has been scheduled in S_1. Thus, the $D^{(1)}$ entries are scheduled as early as possible, and M_1 and M_2 are perfect matchings.

□

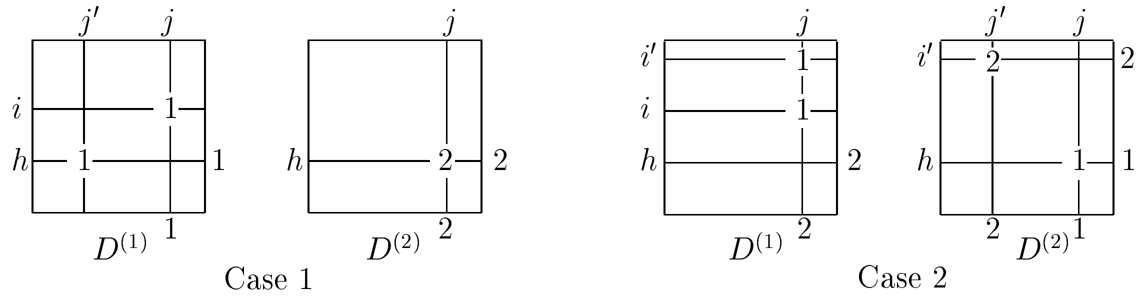


Figure 2: The two cases in the proof of theorem 2.

4 An 0(lower bound) schedule for k matrices

The NP-completeness results proved in the previous section imply that the minimum length of a schedule for k precedence constrained matrices can not be discovered in polynomial time. In other words, this means that when the size of the instance increases it is impossible (modulo P$\neq$NP) to know in polynomial time how many time slots are necessary to complete the schedule. In this section, we estimate such a quantity by proving a purely existential upper bound to the length of an optimal schedule for k precedence constrained matrices. In particular, we prove that a schedule $\mathbf{S}$ for the MIN PCRMS problem exists, such that the length $L(D^{(1)}, D^{(2)}, \ldots, D^{(k)}, \mathbf{S})$ is asymptotically optimal, that is, it matches the lower bound $L(D^{(1)} + D^{(2)} + \ldots + D^{(k)})$ up to constant factors. Let us remark that we do not provide any algorithm able to find such a schedule, but prove its existence, which is not obvious.

This result is obtained by exploiting the same technique used by Leighton, Maggs and Rao in [28]. The strategy is to construct an initial "greedy" schedule $\mathbf{S_0}$, which is as short as possible but where conflicts occur, and to refine it many times until a proper solution is reached. In particular, the schedule $\mathbf{S_{i+1}}$ is obtained from schedule $\mathbf{S_i}$ by "properly" adding delays to packets in order to decrease the number of conflicts. The existence of a proper way of adding delays is ensured by the Lovász Local Lemma [32]. Observe that this lemma in nonconstructive and that is what makes our proof nonconstructive too.

Lemma 1 (Lovász) *Let $A_1, \ldots, A_m$ be a set of "bad" events each occurring with probability p with dependence at most b (that is, every event is mutually* *independent of some set of $m-b-1$ other bad events). If $4pb<1$, then with probability greater than zero, no bad event occurs.*

Before going into details, some definitions are needed. A *T-frame* is a sequence of T consecutive time slots of a given schedule. The *frame congestion C* in a T-frame is the largest number of packets that are simultaneously transmitted or received by the same user during that frame. The *relative congestion ρ* in a T-frame is the ratio C/T of the congestion in the frame to the size of the frame. The relative congestion of a schedule with respect to T-frames is the maximum among the relative congestions in all the T-frames.

Let us now sketch our strategy. Each matrix W_t transmitted in time slot t of the greedy schedule $\mathbf{S_0}$ is such that $s_{ij}=1$ if some $d_{ij}^{(l)}>0$ and all the elements on the same row i of each $D^{(s)}$, with $s<l$, have already been scheduled in some time slot $t'<t$. In all other cases $s_{ij}=0$ (see fig. 3). The length of this initial unfeasible greedy schedule $\mathbf{S_0}$ is not greater than L, where L is the lower bound to the measure of any solution of the MIN PCRMS problem. The relative congestion is $\rho_0=1$ if we consider an I_0-frame, where $I_0=L$ (with possible idle slots added at the end of $\mathbf{S_0}$). Notice that in $\mathbf{S_0}$ the precedence constraints are always satisfied, but the schedule is unfeasible since several conflicts may occur. Indeed, in order to schedule without conflicts the entries of the generic W_t in one time slot, there should be at most one non zero entry (equal to 1) in each line. Since this is not the case, we will refine this initial greedy schedule many times, at each step attempting to reduce the maximum number of conflicts while meeting the precedence constrains.

Let ρ_i be the relative congestion of schedule $\mathbf{S_i}$ with respect to F_i-frames. At step $i+1$ of our refinement procedure, we build schedule $\mathbf{S_{i+1}}$ from schedule $\mathbf{S_i}$. The new refined schedule $\mathbf{S_{i+1}}$ is such that in frames of much smaller size F_{i+1}, the relative congestion ρ_{i+1} is "almost" (in a sense that we will specify later) ρ_i. More precisely, $\mathbf{S_{i+1}}$ is derived from $\mathbf{S_i}$ by assigning a random delay (chosen independently and uniformly in a proper set of values) to each packet. In particular, a packet contained in matrix $D^{(j)}$ is assigned a delay chosen independently and uniformly in the set $\{(j-1)F_i,,\ldots,jF_i-1\}$. If a delay δ_{i+1} is assigned to a packet in $\mathbf{S_{i+1}}$, that packet will be transmitted in time slot $\delta=\sum_{q=0}^{i+1}\delta_q$, where δ_0 is the time slot in which the packet is transmitted in $\mathbf{S_0}$. Notice that the construction of $\mathbf{S_0}$ and the rule for delay assignments always guarantee that the precedence constraints are met.

