



A new Approach for Minimizing Buffer Capacities with Throughput Constraint for Embedded System Design.

Mohamed Benazouz, Olivier Marchetti, Alix Munier-Kordon, Pascal Urard

► To cite this version:

Mohamed Benazouz, Olivier Marchetti, Alix Munier-Kordon, Pascal Urard. A new Approach for Minimizing Buffer Capacities with Throughput Constraint for Embedded System Design.. AICCSA IEEE/ACS International Conference on Computer Systems and Applications, May 2010, Hammamet, Tunisia. pp.1-8, 10.1109/AICCSA.2010.5586972 . hal-00368648

HAL Id: hal-00368648

<https://hal.science/hal-00368648>

Submitted on 17 Mar 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A new Approach for Minimizing Buffer Capacities with Throughput Constraint for Embedded System Design

Mohamed Benazouz, Olivier Marchetti, Alix Munier-Kordon and Pascal Urard [†]

LIP6, Université Pierre et Marie Curie, Paris (France)

[†] Central R&D – STMicroelectronics, Crolles (France)

Email: {Mohamed.Benazouz, Olivier.Marchetti, Alix.Munier}@lip6.fr, Pascal.Urard@st.com

Abstract—The design of streaming (e.g. multimedia or network packet processing) applications must consider several optimizations such as the minimization of the whole surface of the memory needed on a Chip. The minimum throughput of the output is usually fixed. In this paper, we present an original methodology to solve this problem.

The application is modelled using a Marked Timed Weighted Event Graphs (in short MTWEG), which is a subclass of Petri nets. Transitions correspond to specific treatments and the places model buffers for data transfers. It is assumed that transitions are periodically fired with a fixed throughput.

The problem is first mathematically modelled using an Integer Linear Program. We then study for a unique buffer the optimum throughput according to the capacity. A first polynomial simple algorithm computing the minimum surface for a fixed throughput is derived when there is no circuit in the initial MTWEG, which corresponds to a wide class of applications. We prove in this case that the capacities of every buffer may be optimized independently.

For general MTWEG, the problem is NP-Hard and an original polynomial 2-approximation algorithm is presented. For practical applications, the solution computed is very close to the optimum.

Index Terms—Timed Weighted Event Graphs, Synchronous Dataflow, Buffer minimization, Streaming applications.

I. INTRODUCTION AND RELATED WORK

Due to consumers expectations, embedded systems are becoming increasingly complex. For instance, many of mobile phones available on the market can take and display photos, download and play multimedia contents, and naturally allow to hold a telephone conversation. Most of these applications consist in data stream processing and can be splitted into a set of components or tasks performing specific treatments infinitely often and a set of buffers for data exchanges. Currently, the synchronous dataflow paradigm [1] remains the most widely used in this specific area. In this model, an application is modelled by a directed graph where each node (*resp.* arc), called actor, models a component (*resp.* a buffer). In this producer/consumer paradigm, each actor activation requires to consume data in its input buffers. After a deterministic time, the actor will write data in its output buffers. In the Synchronous DataFlow model (in short SDF), the actors production/consumption rates are known at compile time. As the whole application has to be integrated on a

single chip and satisfies high quality requirements, the buffer minimization problem with throughput constraints is crucial for the design of embedded system.

Nevertheless, in this paper, we consider Marked Timed Weighted Event Graph (in short MTWEG) which is a subclass of Petri net. In this model, transitions correspond to treatments of fixed processing times. Each place p models a buffer and has exactly one input arc and one output arc weighted respectively by $u(p)$ and $v(p)$. These values denote the number of tokens that has to be added to (*resp.* removed from) place p . If $u(p) = v(p) = 1$ for every place, we get a Marked Time (non Weighted) Event Graph (in short MTEG).

MTWEG and SDF are clearly equivalent formalisms. However, the Petri net model has been considerably expanded with many theoretical results from various scientific communities, providing a unified model with many results and algorithmic tools.

The determination of the liveness and the computation of the optimal throughput of a MTEG are two fundamental questions which are polynomially solved from a long time [2], [3], [4]. The minimization of a weighted sum of the initial markings for a minimum given throughput of a MTEG is in *NP*, and many authors developed efficient heuristics and exact methods to solve it (*see.* as example [5], [6], [7]). The *NP*-completeness of this problem was proved recently in [8].

The existence of a polynomial algorithm for the liveness and the computation of the throughput of a MTWEG is a difficult question. Up to now, the time complexity of all the algorithms developed to answer these two fundamental questions is exponential in the worst case [9], [10]. The consequence is that the optimization problems on MTWEG are possibly not in *NP*: the evaluation of the feasible solutions is not possible in polynomial time, which limits dramatically the existence of efficient algorithms. For example, Sauer [11] developed an algorithm to minimize the sum of the initial markings for a given throughput which evaluate a feasible solution using an exponential algorithm. The evaluation step of this algorithm limits significantly the size of the instances.

Another way to circumvent this problem is to reduce the set of feasible solutions. Benabid *et al.* [12] developed a polynomial time algorithm for the computation of a periodic firing of the transitions. This result can be regarded as a

generalization of Reiter's result for MTEG [13]. In the case of MTWEG, the existence of a periodic firing of the transitions is clearly more restrictive than the liveness. On the same way, the throughput of a periodic firing is obtained by firing the transitions as soon as possible. However, optimization problems, such as the minimization of the initial markings are now in NP and efficient algorithms may be developed (even if the problem is NP -complete).

The minimization of the buffer capacities under a throughput constraint was also investigated by several authors using SDF model. The main drawback of all these approaches is that their complexity times depend on the numerical values of the instance, and leads, in the worst case, to exponential-time algorithms. Adé *et al.* [14] and Murthy [15] have both characterized the lowest capacity of a given buffer. Murthy has also showed that the minimization problem of buffer capacity of dataflow graph with a given initial buffers state is NP-Complete. To cope with this problem, Adé *et al.* [16], [17] devise interesting heuristics to minimize the buffers sizes of a SDF graph such that there exists a valid schedule. In [18], [19], several buffer minimization problems with throughput constraint are modelled using an Integer Linear Program. However, the number of constraints relies on the numerical values of consumption/production rates of the instance. More recently, in [20], [21] authors have dealt with this problem with throughput constraint based on a state space exploration with model checking techniques. Lastly, Wiggers *et al.* [22] developed a heuristic that aims to build a periodic firings of the transitions that minimizes the buffer capacities of a SDF.