$$
\underset{D^{(1)}}{\begin{array}{|c|c|c|}\hline 2 & 1 & 0 \\ \hline 0 & 3 & 2 \\ \hline 1 & 1 & 1 \\ \hline \end{array}}
\qquad
\underset{D^{(2)}}{\begin{array}{|c|c|c|}\hline 2 & 1 & 1 \\ \hline 0 & 1 & 2 \\ \hline 2 & 2 & 0 \\ \hline \end{array}}
$$

$$
\underset{W_1}{\begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline 0 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}}
\quad
\underset{W_2}{\begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 1 & 1 & 0 \\ \hline \end{array}}
\quad
\underset{W_3}{\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline 1 & 1 & 0 \\ \hline \end{array}}
\quad
\underset{W_4}{\begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}}
\quad
\underset{W_5}{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}}
$$

Figure 3: An instance $D^{(1)}, D^{(2)}$ of the MIN PCRMS problem and the schedule $\mathbf{S_O} = \{W_1, W_2, W_3, W_4, W_5\}$.

In order to establish our result, it is first necessary to show that the sequence of refinements to the greedy schedule can bring to a schedule $\mathbf{S}^*$ with constant relative congestion in each frame of constant size, and whose length is $O(L)$. From $\mathbf{S}^*$ it is then trivial (for instance, by an exhaustive search) to derive a final schedule $\mathbf{S}$ which does not include any conflict, satisfies the precedence constraints and delivers all the packets in $O(L)$ time slots.

For ease of exposition, in the remaining of this section we address quantities that, in general, are not integers to represent frame sizes, packet delays and the similar. As in [28], we observe that rounding these quantities to integer values (when needed) does not affect the proof correctness.

Lemma 2 *In any schedule, if the relative congestion in any frame of size T is at most ρ, for all T, between $y \leq T \leq 2y-1$, then the relative congestion in any frame of size T is at most ρ for all $T \geq y$.*

Proof. The proof of this lemma is the same as that of lemma 3.5 in [28]. For the sake of clarity, it is repeated here.

Let us consider a T'-frame such that $T' > 2y-1$. The first $(\lfloor T'/y \rfloor - 1)y$ time slots of that frame can be broken into y-frames. In each of these frames, at most ρy packets are simultaneously transmitted or received by the same user. A further frame consisting of the remaining T time slots of the T'-frame must be considered. Since $y \leq T \leq 2y-1$, in that T-frame the relative congestion must be at most ρ. □

Lemma 3 *Let $I_i \geq k$ be an integer value. It is possible to choose the packets delays of refinement $i+1$ so that, if the relative congestion of schedule $\mathbf{S_i}$ is ρ_i in any frame of size I_i/k or greater, then the relative congestion of schedule $\mathbf{S_{i+1}}$ is $\rho_{i+1} = \rho_i(1+\alpha/\sqrt{\log I_i})$ in any frame of size $I_{i+1}/k = (\log^2 I_i)/k$ or greater.*

Proof. The delays are assigned as above, and setting $F_i = I_i/k$.

Recall that we consider the schedule $\mathbf{S_i}$ as a sequence of matrices each to be transmitted in a single time slot. In the following, the term "matrix" is used to refer to such matrices.

Let T be any value between I_{i+1}/k and $2I_{i+1}/k - 1$. An event is a pair (l,t), where l is a line (row or column) of a matrix of the schedule and t is the time slot during which the matrix containing that line is scheduled. A bad event (l,t) occurs when more than $\rho_{i+1}T$ packets on the same line l are simultaneously transmitted or received during time slot t. In order to apply Lovász Local lemma, we must bound the dependence b of a bad event and the probability p that a bad event occurs.

Notice that, since the schedule $\mathbf{S_i}$ is given, and according to the delay assignment rule, a bad event (l,t) may occur only due to packets which are on the same line l and are scheduled in $\mathbf{S_i}$ in some time slot t' with $t - I_i + 1 \leq t' \leq t$.

Thus, two bad events (l,t) and (l',t') are not independent only if some packet belongs to both lines l and l' and if $0 \leq t-t' \leq I_i - 1$. Since at most $\rho_i I_i/k$ packets are conflicting in each time slot in $\mathbf{S_i}$ (before the delays are inserted) and each packet is delayed at most $I_i - 1$ time slots, the maximum number of packets that are conflicting in $\mathbf{S_{i+1}}$ (that is, after the delays are inserted) is bounded by $\rho_i I_i^2$. Each of these packets is contained in exactly one other line, and so a bad event depends only on at most $\rho_i I_i^3$ other bad events. Since $\rho_i = O(1)$, as we shall see in theorem 3, $b = O(I_i^3)$.

Let us now bound p. Consider a *fixed* I_{i+1}/k-frame H and a time slot t contained in H. Let p_1 denote the probability that more than $\rho_{i+1}I_{i+1}/k$ packets on the same line l are simultaneously transmitted in time slot t. Since the maximum delay assigned to a packet in the $(i+1)$-th refinement is $I_i - 1$, any packet of l scheduled in $\mathbf{S_i}$ during time slot t or during the $I_i - 1$ time slots preceding t, may use l in time slot t during the same (I_{i+1}/k)-frame H of $\mathbf{S_{i+1}}$. Hence, at most $\rho_i I_i$ packets on the same line are transmitted in time slot t in $\mathbf{S_{i+1}}$. Since the delay of a packet is chosen in a set of size I_i/k and at most I_{i+1}/k delays allow to transmit the packet during the (I_{i+1}/k)-frame, the probability that one of them is transmitted during the (I_{i+1}/k)-frame is at most I_{i+1}/I_i. Thus, the following inequality holds:

$$p_1 \leq \sum_{h=\rho_{i+1}I_{i+1}/k}^{\rho_i(I_i+I_{i+1}/k)} \binom{\rho_i(I_i + I_{i+1}/k)}{h} \left(\frac{I_{i+1}}{I_i}\right)^h \left(1 - \frac{I_{i+1}}{I_i}\right)^{\rho_i(I_i+I_{i+1}/k)-h}$$