The aim of this paper is to develop simple and efficient algorithms to solve the minimization of the overall number of initial tokens in a Timed Weighted Event Graph for a periodic firing with a given period. Section 2 is dedicated to basic definitions and the description of our problem. In Section 3, we show the modelling of a car radio using a MTWEG. In Section 4, we show that our problem can be formulated using an Integer Linear Program. The single buffer case is fully studied in Section 5 and a polynomial time algorithm is derived for an important sub-case of MTWEG. Section 6 is devoted to the presentation of a simple 2-approximation algorithm. Section 7 presents the resolution of the example using our algorithm. Section 8 is our conclusion.

II. MODEL AND NOTATIONS

A Marked Timed Weighted Event Graph $\mathcal{G} = (T, P, l, M_0)$ is defined by a set of places $P = \{p_1, \dots, p_m\}$ and a set of transitions $T = \{t_1, \dots, t_n\}$. Every place $p \in P$ is defined between two transitions t_i and t_j and is denoted by $p = (t_i, t_j)$. Each place $p \in P$ is initially marked by $M_0(p) \in \mathbb{N}$ tokens and is associated with two strictly positive integers $u(p)$ and $v(p)$ called the marking functions (*see* figure 1).

For any transition $t_i \in T$, we set $\mathcal{P}^+(t_i) = \{p = (t_i, t_j) \in P, t_j \in T\}$ and $\mathcal{P}^-(t_i) = \{p = (t_j, t_i) \in P, t_j \in T\}$.

It is assumed that two successive firings of the same transition cannot overlap: this is modeled by a self-loop place $p = (t_i, t_i)$, $\forall t_i \in T$ with $u(p) = v(p) = 1$ and $M_0(p) = 1$. For a sake of simplicity, these loops are not pictured.

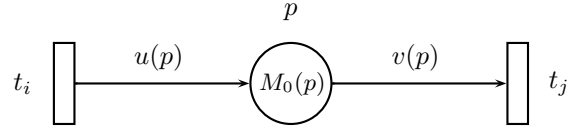


Fig. 1. A place $p = (t_i, t_j)$ of a MTWEG.

We also suppose that a processing time $\ell(t_i)$ is associated to every transition t_i . If t_i is fired at time τ , $v(p)$ tokens are removed from every place $p \in \mathcal{P}^-(t_i)$. At time $\tau + \ell(t_i)$, $u(p)$ tokens are added to every place $p \in \mathcal{P}^+(t_i)$. The instantaneous marking of a place $p \in P$ at time $\tau \geq 0$ is denoted by $M(\tau, p)$. Clearly, $M(0, p) = M_0(p)$.

A place $p = (t_i, t_j)$ has a bounded capacity $F(p) > 0$ if the number of tokens stored in p can not exceed $F(p)$: $\forall \tau \geq 0, M(\tau, p) \leq F(p)$. A MTWEG $\mathcal{G} = (T, P, M_0, l, F)$ is said to be a bounded capacity graph if the capacity of every place $p \in P$ is bounded by $F(p)$. It is proved in [23] that every place $p = (t_i, t_j)$ with bounded capacity may be replaced by a couple of places $(p_1 = (t_i, t_j), p_2 = (t_j, t_i))$ denoted by $(p_1, p_2)_c$ with the initial marking $M_0(p_1) = M_0(p)$ and $M_0(p_2) = F(p) - M_0(p)$. So, in this paper, we only consider symmetric MTWEG: every place $p = (t_i, t_j)$ is associated with a backward place $p' = (t_j, t_i)$ modelling the limited capacity. Note that a symmetric MTWEG is strongly connected: for every couple of vertices $(x, y) \in (P \cup T)^2$, there exists a path in \mathcal{G} from x to y . In addition, to every couple of places $(p, p')_c$ modelling a buffer is associated a non negative cost by unit of capacity denoted by $\theta(p) = \theta(p')$.

For any couple of integers $(a, b) \in \mathbb{N}^2$, $\gcd(a, b)$ denotes the greatest common divisor of a and b .

A. Schedules

Let \mathcal{G} be a MTWEG. A schedule is a function $s : T \times \mathbb{N}^* \rightarrow \mathbb{Q}^+$ which associates, with any tuple $(t_i, q) \in T \times \mathbb{N}^*$, the starting time of the q th firing of t_i . There is a strong relationship between a schedule and the corresponding instantaneous marking. Indeed, a schedule is feasible if the number of tokens of every place $p = (t_i, t_j)$ remains non negative at each time instant.

It has been proved in [24] that the initial marking $M_0(p)$ of any place $p = (t_i, t_j)$ may be replaced by $\left\lfloor \frac{M_0(p)}{\gcd(u(p), v(p))} \right\rfloor \gcd(u(p), v(p))$ without any influence on s . Thus, we assume that the initial marking $M_0(p)$ of every place $p = (t_i, t_j) \in P$ is a multiple of $\gcd(u(p), v(p))$.

The throughput of a transition t_i for a schedule s is defined as

$$\lambda^s(t_i) = \lim_{q \rightarrow \infty} \frac{q}{s(t_i, q)}.$$

A schedule s is periodic if there exists a vector $w = (w_1, \dots, w_n) \in \mathbb{Q}^{+n}$ such that, for any couple $(t_i, q) \in T \times \mathbb{N}^*$, $s(t_i, q) = s(t_i, 1) + (q - 1)w_i$. w_i is then the period of the transition t_i and $\lambda^s(t_i) = \frac{1}{w_i}$ its throughput.

III. EXAMPLE

A. Description

Let us consider a car-radio application described in [25]. The inputs of such systems are basically a MP3-reader and a cell phone. The output is a mixed sound from these two streams. Without any additional treatment, the output is reintroduced in the system through the cell phone, causing an echo effect. In order to obtain a pure speech in the cell phone, an additional input stream, corresponding to a microphone is added.

Figure 2 presents the streams and the main treatments. The first stream entrance, modelled by t_7 is the MP3 reader. t_{10} corresponds to the entrance of the additional microphone. t_9 is the output. t_3 is the audio echo cancellation task. t_1 mixes the two input streams. t_5 produces a pure speech from the streams t_3 and the cell phone.