Assume that there are n independent Bernoulli trials, each of which is successful with probability q. Let Q denote the number of successes in the n trials, and $\mu = nq$ is the expectation of Q. Thus, the following inequality [2], holds for $0 \leq \gamma \leq 1$,

$$\Pr[Q \geq (1+\gamma)\mu] \leq e^{-\gamma^2\mu/3}.$$

In our case $n = \rho_i(I_i + I_{i+1}/k)$ and $q = I_{i+1}/I_i$. Thus, $\mu = \rho_i(I_i + I_{i+1}/k)\frac{I_{i+1}}{I_i}$. When $\gamma = \sqrt{\frac{3h_0}{\log I_i}}$, (where h_0 is any positive constant), and if Q denotes the number of bad events, we have:

$$\Pr[Q \geq (1+\gamma)\mu] \leq e^{-\frac{h_0 k \rho_i (I_i+I_{i+1}) I_{i+1}}{k I_i \log I_i}} \leq e^{-\frac{h_0 \rho_i (I_i+I_{i+1}) I_{i+1}}{k I_i \log I_i}} \leq e^{-\frac{h_0 \log I_i}{k}} \leq \frac{1}{I_i^{h_0/k}},$$

after choosing $(1+\gamma)\mu = \rho_{i+1}I_{i+1}$. Thus, with the above choice of γ, $\rho_{i+1} \leq \rho_i\left(1 + \frac{3\sqrt{3h_0}}{\sqrt{\log I_i}}\right)$ when $h_0 > \frac{\log^5 I_i}{3I_i^2}$.

The second step for bounding p, consists in bounding the probability p_2 that more than $\rho_{i+1}I_{i+1}$ packets on the same line are simultaneously transmitted in *any* I_{i+1}-frame containing t. Since there are at most I_{i+1} such frames, $p_2 \leq O(I_i^2)p_1$. Thus, for any $h_1 > 0$, there exists h_0 such that $p_2 \leq 1/I_i^{h_1}$.

The probability p can be bounded by observing that there are at most I_{i+1}/k possible frame sizes between I_{i+1}/k and $2I_{i+1}/k - 1$, and at most $O(I_i^2)$

T-frames containing t, with $I_{i+1} \leq T \leq 2I_{i+1} - 1$. Thus, the probability p that more than $\rho_{i+1}T$ packets are conflicting in a T-frame is at most $p \leq 1/I_i^{h_2}$ for any $h_2 > 0$ depending on h_0.

Since there exist h_2 such that $4pb < 1$, we can finally apply the Lovàsz Local Lemma to establish that there exists a way of assigning the delays such that the relative congestion in any T-frame of $\mathbf{S_{i+1}}$ is at most ρ_{i+1}, with $I_{i+1} \leq T \leq 2I_{i+1} - 1$. By lemma 2, the same bound on the relative congestion holds for any $T \geq I_{i+1}$. □

It is now possible to prove the main result of this section:

Theorem 3 *Given any instance* $(D^{(1)}, D^{(2)}, \ldots, D^{(k)})$ *of the* MIN PCRMS *problem, there always exists a schedule* $\mathbf{S}$ *whose length* $L(D^{(1)}, D^{(2)}, \ldots, D^{(k)}, \mathbf{S}) = O(L)$.

Proof. $\mathbf{S}$ is built by first incrementally refining $\mathbf{S_0}$ till a schedule $\mathbf{S}^* = \mathbf{S_j}$ (having constant relative congestion in frames of size $I_j/k = O(1)$) is reached, and then by solving the conflicts in each frame. The existence of $\mathbf{S_j}$ is due to lemma 3 and to the frame sizes and the relative congestions which are given by the following recurrences:

$$I_{i+1} = \begin{cases} L & \text{if } i = 0 \\ \log^2 I_i & \text{if } i > 0 \end{cases}$$

$$\rho_{i+1} = \begin{cases} O(1) & \text{if } i = 0 \\ \rho_i \left(1 + \frac{O(1)}{\sqrt{I_i}}\right) & \text{if } i > 0 \end{cases}.$$

Such recurrences have solutions $I_j = O(k)$ and $\rho_j = O(1)$, where $j = O(\log^* L)$.

Since a packet is charged with a delay of at most

$$L + \log^2 L + \log^2 \log^2 L + \ldots = O(L),$$

in $\mathbf{S_j}$, the total length of $\mathbf{S_j}$ is $O(L)$.

Let us now show how to construct $\mathbf{S}$ from $\mathbf{S_j}$. I_j/k is constant, and so $\mathbf{S_j}$ contains $O(L)$ disjoint (I_j/k)-frames. Furthermore, each such frame has a constant number of conflicts. Thus, all conflicts are eliminated by scheduling each such frame independently from the others and by increasing the size of every (I_j/k)-frame by a costant factor. Hence, in order to solve all the conflicts, we have increased the length of $\mathbf{S_j}$ by $O(L)$ time slots. The length of the resulting final schedule $\mathbf{S}$ is then $O(L)$. □

5 Approximation

The strong NP-completeness of MIN PCMS implies that it is impossible to find an optimum solution in polynomial (and also in pseudopolynomial) time. Thus, we are left with the possibility of finding fast "good" approximate solutions. We thus search for polynomial time algorithms able to find approximate solutions, that is, solutions whose measures have constant relative error with respect to measures of the optimal ones. Here, the relative error or performance ratio of a minimization problem Π is defined as follows:

$$\frac{m(Sol_A)}{m(Sol^*)}$$

where Sol^* is an optimum solution, Sol_A is the approximate solution found by algorithm A, and $m(Sol^*)$ and $m(Sol_A)$ are, respectively, their measures. A problem is said to be ϵ-approximable if a polynomial time algorithm A exists such that the relative error is never greater than ϵ.

In this section, we confine ourselves to the problem of finding a MIN PCRMS for only two matrices $D^{(1)}$ and $D^{(2)}$, and call this problem the MIN PCRMS-2 problem. The first question that we can answer at this point is whether the problem is in the approximation complexity class PTAS, or not, that is whether the problem can be approximated within *any* relative error or not [7]. We show that the answer to this question is negative.

Theorem 4 *No ϵ-approximation algorithm A can exist for problem* PCRMS *with relative error $\epsilon < 4/3$.*

Proof. From the NP-completeness proof of theorems 1 and 2 we know that

- if two perfect matchings in the graph G exist, then $L(D^{(1)}, D^{(2)}) = 3$; while,

- if two perfect matchings in the graph G do not exist, then $L_A(D^{(1)}, D^{(2)}) \geq 4$.

Assume that A is an ϵ-approximation algorithm. If $L(D^{(1)}, D^{(2)}) = 3$, then we have

$$R_A(D^{(1)}, D^{(2)}) = \frac{L_A(D^{(1)}, D^{(2)})}{L(D^{(1)}, D^{(2)})} = \frac{L_A(D^{(1)}, D^{(2)})}{3} \leq \epsilon \tag{1}$$

and therefore $L_A(D^{(1)}, D^{(2)}) \leq 3\epsilon$. Otherwise, we know that $L_A(D^{(1)}, D^{(2)}) \geq L(D^{(1)}, D^{(2)}) \geq 4$.

From these last two inequalities, we deduce that it cannot be $3\epsilon < 4$. In fact, if it were the case, we could be able to solve in polynomial time the NP-complete problem DPM(e.g. by running the approximation algorithm). □

Since we know that the problem cannot be approximated within *any* relative error, we are now left with the question of whether there exists *some* constant relative error within which the problem is approximable or not. In the following, we present two ϵ-approximation algorithms for the MIN PCRMS-2 problem, the first one with $\epsilon = 2$ and the second one with $\epsilon < 2$. Thus, we can affirmatively answer the previous question, and assert that the problem is in the class APX [7], the class of optimization problems that are approximable within a constant relative error.

Let us consider the following naive algorithm, whose time complexity is $O(n^{4.5})$.

<u>Algorithm $A1$</u>

1. find an optimum ML-SN SCHEDULING [2] for $D^{(1)}$;
2. schedule $D^{(1)}$ accordingly;
3. find an optimum schedule for $D^{(2)}$;
4. schedule $D^{(2)}$ accordingly, and after the schedule for $D^{(1)}$.

[2]The problem of finding a ML-SN-schedule for a single matrix is identical to the basic problem, and thus is optimally solvable in $O(n^{4.5})$ time [8].

We show that this is a 2-approximation algorithm for MIN PCRMS-2.

Theorem 5 *$A1$ is an approximation algorithm for the* MIN PCRMS-*2 problem with relative error*

$$R_{A1} \leq 2.$$

Proof. The algorithm puts the optimum solutions of the basic problem for matrices $D^{(1)}$ and $D^{(2)}$ one after the other. So, we have that $L_{A1}(D^{(1)}, D^{(2)}) = L(D^{(1)}) + L(D^{(2)})$. Since two trivial lower bounds on the optimum solution $L(D^{(1)}, D^{(2)})$ of the MIN PCRMS-2 problem are $L(D^{(1)}, D^{(2)}) \geq L(D^{(1)})$ and $L(D^{(1)}, D^{(2)}) \geq L(D^{(2)})$, we have $L_{A1}(D^{(1)}, D^{(2)}) \leq 2L(D^{(1)}, D^{(2)})$. That is

$$R_{A1}(D^{(1)}, D^{(2)}) = \frac{L_{A1}(D^{(1)}, D^{(2)})}{L(D^{(1)}, D^{(2)})} \leq 2.$$

□

Observe that this bound on the relative error of algorithm $A1$ cannot be improved, since there are instances for which the bound is tight. As an example, consider two matrices $D^{(1)}$ and $D^{(2)}$ such that $L(D^{(1)}) = L(D^{(2)}) = L(D^{(1)} + D^{(2)})$, and $D^{(1)}$ has all the entries on even rows equal to zero, while $D^{(2)}$ has all the entries on odd rows equal to zero. Then, Algorithm $A1$ generates a schedule for $D^{(1)}$ and $D^{(2)}$ with $L_{A1}(D^{(1)}, D^{(2)}) = L(D^{(1)}) + L(D^{(2)}) = 2L(D^{(1)})$. But entries in $D^{(2)}$ are in rows that correspond to exposed rows in $D^{(1)}$, so they don't have to wait before being scheduled, namely $\mathcal{R}$ is empty. Hence, an optimum solution for this MIN PCRMS-2 problem can be found by optimally scheduling $D^{(1)} + D^{(2)}$. The value of the optimum solution is $L(D^{(1)}, D^{(2)}) = L(D^{(1)} + D^{(2)}) = L(D^{(1)})$ and the resulting relative error is 2.

Our second algorithm avoids these situations, by simply anticipating the schedule of some entry in $D^{(2)}$ as soon as possible (that is, when the corresponding row in $D^{(1)}$ becomes empty), but only if the total length of the final schedule does not increase.