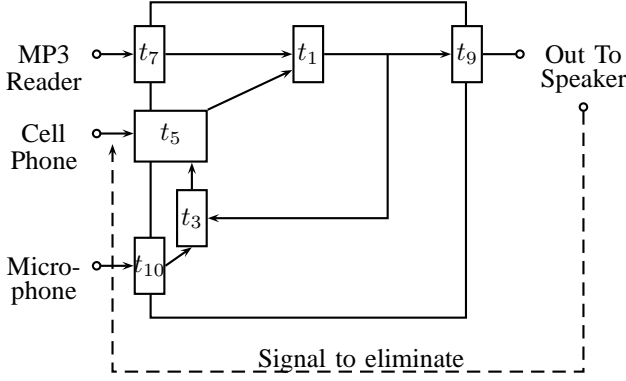


Fig. 2. Block diagram of a car-radio application

Figure 3 shows the modelling of the whole application by a MTWEG \mathcal{G} . Transitions t_2 , t_4 , t_6 and t_8 are simple rate converters. Places model intermediate buffers of limited capacity between the components.

The cost in such signal processing application may be interpreted by the size of samples exchanged between processes. Since in our case they exchange samples of the same size, we set $\theta(p) = 1$ for every places $p \in P$.

B. The normalization step

A MTWEG is said to be normalized if all adjacent marking functions of every transition $t_i \in T$ are equal to a single value denoted by Z_i , i.e. $\forall p \in \mathcal{P}^+(t_i), u(p) = Z_i$ and $\forall p \in \mathcal{P}^-(t_i), v(p) = Z_i$. Note that the number of tokens remains constant in every circuit of a normalized graph. This simple important property eases the study of the liveness and the computation of periodic schedules (see. as example Theorem 1 below). We briefly recall here how every live symmetric MTWEG may be transformed into a normalized graph.

Let us first define the weight (or gain [26]) of every path μ of a MTWEG \mathcal{G} by

$$W(\mu) = \prod_{p \in P \cap \mu} \frac{u(p)}{v(p)}.$$

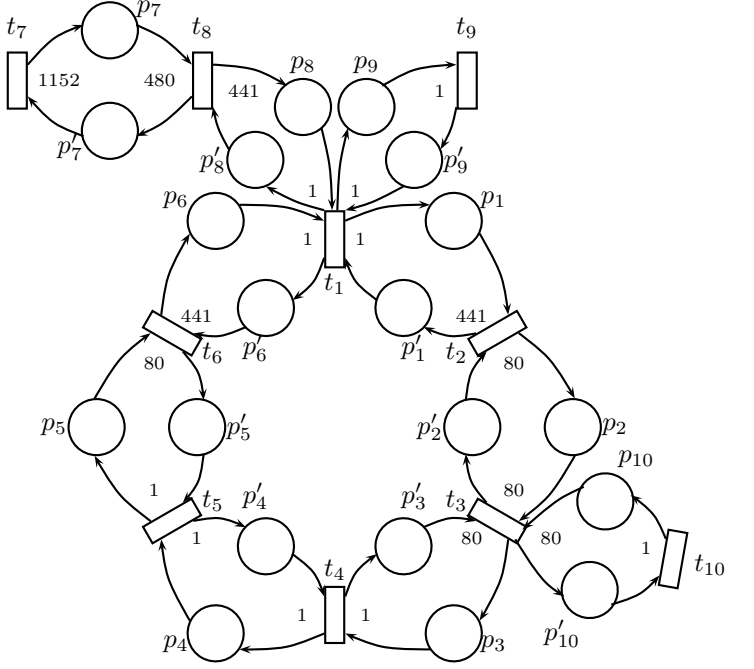


Fig. 3. A MTWEG \mathcal{G} modelling a car-radio application

Roughly speaking, for any circuit c of a MTWEG, $W(c)$ can be viewed as the production rate of tokens on c . So, if $W(c) < 1$ for a given circuit, the number of tokens on this circuit decreases after a finite firing sequence and therefore it leads to a deadlock situation [9], [10].

In the sequel, a MTWEG \mathcal{G} is said to be unitary if every circuit of \mathcal{G} has a weight exactly equal to 1.

Every symmetric MTWEG \mathcal{G} considered in our study is unitary: if it is not, there exists at least one circuit c of \mathcal{G} such that $W(c) > 1$. So, the reverse circuit c' of c verifies $W(c') = \frac{1}{W(c)} < 1$ and thus, \mathcal{G} is not live and no feasible schedule exists.

It is stated in [24] that any unitary MTWEG can be polynomially transformed into an equivalent normalized MTWEG, i.e. with the same feasible schedules. It is firstly proved that, for any value $\alpha \in \mathbb{Q}^{+*}$, the markings functions and the initial markings of any place $p \in P$ may be replaced simultaneously by respectively $\alpha u(p)$, $\alpha v(p)$ and $\alpha M_0(p)$ without any influence on the schedules. Then, it is shown that positive integer values $\alpha(p)$, $p \in P$ such that, $\forall t_i \in T$, there exists an integer Z_i with, $\forall p \in \mathcal{P}^+(t_i), \alpha(p)u(p) = Z_i$ and $\forall p \in \mathcal{P}^-(t_i), \alpha(p)v(p) = Z_i$ may be computed in polynomial time. Values Z_i , $t_i \in T$ replace marking functions as said before.

The MTWEG depicted by Figure 3 is clearly not normalized. Moreover, since it is symmetric, we may have, for every couple of places $(p_i, p'_i)_c$ with $i \in \{1, \dots, 10\}$, $\alpha(p_i) = \alpha(p'_i)$. For our example, this yields to define the following system of equations:

$$\begin{cases} Z_1 = \alpha(p_1) = \alpha(p_6) = \alpha(p_8) = \alpha(p_9) \\ Z_2 = 441\alpha(p_1) = 80\alpha(p_2) \\ Z_3 = 80\alpha(p_2) = 80\alpha(p_3) = 80\alpha(p_{10}) \\ Z_4 = \alpha(p_3) = \alpha(p_4) \\ Z_5 = \alpha(p_4) = \alpha(p_5) \\ Z_6 = 80\alpha(p_5) = 441\alpha(p_6) \\ Z_7 = 1152\alpha(p_7) \\ Z_8 = 480\alpha(p_7) = 441\alpha(p_8) \\ Z_9 = \alpha(p_9) \\ Z_{10} = \alpha(p_{10}) \end{cases}$$

The minimal solution is given by $(\alpha(p_1), \dots, \alpha(p_{10})) = (80 \ 441 \ 441 \ 441 \ 441 \ 80 \ 73.5 \ 80 \ 80 \ 441)$ and corresponds to the new marking functions $Z = (80 \ 35280 \ 35280 \ 441 \ 441 \ 35280 \ 84672 \ 35280 \ 80 \ 441)$.

Remark 1. Normalization affects the cost of places. Indeed, for every couple of places $(p, p')_c$ in the initial graph, initial markings were multiplied by $\alpha(p)$ during the normalization step. So, to get the real overall capacity cost for this new normalized graph we have to divide the cost $\theta(p)$ by $\alpha(p)$.

For our example, we get $(\theta(p_1), \dots, \theta(p_{10})) = \left(\frac{1}{80} \ \frac{1}{441} \ \frac{1}{441} \ \frac{1}{441} \ \frac{1}{441} \ \frac{1}{80} \ \frac{1}{73.5} \ \frac{1}{80} \ \frac{1}{80} \ \frac{1}{441} \right)$.

C. Evaluation of the processing times

For our example, the throughput for the output transition t_9 must be equal to 44.1kHz that is to say $\ell(t_9) = w_9 = \frac{1}{44.1}ms$. The following theorem, proved by [12], will provide an upper bound on the processing times of the other transitions.

Theorem 1. Let \mathcal{G} be a normalized MTWEG. For any feasible periodic schedule s of \mathcal{G} , there exists $K \in \mathbb{Q}^{*+}$ such that, for any couple of transitions $(t_i, t_j) \in T^2$, $\frac{w_i}{Z_i} = \frac{w_j}{Z_j} = K$. Moreover, s is feasible iff, for any place $p = (t_i, t_j) \in P$,

$$s(t_j, 1) - s(t_i, 1) \geq \ell(t_i) + K(Z_j - M_0(p) - gcd_{i,j}).$$

where $gcd_{i,j} = gcd(Z_i, Z_j)$.

By Theorem 1, we derive that $K = \frac{w_9}{Z_9} = 2.83 \times 10^{-4}ms$. The processing time of transition t_3 is fixed for physical considerations to $\ell(t_3) = 9.091ms$. For the other transitions, the processing time must be at most equal to w_i , thus we set $\ell(t_i) = w_i$. These values are reported in Table I.

IV. FORMULATION USING AN INTEGER LINEAR PROGRAM

Let \mathcal{G} be a normalized symmetric MTWEG and $K \in \mathbb{Q}^+$ a fixed value for the period. As seen before, a couple of places $(p, p')_c$ models a buffer of (unknown) minimum capacity $F(p, p') = M_0(p) + M_0(p')$. Moreover, data stored in a buffer have all the same size depending on the buffer and denoted by $\theta(p)$. The general problem considered is to find an initial marking $M_0(p)$, $p \in P$ such that:

- 1) The weighted capacity, proportional to the whole surface of the buffers $\sum_{p \in P} \theta(p)F(p, p') = \sum_{p \in P} \theta(p)M_0(p)$ is minimum.

- 2) There exists a periodic schedule with a period at most equal to K .

The problem may be formulated by the following Integer Linear Program $\Pi(K)$:

$$\begin{aligned} \min \left(\sum_{p \in P} \theta(p)M_0(p) \right) \quad \text{subject to} \\ \begin{cases} \forall p = (t_i, t_j) \in P, & s(t_j, 1) - s(t_i, 1) \geq \ell(t_i) + \\ & K(Z_j - M_0(p) - gcd_{i,j}) \\ \forall p = (t_i, t_j) \in P, & M_0(p) = k_{i,j} \cdot gcd_{i,j} \\ \forall p = (t_i, t_j) \in P & k_{i,j} \in \mathbb{N} \\ \forall t_i \in T, & s(t_i, 1) \geq 0 \end{cases} \end{aligned}$$

The first inequality expresses the necessary and sufficient condition associated with a place p on the first starting times of a feasible periodic schedule following Theorem 1. The second equality comes from the restriction of $M_0(p)$, $p = (t_i, t_j) \in P$ to multiples of $gcd_{i,j} = gcd(Z_i, Z_j)$.

Let us note that, if the initial marking $M_0(p)$, $p \in P$ is fixed, the corresponding optimum period may be easily computed: for any place $p = (t_i, t_j) \in P$, let us denote by

$$H(p) = M_0(p) + gcd_{i,j} - Z_j \quad \text{and} \quad L(p) = \ell(t_i).$$

For a circuit c , $H(c) = \sum_{p \in c} H(p)$ and $L(c) = \sum_{p \in c} L(p)$. Theorem 2 expresses necessary and sufficient condition for the existence of a periodic schedule deduced easily from Bellman-Ford algorithm [27].

Theorem 2 ([12]). There exists a periodic schedule iff, for every circuit c of G , $H(c) > 0$.

The minimum feasible value K^{opt} of K is then:

$$K^{opt} = \max_{c \in C(\mathcal{G})} \frac{L(c)}{H(c)} \quad (1)$$

where $C(\mathcal{G})$ denotes the set of circuits of \mathcal{G} .

V. STUDY OF A BUFFER

A. Relationship between the optimum period and the capacity of a buffer

Let us consider a buffer modelled by a MTWEG \mathcal{G} with a couple of transitions (t_1, t_2) and a couple of places $(p_1, p_2) \in P^2$ with $p_1 = (t_1, t_2)$ and $p_2 = (t_2, t_1)$. We study here the relationship between the minimum period K of a periodic schedule and the capacity $F(p_1, p_2) = M_0(p_1) + M_0(p_2)$ of the corresponding buffer.

First lemma expresses a lower bound on the capacity:

Lemma 1. The minimum capacity for the existence of a periodic schedule is $F^{min}(p_1, p_1) = Z_1 + Z_2 - gcd_{1,2}$.

Proof: \mathcal{G} has three elementary circuits $c_1 = (t_1, t_1)$, $c_2 = (t_2, t_2)$ and $c_3 = (t_1, t_2, t_1)$. By Theorem 2, there exists a periodic schedule if $H(c_1) = Z_1 > 0$, $H(c_2) = Z_2 > 0$ and $H(c_3) = F(p_1, p_2) + 2gcd_{1,2} - (Z_1 + Z_2) > 0$. The last inequality is equivalent to $F(p_1, p_2) \geq (Z_1 + Z_2) - gcd_{1,2}$ since initial marking $M_0(p_1)$ and $M_0(p_2)$ are multiples of $gcd_{1,2}$. ■

One can note that $F^{min}(p_1, p_2)$ does not depend on the computation times of transitions (t_1, t_2) . Moreover, this value

TABLE I
PROCESSING TIMES $\ell(t_i)$, $t_i \in T$ IN MILLISECONDS

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
ℓ	2.3×10^{-2}	10	9.091	0.125	0.125	10	24	10	2.3×10^{-2}	0.125

is also equal to the lowest capacity of a given buffer ([14], [15]).