Algorithm $A2$

1. Initialize $U = D^{(1)}$, the matrix of unscheduled entries;

2. "check for anticipation": for any empty row i in $D^{(1)}$, check if one of the following cases holds (so that one entry in $D^{(2)}$ can be anticipated):

   (a) there is a critical column j in $D^{(2)}$ that has a non zero entry $d_{ij}^{(2)}$ in row i such that the j-th column is not critical in U;

   (b) row i in $D^{(2)}$ is critical and has one non zero entry $d_{ij}^{(2)}$ such that the j-th column is not critical in U;

   if no entry can be anticipated then go to step 4, else "anticipate";

3. "anticipate": let the new value in the i-th row and in the j-th column in U be

$$u_{ij} := \min\{d_{ij}^{(2)}, L(U) - c_j\}$$

   and in $D^{(2)}$ be

$$d_{ij}^{(2)} := d_{ij}^{(2)} - u_{ij};$$

4. find an optimum schedule $(S_1, S_2, \ldots, S_{L(U)})$ for U such that if U is not QDS, then $S_{L(U)}$ has at least one empty row;

5. schedule U accordingly, until time step t when some row i becomes empty and let $U := U - \sum_{h=1}^{t} S_h$;

6. if $t < L(U)$ then "check for anticipation", else go to step 7;

7. find an optimum schedule for all the entries not already scheduled in $D^{(2)}$;

8. schedule these entries of $D^{(2)}$ accordingly, and after the schedule already constructed.

We claim that this is an ϵ-approximation algorithm for MIN PCRMS-2, with $\epsilon < 2$.

Theorem 6 *A2 is an approximation algorithm for the* MIN PCRMS-*2 problem with relative error*

$$R_{A2} < 2.$$

Proof. We first observe that condition in step 4 can always be satisfied. Indeed, if U is not QDS, there must be at least one row that is not critical, and at least one switching matrix where no entry from that row is scheduled.

If that switching matrix S_h is not the last one of the schedule, we can always permute it with $S_{L(U)}$.

Note that, in general, $L_{A2}(D^{(1)}, D^{(2)}) \le L_{A1}(D^{(1)}, D^{(2)})$, since when algorithm $A2$ anticipates, it gets a solution not worse than the solution given by algorithm $A1$, and when it does not anticipate, it gets the $A1$ solution.

We consider two cases:

1. $L(D^{(1)}) \ne L(D^{(2)})$.
   Without loss of generality, we can assume $L(D^{(1)}) > L(D^{(2)})$. Using the same bounds of theorem 5 proof, we have $L_{A2}(D^{(1)}, D^{(2)}) \le L_{A1}(D^{(1)}, D^{(2)}) = L(D^{(1)}) + L(D^{(2)}) < 2L(D^{(1)}) \le 2L(D^{(1)} + D^{(2)}) \le 2L(D^{(1)}, D^{(2)})$. So, in this case, $L_{A2}(D^{(1)}, D^{(2)}) < 2L(D^{(1)}, D^{(2)})$.

2. $L(D^{(1)}) = L(D^{(2)})$.
   We consider two different subcases:

   (a) $L(D^{(1)}) = L(D^{(2)}) < L(D^{(1)}) + L(D^{(2)})$.
   In this case, $L_{A2}(D^{(1)}, D^{(2)}) \le L(D^{(1)}) + L(D^{(2)}) = 2L(D^{(1)}) < 2L(D^{(1)} + D^{(2)}) \le L(D^{(1)}, D^{(2)})$. So, it results $L_{A2}(D^{(1)}, D^{(2)}) < 2L(D^{(1)}, D^{(2)})$.

   (b) $L(D^{(1)}) = L(D^{(2)}) = L(D^{(1)}) + L(D^{(2)})$.
   The analysis of the previous cases holds for algorithm $A2$ as well as for algorithm $A1$, but in this particular subcase we show that the solutions given by the two algorithms are different. If this were not the case (that is, if we could not anticipate and decrease the length of the final schedule), we would have $L_{A2}(D^{(1)}, D^{(2)}) = L_{A1}(D^{(1)}, D^{(2)}) = L(D^{(1)}) + L(D^{(2)}) = 2L(D^{(1)} + D^{(2)})$, and we could not assert that $R_{A2} < 2$.

   In general, $D^{(2)}$ will have both critical rows and critical columns. For every critical row i, the corresponding row in $D^{(1)}$ must be empty, since otherwise it would be $L(D^{(1)} + D^{(2)}) > L(D^{(2)})$. Furthermore, one of the two conditions at step 2 of algorithm $A2$ must be met and one entry in row i of $D^{(2)}$ is anticipated before the construction of any schedule for $D^{(1)}$. Observe that any of those anticipations does not increase $L(D^{(1)})$. So, we need to show only that one entry is anticipated from every critical line in $D^{(2)}$, so that the lower bound of what is left to be scheduled in

$D^{(2)}$ is decreased. Indeed, if there is no critical column in $D^{(2)}$ or for each critical column, a non zero entry can be anticipated at the beginning of the algorithm, then we are done. Suppose now that there is a critical column j in $D^{(2)}$ from which no entry could be anticipated at the beginning of the algorithm. We know that the corresponding column in $D^{(1)}$ must be empty and that there exist at least a non critical row in $D^{(1)}$. So, the schedule set up in step 4 has at least one empty row i in the last matrix of the schedule. The original schedule performed at step 4, can always be permuted in such a way that the entry in row i and column j of $D^{(2)}$ is non zero. At this point, we are sure that at least one entry can be anticipated, thus lowering $L(D^{(2)})$.

□

Notice that if the lower bounds of $D^{(1)}$, $D^{(2)}$, and $D^{(1)} + D^{(2)}$ are due all to the same critical line, then algorithm $A2$ (and also algorithm $A1$) is optimal. Indeed, in this case algorithm $A2$ behaves like $A1$: it never anticipates. Besides, $L(D^{(1)}+D^{(2)})$ is a lower bound to the optimum solution. So, it results that

$$L_{A2}(D^{(1)}, D^{(2)}) = L_{A1}(D^{(1)}, D^{(2)}) = L(D^{(1)})+L(D^{(2)}) = L(D^{(1)}+D^{(2)}) = L(D^{(1)}, D^{(2)}).$$

6 Conclusions and open problems

In this paper, the complexity of minimum length scheduling problem for precedence constrained messages in single-hop distributed systems has been investigated. We shown that the problem is strong NP-complete even in very restricted cases. Besides, an approximation algorithm with relative error 2 has been presented. Finally, it has been established that no approximation algorithm can exist with relative error smaller that 4/3. An important problem left open is the existence of algorithms with relative error R in the interval

$$\frac{4}{3} \leq R < 2.$$

From a complexity point of view, it is interesting to establish the APX-completeness [7] of the MIN PCRMS-2 problem and the approximability of the general MIN PCMS problem. Concerning this last point, it can be possible to turn the proof of theorem 3 into a constructive one, similarly to what has ben done in [29] for the non constructive result of [28]. Unfortunately, it seems that the techniques used in [29] are hardly applicable to our problem, mainly because the time complexity of the proposed algorithm is a function of the number of packets to be transmitted. A similar algorithm solving the general MIN PCMS problem should thus require pseudopolynomial time.

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan. "Efficient routing and scheduling algorithms for optical networks". *Proc. of 5th ACM/SIAM Symposium on Discrete Algorithms*, pp. 412–423, January 1994.

[2] D. Angluin and L.G. Valiant. "Fast probabilistic algorithms for hamiltonian circuits and matchings". *Journal of Computer and System Sciences*, vol. 18 (2), pp. 155–193, April 1979.

[3] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, New York, 1992.

[4] P. Barcaccia and M.A. Bonuccelli. "Polynomial time optimal algorithms for time slot assignment of variable bandwidth systems". *ACM/IEEE Transactions on Networking*, vol. 2, pp. 247–251, 1994.

[5] A.A. Bertossi, G. Bongiovanni, and M.A. Bonuccelli. "Time slot assignment in SS/TDMA systems with intersatellite links". *IEEE Transactions on Communication*, vol. 35, pp. 602-608, 1987.

[6] M.S. Borella and B. Mukherjee. "Efficient scheduling of nonuniform packet traffic in a WDM/TDM local ligthwave network with arbitrary transceiver tuning latencies". *IEEE J. on Selected Areas on Communication*, vol. 14, pp. 923–934, June 1996.

[7] D.P. Bovet and P. Crescenzi. *Introduction to the Theory of Complexity*. Prentice Hall, New York, 1993.

[8] G. Bongiovanni, D. Coppersmith and C.K. Wong. "An optimal time slot assignment algorithm for a SS/TDMA system with variable number of transponders". *IEEE Transactions on Communication*, vol. 29, pp. 721–726, 1981.

[9] M.A. Bonuccelli, I.S. Gopal, and C.K. Wong. "Incremental time slot assignment in SS/TDMA satellite systems". *IEEE Transactions on Communication*, vol. 39, pp. 1147–1156, 1991.

[10] M.A. Bonuccelli. "A fast time slot assignment algorithm for TDM hierarchical switching systems". *IEEE Transactions on Communication*, vol. 37, pp. 870–874, 1989.

[11] M.A. Bonuccelli. "Traffic scheduling in telecommunication systems and network flow". In *IEEE Variational Inequalities and Network Equilibrium Problems*, Erice, Italy, Plenum Publ. Corp., 9-19, 1994.

[12] G. Bongiovanni, D.T. Tang, and C.K. Wong. "A general multibeam satellite switching algorithm" *IEEE Transactions on Communication*, vol. 29, pp. 1025–1036, 1981.

[13] W.T. Chen, P.R. Sheu and J.H. Yu. "Time slot assignment in TDM multicast switching systems". *IEEE Transactions on Communication*, vol. 42, pp. 149–165, 1994.

[14] S. Chalasani and V. Varma. "Efficient time slot assignment algorithms for SS/TDMA systems with variable bandwidth beams". In *Proc. of IEEE INFOCOM 91*, pp. 658–667, 1991.

[15] H. Choi, H.A. Choi and M. Azizoglu. "Efficient scheduling of transmissions in optical broadcast networks". *IEEE/ACM Transactions on Networking*, vol. 4, pp. 913–920, Dec. 1996.

[16] K.Y. Eng and A.S. Acampora. "Fundamental conditions governing TDM switching assignments in terrestrial and satellite networks". *IEEE Transactions on Communication*, vol. 35, pp. 755–761, 1987.

[17] S. Even, A. Itai and A. Shamir. "On the Complexity of Time Table and Multicommodity Flow Problems", *SIAM Journal on Computing*, vol. 5, pp. 691–703, 1976.

[18] A. Ganz and Y. Gao, “A time-wavelength assignment algorithm for a WDM star network”. *Proc. IEEE INFOCOM'92*, pp. 2144–2150, 1992.

[19] A. Ganz and Y. Gao. “Efficient algorithms for SS/TDMA scheduling”. *IEEE Transactions on Communication*, vol. 8, pp. 1367–1374, 1992.

[20] M.R. Garey and D.S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[21] H. Gabow, T. Nishizeki, O. Kariv, D. Leven and O. Terada. “Algorithms for edge coloring graphs”. Technical Report 41/85, University of Colorado, 1985.

[22] I.S. Gopal, D. Coppersmith and C.K. Wong. “Minimizing packet waiting time in a multibeam satellite system”. *IEEE Transactions on Communication*, vol. 30, pp. 305–316, 1982.

[23] I.S. Gopal and C.K. Wong. “Minimizing the number of switching in a SS/TDMA system”. *IEEE Transactions on Communication*, vol. 33, pp. 497–501, 1985.

[24] K. Hwang. *Advanced Computer Architecture*. McGraw-Hill Book Company, New York, 1993.

[25] N. Huang, C. Wu, and G. Ma. “A time-wavelength scheduling algorithm for interconnected WDM star networks”. *Proc. of IEEE INFOCOM 94* 1994.

[26] F. Jia, B. Mukherjee, and J. Iness. “Scheduling variable-length messages in single-hop multichannel local ligthwave networks”. *IEEE/ACM Transactions on Networking*, vol. 3, pp. 477–489, Aug. 1995.