Let us suppose now that $F(p_1, p_2) \geq F^{\min}(p_1, p_2)$. From equation 1 in the previous section, the optimum value of the period for a fixed capacity $F(p_1, p_2)$ is

$$K^{\text{opt}} = \max \left\{ \frac{\ell(t_1)}{Z_1}, \frac{\ell(t_2)}{Z_2}, \frac{\ell(t_1) + \ell(t_2)}{F(p_1, p_2) + 2gcd_{1,2} - (Z_1 + Z_2)} \right\}$$

Without loss of generality, we may assume that $\frac{\ell(t_1)}{Z_1} \geq \frac{\ell(t_2)}{Z_2}$ and thus

$$K^{\text{opt}} = \max \left\{ \frac{\ell(t_1)}{Z_1}, \frac{\ell(t_1) + \ell(t_2)}{F(p_1, p_2) + 2gcd_{1,2} - (Z_1 + Z_2)} \right\}.$$

From a certain value of $F(p_1, p_2)$, K^{opt} remains equal to $\frac{\ell(t_1)}{Z_1}$. The following lemma characterizes this bound:

Lemma 2. *The maximum value of capacity is $F^{\max}(p_1, p_2) = \left\lceil \frac{\ell(t_2)}{\ell(t_1)} \cdot \frac{Z_1}{gcd_{1,2}} \right\rceil gcd_{1,2} + 2Z_1 + Z_2 - 2gcd_{1,2}$.*

Proof:

$F^{\max}(p_1, p_2)$ is the minimum value for $F(p_1, p_2)$ such as

$$\frac{\ell(t_1)}{Z_1} \geq \frac{\ell(t_1) + \ell(t_2)}{F(p_1, p_2) + 2gcd_{1,2} - (Z_1 + Z_2)}$$

Thus, we get

$$F(p_1, p_2) \geq \left(\frac{\ell(t_2)}{\ell(t_1)} + 2 \right) Z_1 + Z_2 - 2gcd_{1,2}.$$

Now, since $F^{\max}(p_1, p_2)$ is divisible by $gcd_{1,2}$,

$$F^{\max}(p_1, p_2) = \left\lceil \frac{\ell(t_2)}{\ell(t_1)} \cdot \frac{Z_1}{gcd_{1,2}} \right\rceil gcd_{1,2} + 2Z_1 + Z_2 - 2gcd_{1,2}.$$

Hence, the lemma. ■

The following theorem studies the determination of the optimum period for $F(p_1, p_2)$ between these two bounds:

Theorem 3. *The function $F(p_1, p_2) \rightarrow K^{\text{opt}}$ is strictly decreasing and strictly convex for $F(p_1, p_2) \in \{F^{\min}(p_1, p_2), F^{\min}(p_1, p_2) + gcd_{1,2}, \dots, F^{\max}(p_1, p_2) - gcd_{1,2}, F^{\max}(p_1, p_2)\}$.*

Proof: $M_0(p_1)$ and $M_0(p_2)$ are divisible by $gcd_{1,2}$, so $F(p_1, p_2) = M_0(p_1) + M_0(p_2)$ is. Now, for $F(p_1, p_2) \in \{F^{\min}(p_1, p_2), F^{\min}(p_1, p_2) + gcd_{1,2}, \dots, F^{\max}(p_1, p_2) - gcd_{1,2}, F^{\max}(p_1, p_2)\}$, the corresponding value of K^{opt} is $\frac{\ell(t_1) + \ell(t_2)}{F(p_1, p_2) + 2gcd_{1,2} - (Z_1 + Z_2)}$ which is strictly decreasing and strictly convex. Hence the result. ■

This last theorem provides a good rule of thumb for the designer: “The more you increase buffer capacity, the less it increases the throughput.”

Let $K^{\min}(p_1, p_2)$ (resp. $K^{\max}(p_1, p_2)$) denotes the value of K^{opt} for a buffer capacity equal to $F^{\min}(p_1, p_2)$ (resp. $F^{\max}(p_1, p_2)$).

For example, let us consider the couple of places $(p_7, p_7)_c$ from Figure 3. As seen before, normalized values for the adjacent transitions t_7 and t_8 are respectively $Z_7 = 84672$ and $Z_8 = 35280$ and $gcd_{7,8} = 7056$. We get the bounds $F^{\min}(p_7, p_7') = 16gcd_{7,8}$, $K^{\min}(p_7, p_7') = 4.82 \times 10^{-3}ms$, $F^{\max}(p_7, p_7') = 32gcd_{7,8}$ and $K^{\max}(p_7, p_7') = 2.83 \times 10^{-4}ms$. Figure 4 presents the variation of K^{opt} according to $F(p_7, p_7')$.

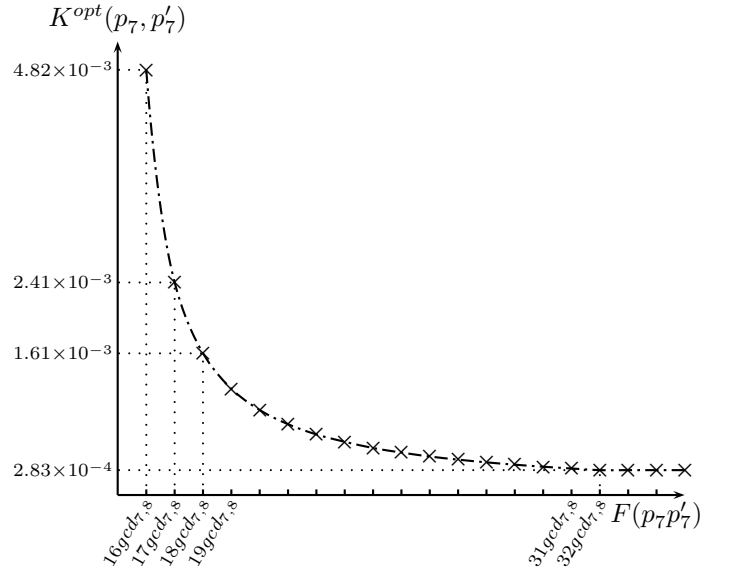


Fig. 4. K^{opt} according to $F(p_7, p_7') = M_0(p_7) + M_0(p_7')$. $F^{\min}(p_7, p_7') = 16gcd_{7,8}$, $K^{\min}(p_7, p_7') = 4.82 \times 10^{-3}ms$, $F^{\max}(p_7, p_7') = 32gcd_{7,8}$ and $K^{\max}(p_7, p_7') = 2.83 \times 10^{-4}ms$.