[27] R. Jain, K. Somalwar, J. Werth and J.C. Browne. “Heuristics for Scheduling I/O Operations”. *IEEE Transactions on Parallel and Distributed Systems*, vol. 8 (3), pp. 310–320, March 1997.

[28] F. T. Leighton, B. M. Maggs and S. B. Rao. "Packet Routing and Job-Shop Scheduling in O(Congestion + Dilation) Steps. *Combinatorica*, vol. 14 (2), pp. 167–186, 1994.

[29] F. T. Leighton, B. M. Maggs and A. Richa. "Fast algorithms for finding O(Congestion + Dilation) packet routing schedules. Technical Report CMU-CS-96-152, School of Computer Science, Carnegie-Mellon University, 1996. To appear in *Combinatorica*.

[30] B. Mukherjee, J. Iness and F. Jia. "Scheduling variable-length messages in a single-hop multichannel local lightwave network". *IEEE/ACM Transactions on Networking*, vol. 3, 1995.

[31] G.R. Pieris and G.H. Sasaki. "Scheduling transmissions in WDM broadcast-and-select networks." *IEEE/ACM Transactions on Networking*, vol. 2, pp. 105–110, Apr. 1994.

[32] J. Spencer. *Ten Lectures on the Probabilistic Method*. SIAM, Philadelphia, PA, 1987.

[33] M. Schwartz. *Telecommunication Networks: Protocols, Modelling and Analysis*. Addison Wesley, Reading, MA, 1987.

[34] G.N. Rouskas and M.H. Ammar. "Analysis and optimization of transmission schedules for single-hop WDM networks". *IEEE/ACM Transactions on Networking*, vol. 3, pp. 211–221, Apr. 1995.

[35] G.N. Rouskas and V. Sivaraman. "On the design of optimal TDM schedules for broadcast WDM networks with arbitrary transceiver tuning latencies". *Proc. IEEE INFOCOM'96*, pp. 1217–1224, 1996.

[36] H.Y. Tyan, C.J. Hou, B. Wang, and C.C. Han. "On supporting time-constrained communications in WDMA-based star-coupled optical networks". *Proc. IEEE Real Time Systems Symposium*. 1996, pp. 175–184.

[37] W.W. Wu. "On the efficacy of traffic assignment in satellite-switched TDMA systems". In *Proc. 4th Conference on Digital Satellite Communication*, Montreal, P.Q., Canada, 1978.

Appendix: The Double Perfect Matching Problem

The Double Perfect Matching problem is the problem of finding a perfect matching in a bipartite graph, such that, after the deletion of the edges of that matching, there exists another perfect matching in the induced graph. Formally,

Definition 4 (Double Perfect Matching (DPM)) *Given a bipartite graph $G = (X, Y, E)$ with an even number of vertices, does there exist a perfect matching M_1 for G, such that the graph $G' = (X', Y', E \setminus M_1)$ obtained from G by removing the edges in M_1, contains a perfect matching M_2 ?*

Here X' and Y' denote the sets of the non isolated nodes in X and Y (respectively), after the removal of the edges in M_1. Gabow et al. proved in [21] that this problem is NP-complete even if $|X| = |Y|$ and the maximum degree Δ of a node in the graph is three. We are now going to recall this proof. It consists of two polynomial time transformations: the first one from the Restricted Time Table (RTT) problem to the Regular Restricted Time Table (RRTT) problem and the second one from the RRTT problem to the DPM problem. Observe that the RTT problem is a well known NP-complete problem (see [17]). We can define the two Time Table problems as follows.

Definition 5 (Restricted Time Table (RTT)) *Given a bipartite graph $G_{RTT} = (T_{RTT}, C_{RTT}, E_{RTT})$ with $\Delta = 3$, such that*

- $\deg(t) \in \{2, 3\}$ *for each node $t \in T_{RTT}$;*
- *if* $\deg(t) = 2$, *then node t has a forbidden color i, $1 \leq i \leq 3$,*

does there exist an edge coloring of G_{RTT} (with colors in the set $\{1, 2, 3\}$) such that no edge incident in a node with forbidden color i ($i = 1, 2, 3$) is colored with i?

Observe that this is in fact a restatement of the classical RTT problem as formulated in [17]. Nodes in T_{RTT} correspond to teachers, nodes in C_{RTT} correspond to classes, edges correspond to teaching requirements, and forbidden colors correspond to forbidden hours.

Definition 6 (Regular Restricted Time Table (RRTT)) *Given a bipartite graph $G_{RRTT} = (T_{RRTT}, C_{RRTT}, E_{RRTT})$ with $\Delta = 3$, such that*

- $\delta(t) \in \{2, 3\}$ *for each node* $t \in T_{RRTT}$,
- *if* $\delta(t) = 2$, *then node* t *has a forbidden color* i, $1 \le i \le 3$,
- *each color is forbidden the same number of times, and*
- $\delta(c) = 3$ *for each node* $c \in C_{RRTT}$,

does there exist an edge coloring of G_{RRTT} (with colors in the set $\{1, 2, 3\}$) such that no edge incident in a node with forbidden color i ($i = 1, 2, 3$) is colored with i?

Theorem 7 *The* RRTT *problem is NP-complete.*

Proof. (See [21]). The NP-complete problem RTT can be reduced to the RRTT problem. Given an instance G_{RTT} of RTT, let us set up an instance $f(G_{RTT})$ of RRTT in the following way.

- Make three copies of G_{RTT}.
- Cyclically permute the forbidden colors in each copy.
- For each node $\overline{c} \in C_{RTT}$ with $\delta(\overline{c}) = 2$, add a new node $\overline{t}$ and three edges joining $\overline{t}$ with the three copies $\overline{c}_1, \overline{c}_2, \overline{c}_3$ of $\overline{c}$.
- For each node $c \in C_{RTT}$ with $\delta(c) = 1$, identify all its three copies in a single node.

It is easy to see that this new graph $f(G_{RTT})$ is in fact an instance of the RRTT problem. Furthermore[3],

$$G_{RTT} \in \text{YES}_{RTT} \Leftrightarrow G_{RRTT} \in \text{YES}_{RRTT}$$

($\Rightarrow$) Assume a that a coloring for an instance G_{RTT} of RTT exists. Then, we can obtain a coloring for the transformed instance as follows. Permute the colors of the edges in the three copies of G, according to the permutation

[3] In the following, YES_A denotes the set of YES instances of a decisional problem A.

of the forbidden colors. If the three new edges incident in node $\overline{t}$, use all the three colors, the edge $(\overline{t}, \overline{c}_i)$ is colored with the color not already used by the other two edges incident in $\overline{c}_i$ $(i = 1, 2, 3)$.

($\Leftarrow$) Assume now that a coloring of $f(G_{RTT})$ exists. A coloring for G_{RTT} is given by the coloring of the subgraph of $f(G_{RTT})$ corresponding to the copy of G_{RTT} with the original forbidden colors.

□

Theorem 8 *The* DPM *problem is NP-complete.*

Proof. (See [21]). The NP-complete problem RRTT can be reduced to the DPM problem. Given an instance G_{RRTT} of RRTT, let us build an instance $G = (T, C, E)$ of DPM in the following way. Let $t \in T_{RRTT}$ be a node in G_{RRTT} with forbidden color i.

- If $i = 2$, then add node t_2 to T, node c_1 to C, and the two edges (t, c_1) and (t_2, c_1) to E.
- If $i = 3$, then add node c_3 to C and the edge (t, c_3) to E.

This new graph $G = (T, C, E)$ is an instance of the DPM problem. Furthermore, $|T| = |C|$. Let us now show that

$$G_{RRTT} \in \text{YES}_{RRTT} \Leftrightarrow G \in \text{YES}_{DPM}.$$

($\Rightarrow$) Assume a that a coloring for an instance G_{RRTT} of RRTT exists. Then, the matching M_1 in G corresponds to the edges in G_{RRTT} colored with color 3 and all the new edges (t_2, c_1) and (t, c_3). The matching M_2 is formed by the edges in G_{RRTT} colored with color 2 and all the new edges (t, c_1).

Let us show that M_1 is a perfect matching in G, as every node in G is incident to exactly one edge of M_1. Only the t_2-"type" nodes have degree one in T, and the (t_2, c_1)-"type" edges are in M_1. Besides, only the c_3-"type" nodes have degree one in C, and the (t, c_3)-"type" edges are in M_1. Only the nodes with forbidden color 1 in G_{RRTT} have degree two in T, and one of their edges must be colored with color 3. Only the c_1-"type" nodes have degree two in C, and the (t_2, c_1)-"type" edges are in M_1. Degree three nodes in T

are the nodes having degree three in G_{RRTT} (such nodes must have one of their edges colored with color 3), the nodes with forbidden color 2 in G_{RRTT} (they must have one of their edges colored with color 3), and nodes with forbidden color 3 in G_{RRTT} (for which the edge (t, c_3) is in M_1).

M_2 is a perfect matching in the new graph G' induced by the removal of the edges in M_1. All the nodes having degree two or three in G (and having degree one or two in G') are incident to an edge of M_2. In fact, degree two nodes of T are the nodes with forbidden color 1 in T_{RRTT}. Thus one of their edges must be colored with color 2. C nodes with degree two are only of c_1-"type". So, the (t, c_1)-"type" edges are in M_2. Nodes in T having degree three are the nodes with degree three in T_{RRTT} (they must have one of their edges colored with color 2), the nodes in T_{RRTT} with forbidden color 2 (that have their edge (t, c_1) in M_2), and finally the nodes in T_{RRTT} with forbidden color 3 (that must have one of their edges colored with color 2). C nodes with degree three are only the nodes that have degree three also in C_{RRTT}, so they surely have one of their edges colored with color 2.

($\Leftarrow$) Assume now that the two matchings M_1 and M_2 exist. Then, a coloring for G_{RRTT} is obtained by coloring all the edges in $E \setminus (M_1 \cup M_2)$ with color 1, all the edges in $M_2 \cap E_{RRTT}$ with color 2, and all the edges in $M_1 \cap E_{RRTT}$ with color 3. □

From the proof of this theorem, we deduce that the DPM problem is NP-complete even if the bipartite graph $G = (X, Y, E)$ has maximum degree three and $|X| = |Y|$. Furthermore, we now prove that the following corollary holds.

Corollary 1 *The* DPM *problem is NP-complete even if*

$$|X^i| = |Y^i|, \quad \forall i = 1, 2, 3,$$

where $X^i = \{x \in X : \delta(x) = i\}$ *and* $Y^i = \{y \in Y : \delta(y) = i\}$.

Proof. This corollary follows from simple considerations on the previous NP-completeness proof. The bipartite graph G, obtained by transforming the RRTT problem as in the proof of the theorem 8, has the following properties.

a) $|X^1| = |Y^1|$.

Degree one nodes in X are only of "t_2–type" (we introduced one of these nodes for each node in T with forbidden color 2). Degree one nodes in Y are only of "c_3–type" (we introduced one of these nodes for each node in T with forbidden color 3). Since each color in T is forbidden the same number of times, we have $|X^1| = |Y^1|$.

b) $|X^2| = |Y^2|$.
Degree two nodes in X are only the nodes with forbidden color 2. Y nodes with degree two are only of "c_1–type" (we introduced one of these nodes for each node in T with forbidden color 2). Each color in T is forbidden the same number of times, so we have $|X^2| = |Y^2|$.

c) $|X^3| = |Y^3|$.
Follows from a) and b) and from the fact that $|X| = |Y|$.

□