Conversely, for any fixed value $K \in [K^{\min}(p_1, p_2), K^{\max}(p_1, p_2)]$, the optimum value of capacity is $F_K(p_1, p_2) = \left(\left\lceil \frac{\ell(t_1) + \ell(t_2)}{K \cdot gcd_{1,2}} \right\rceil - 2 \right) gcd_{1,2} + (Z_1 + Z_2)$. Thus, the complexity to obtain the optimum capacity for a unique buffer is the computation of $gcd_{1,2}$ which is $O(\log(\max\{Z_i, Z_j\}))$.

B. Consequence for a symmetric MWTEG without circuits of more than two transitions

Let us suppose that \mathcal{G} is a symmetric MWTEG without any circuit of more than two transitions, i.e. the only circuits come from the buffer limitation. The undirected graph $G = (T, E)$, for which every couple of places $(p, p')_c \in P^2$ with $p = (t_i, t_j)$ and $p' = (t_j, t_i)$ is replaced by a unique edge $(t_i, t_j) \in E$, is a tree. This limitation on the structure of \mathcal{G} was extensively studied by many authors and corresponds to most of the applications (see. as example [22], [20], [21]).

Starting from the Integer Linear Program $\Pi(K)$ defined previously, we observe that there exists a periodic schedule of period K iff

- 1) $K \geq \max_{t_i \in T} \frac{\ell(t_i)}{Z_i}$ and,
- 2) for every circuit $c = (t_i, t_j, t_i)$ corresponding to a buffer, $L(c) - KH(c) \leq 0$.

The important point is that an optimum initial marking may be obtained by minimizing separately the capacity of each buffer.

For any fixed period $K \geq \max_{t_i \in T} \frac{\ell(t_i)}{Z_i}$, an optimum marking of places $p \in P$ is defined as follows. Let $(p, p')_c$ be a couple of places corresponding to a buffer:

- 1) If $K \leq K^{max}(p, p')$, then the capacity of the corresponding buffer must be equal to $F_K(p, p')$. So, we set $M_0(p) = F_K(p, p')$ and $M_0(p') = 0$;
- 2) Else, $K > K^{max}(p, p')$ and this period may be achieved by the minimum capacity $F^{min}(p, p')$. We set then $M_0(p) = F^{min}(p, p')$ and $M_0(p') = 0$.

This solution is clearly minimum for every buffer. So, it minimizes the overall weighted capacity of \mathcal{G} . The complexity time of this algorithm is $O(m \log(\max_{i \in \{1, \dots, n\}} Z_i))$.

C. Example

The example depicted by Figure 3 does not verify the previous assumption. However, we may consider as example the subgraph \mathcal{G}' limited to the transitions $T' = \{t_1, t_5, t_6, t_7, t_8, t_9\}$ corresponding to the mixing of the sounds coming from the MP3 reader and the cell phone to the output. The corresponding undirected graph defined as $G' = (T', \{\{t_7, t_8\}, \{t_5, t_6\}, \{t_6, t_1\}, \{t_8, t_1\}, \{t_1, t_9\}\})$ is clearly a tree.

Optimum values of capacity for this subgraph were calculated for $K = 2.83 \times 10^{-4}ms$ and different processing time of t_8 . Values of the other transitions are fixed following Table I to $\ell(t_1) = \ell(t_9) = 2.3 \times 10^{-2}ms$, $\ell(t_6) = 10ms$ and $\ell(t_7) = 24ms$. Table II summarizes minimum capacities for different values of $\ell(t_8)$.

Note that the processing time of $\ell(t_8)$ only influences adjacent buffers corresponding to couples of places $(p_7, p'_7)_c$ and $(p_8, p'_8)_c$. Their capacity decreases when $\ell(t_8)$ decreases. This point can be proved easily from the Integer Linear System $\Pi(K)$. It was also noticed experimentally by [22].

TABLE II
OPTIMAL INITIAL MARKINGS OF THE SUBGRAPH \mathcal{G}' FOR DIFFERENT
PROCESSING TIME OF t_8 .

$\ell(t_8) =$	10	7.5	5	2.5
$\theta(p_6)F(p_6, p'_6)$	882	882	882	882
$\theta(p_7)F(p_7, p'_7)$	3072	2976	2880	2784
$\theta(p_8)F(p_8, p'_8)$	882	772	662	552
$\theta(p_9)F(p_9, p'_9)$	2	2	2	2
Sum	4838	4632	4426	4220

VI. AN APPROXIMATION ALGORITHM FOR THE GENERAL CASE

We suppose here that $K \geq \max_{t_i \in T} \frac{\ell(t_i)}{Z_i}$ is a fixed value. Let us consider the Linear Program $\Pi^*(K)$ obtained from $\Pi(K)$

by replacing the condition $k_{i,j} \in \mathbb{N}$ by $k_{i,j} \in \mathbb{Q}^+$.

The idea of our approximation algorithm for the general case is to compute first polynomially an optimum solution $M_0^*(p) \in \mathbb{Q}$, $p \in P$ of $\Pi^*(K)$. A feasible solution $M_0(p)$ of $\Pi(K)$ is then deduced using a rounding algorithm.

Lemma 3. Let $(p, p')_c$ be a couple of place with $p = (t_i, t_j)$ and $p' = (t_j, t_i)$. Let also the value

$$F_K^*(p, p') = \frac{\ell(t_i) + \ell(t_j)}{K} - 2gcd_{i,j} + (Z_i + Z_j).$$

Every feasible solution M_0^* of $\Pi^*(K)$ verifies $M_0^*(p) + M_0^*(p') \geq F_K^*(p, p')$.

Proof: Let the circuit $c = (t_i, p, t_j, p', t_i)$ corresponding to a buffer. Then, $L(c) - KH(c) = \ell(t_i) + \ell(t_j) - K(Z_i + Z_j - M_0^*(p) - M_0^*(p') - 2gcd_{i,j})$. Since M_0^* is feasible for $\Pi^*(K)$, we get $L(c) - KH(c) \leq 0$ and thus $M_0^*(p) + M_0^*(p') \geq \frac{\ell(t_i) + \ell(t_j)}{K} - 2gcd_{i,j} + (Z_i + Z_j)$, the result. ■

The following theorem characterizes an optimum feasible solution of $\Pi^*(K)$:

Theorem 4. Let $M_0^*(p)$, $p \in P$ defined as, for every couple of place $(p, p')_c$, $M_0^*(p) = M_0^*(p') = \frac{1}{2}F_K^*(p, p')$. Then, $M_0^*(p)$, $p \in P$ is an optimum solution of $\Pi^*(K)$.

Proof: M_0^* is a feasible solution if, for every circuit $c = (t_1, p_1, t_2, p_2, \dots, t_k, p_k, t_{k+1})$ with $t_1 = t_{k+1}$, $L(c) - KH(c) = \sum_{i=1}^k \ell(t_i) + K(\sum_{i=1}^k Z_i - \sum_{i=1}^k M_0^*(p_i) - \sum_{i=1}^k gcd_{i,i+1}) \leq 0$. By definition of $M_0^*(p_i)$, we have

$$-M_0^*(p_i) + \frac{1}{2} \left(\frac{\ell(t_i) + \ell(t_{i+1})}{K} - 2gcd_{i,i+1} + Z_i + Z_{i+1} \right) = 0$$

By summing these equalities for all places of the circuit, we get

$$-\sum_{i=1}^k M_0^*(p_i) + \sum_{i=1}^k \frac{\ell(t_i)}{K} - \sum_{i=1}^k gcd_{i,i+1} + \sum_{i=1}^k Z_i = 0$$

$$\Updownarrow \times K$$

$$-K \sum_{i=1}^k M_0^*(p_i) + \sum_{i=1}^k \ell(t_i) - K \sum_{i=1}^k gcd_{i,i+1} + K \sum_{i=1}^k Z_i = 0$$

and thus

$$L(c) - KH(c) = 0.$$

So, M_0^* is a feasible solution. Now, from Lemma 3, $M_0^*(p) + M_0^*(p')$ is exactly the lower bound of the capacity of the buffer (p, p') and thus $\sum_{p \in P} \theta(p) M_0^*(p)$ is optimum. So, theorem holds. ■

Our approximation algorithm consists in setting, for every couple of places $(p, p')_c$ corresponding to a buffer with $p = (t_i, t_j)$ and $p' = (t_j, t_i)$, $M_0^*(p) = M_0^*(p') = \frac{1}{2}F_K^*(p, p')$ and $M_0^{App}(p) = M_0^{App}(p') = \left\lceil \frac{M_0^*(p)}{gcd_{i,j}} \right\rceil gcd_{i,j}$.

$M_0^{App}(p)$, $p \in P$ is clearly a feasible solution of $\Pi(K)$. The following lemma provides a lower bound of the overall capacity:

Lemma 4. Let $(p, p')_c$ be a couple of place with $p = (t_i, t_j)$ and $p' = (t_j, t_i)$ and let $a_K(p, p') = \left\lceil \frac{F_K^*(p, p')}{gcd_{i,j}} \right\rceil gcd_{i,j}$. Then $A_K = \frac{1}{2} \sum_{(p, p')_c \in P^2} \theta(p) a_K(p, p')$ is a lower bound of the minimum overall weighted capacity for a fixed period K .

Proof: Any feasible solution $M_0(p)$, $p \in P$ of $\Pi(K)$ verifies $M_0(p) + M_0(p') \geq M_0^*(p) + M_0^*(p')$.

From Lemma 3, $M_0^*(p) + M_0^*(p') \geq F_K^*(p, p')$, so $M_0(p) + M_0(p') \geq F_K^*(p, p')$. Since $M_0(p) + M_0(p')$ is divisible by $gcd_{i,j}$,

$$M_0(p) + M_0(p') \geq a_K(p, p')$$

and lemma is proved. \blacksquare

This last theorem bounds the overall capacity obtained by our algorithm:

Theorem 5.

$$\sum_{p \in P} \theta(p) M_0^{App}(p) \leq A_K + \frac{1}{2} \sum_{\substack{(p, p')_c \in P^2, \\ p = (t_i, t_j)}} \theta(p) gcd_{i,j}$$

Proof: For every couple of places $(p, p')_c \in P^2$ with $p = (t_i, t_j)$,

$$M_0^{App}(p) + M_0^{App}(p') = 2 \left\lceil \frac{F_K^*(p, p')}{2gcd_{i,j}} \right\rceil gcd_{i,j}.$$

Now, since

$$\begin{aligned} 2 \left\lceil \frac{F_K^*(p, p')}{2gcd_{i,j}} \right\rceil gcd_{i,j} &\leq \left\lceil \frac{F_K^*(p, p')}{gcd_{i,j}} \right\rceil gcd_{i,j} + gcd_{i,j} \\ &= a_K(p, p') + gcd_{i,j}, \end{aligned}$$

then

$$\sum_{p \in P} \theta(p) M_0^{App}(p) \leq \sum_{p \in P} \theta(p) a_K(p, p') + \frac{1}{2} \sum_{\substack{(p, p')_c \in P^2, \\ p = (t_i, t_j)}} \theta(p) gcd_{i,j}$$

and theorem holds. \blacksquare

A_K and $\sum_{p=(t_i, t_j) \in P} \theta(p) gcd_{i,j}$ are both lower bounds of the overall weighted capacity, and thus, we get a 2-approximation algorithm. This worse performance may be easily achieved for a symmetric Timed (non weighted) Event Graph.

VII. APPLICATION TO THE CAR-RADIO

Table III summarizes the values obtained for our example by the previous approximation algorithm.

We get the lower bound $A_K = 6348$ and the overall capacity of our solution is equal to 6350. It is thus at less than 0.04 percent from the optimum. Note that the capacity of most of the buffers equals the minimum value, thus it is optimum.

In [25], the whole circuit is not considered. However, the computed capacities for buffer $(p_3, p'_3)_c$ is 158, which is not minimum.

TABLE III
OPTIMAL BUFFERS CAPACITIES FOR THE MTWEG PICTURED BY FIGURE 3

Buffers	$\theta(p_i) a_K(p_i, p'_i)$	$\theta(p_i) F_K^{App}(p_i, p'_i)$
$(p_1, p'_1)_c$	882	882
$(p_2, p'_2)_c$	160	160
$(p_3, p'_3)_c$	153	154
$(p_4, p'_4)_c$	2	2
$(p_5, p'_5)_c$	160	160
$(p_6, p'_6)_c$	882	882
$(p_7, p'_7)_c$	3072	3072
$(p_8, p'_8)_c$	882	882
$(p_9, p'_9)_c$	2	2
$(p_{10}, p'_{10})_c$	153	154
Sum	6348	6350

VIII. CONCLUSION

We presented in this paper an original approach to solve efficiently the minimization of the buffer capacities with throughput constraints for a class of streaming applications. A simple mathematical modelization using an Integer Linear Program was first introduced. An exact polynomial simple algorithm was deduced from the theoretical study of the influence of the capacity on the throughput for an important special case of the MTWEG. A general polynomial approximation algorithm was also developed for the general case. The solutions obtained for simple practical examples are very close to the optimum and beats previous works. This new approach can be easily implemented to automatized the design of such systems.

REFERENCES

- [1] E. A. Lee and D. G. Messerschmitt, "Synchronous data flow," *IEEE Proceedings of the IEEE*, vol. 75, no. 9, 1987.
- [2] F. Commoner, A. W. Holt, S. Even, and A. Pnueli, "Marked directed graphs," *J. Comput. Syst. Sci.*, vol. 5, no. 5, pp. 511–523, 1971.
- [3] P. Chr tienne, "Les r seaux de petri temporis s," Ph.D. dissertation, Th se d tat, Universit  Pierre et Marie Curie, 1983.
- [4] C. V. Ramamoorthy and G. S. Ho, "Performance evaluation of asynchronous concurrent systems using petri nets," *IEEE Transactions on Software Engineering*, vol. SE-6, no. 5, pp. 440–449, September 1980.
- [5] S. Gaubert, "An algebraic method for optimizing resources in timed event graphs," in *9th conference on Analysis and Optimization of Systems*, vol. 144. Springer, 1990.
- [6] A. Giua, A. Piccaluga, and C. Seatzu, "Firing rate optimization of cyclic timed event graphs," *Automatica*, vol. 38, no. 1, 2002.
- [7] S. Laftit, J.-M. Proth, and X. Xie, "Optimization of invariant criteria for event graphs," *IEEE Transactions on Automatic Control*, vol. 37, no. 5, 1989.
- [8] O. Marchetti, "Dimensionnement des m moires pour syst mes embarqu s," Ph.D. dissertation, Universit  Pierre et Marie Curie, 2006.
- [9] E. Teruel, P. Chrzastowski-Wachtel, J. M. Colom, and M. Silva, "On weighted T-systems," in *Proceedings of the 13th International Conference on Application and Theory of Petri Nets 1992, Lecture Notes in Computer Science*, vol. 616. Springer, 1992.
- [10] A. Munier, "R gime asymptotique optimal d un graphe d  v nements temporis  g n ralis : application   un probl me d assemblage," *RAIRO-Automatique Productique Informatique Industrielle*, vol. 27, no. 5, pp. 487–513, 1993.
- [11] N. Sauer, "Marking optimization of weighted marked graph," *Discrete Event Dynamic Systems*, vol. 13, no. 3, pp. 245–262, 2003.
- [12] A. Benabid, C. Hanen, O. Marchetti, and A. M. Kordon, "Periodic schedules for unitary timed weighted event graphs," in *Conference ROADEF'08*. Clermont-Ferrand: Presses Universitaires de l'Universit  Blaise Pascal, 2008, pp. 17–31.
- [13] R. Reiter, "Scheduling parallel computations," *Journal of the Association for Computing Machinery*, vol. 15, no. 4, pp. 590–599, 1968.

- [14] M. Adé, R. Lauwereins, and J. A. Peperstraete, "Buffer memory requirements in dsp applications," in *In Proceedings of the IEEE Workshop on Rapid System Prototyping*. Washington, DC, USA: IEEE Computer Society, 1994, pp. 108–123.
- [15] P. K. Murthy, "Scheduling techniques for synchronous and multi-dimensional synchronous dataflow." Ph.D. dissertation, University of California, Berkeley, 1996.
- [16] M. Adé, R. Lauwereins, and J. A. Peperstrate, "Implementing dsp applications on heterogeneous targets using minimal size data buffers," in *RSP '96: Proceedings of the 7th IEEE International Workshop on Rapid System Prototyping (RSP '96)*. Washington, DC, USA: IEEE Computer Society, 1996, p. 166.
- [17] M. Adé, R. Lauwereins, and J. A. Peperstraete, "Data memory minimisation for synchronous data flow graphs emulated on dsp-fpga targets," in *DAC*, 1997, pp. 64–69.
- [18] R. Govindarajan and G. Gao, "Rate-optimal shcedule for multi-rate dsp computations," *Journal of VLSI Signal Processing*, no. 9, pp. 211–235, 1995.
- [19] R. Govindarajan, G. R. Gao, and P. Desai, "Minimizing memory requirements in rate-optimal schedule in regular dataflow networks," *Journal of VLSI Signal Processing*, vol. 31, no. 3, 2002.
- [20] S. Stuijk, M. Geilen, and T. Basten, "Exploring trade-offs in buffer requirements and throughput constraints for synchronous dataflow graphs," in *43rd Design Automation Conference, DAC 2006*. ACM Press, 2006.
- [21] A. H. Ghamarian, M. Geilen, T. Basten, and S. Stuijk, "Parametric throughput analysis of synchronous data flow graphs," in *DATE*, 2008, pp. 116–121.
- [22] M. Wiggers, M. Bekooij, P. Jansen, and G. Smit, "Efficient computation of buffer capacities for multi-rate real-time systems with back-pressure," in *CODES+ISSS '06: Proceedings of the 4th international conference on Hardware/software codesign and system synthesis*. New York, NY, USA: ACM, 2006, pp. 10–15.
- [23] O. Marchetti and A. Munier-Kordon, "Minimizing place capacities of weighted event graphs for enforcing liveness," *Discrete Event Dynamic Systems*, vol. 18, no. 1, pp. 91–109, 2008.
- [24] —, "A sufficient condition for the liveness of weighted event graphs," *To appear in EJOR, LIP6 Research Report*, <http://ftp.lip6.fr/lip6/reports/2005/lip6-2005-004.pdf>, 2008.
- [25] M. H. Wiggers, M. J. G. Bekooij, P. G. Jansen, and G. J. M. Smit, "Efficient computation of buffer capacities for cyclo-static real-time systems with back-pressure," in *RTAS '07: Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 281–292.
- [26] R. M. Karp and R. E. Miller, "Properties of a model for parallel computations: Determinacy, termination, queueing," *SIAM*, vol. 14, no. 63, pp. 1390 – 1411, 1966.
- [27] T. H. Cormen, C. E. Leiseerson, R. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 1990.