A Buffer Minimization Problem for the Design of Embedded Systems

Alix Munier Kordon Jean-Baptiste Note

October 8, 2002

Abstract

We consider a set of n tasks, each of them is composed by a set of sequential operations. A set of buffers $\mathcal{B}$ is given : each buffer $b \in \mathcal{B}$ is defined between two tasks $T_i \to T_j$, has a weight w_b and is managed as a FIFO structure. Some operations from T_i write data in the buffer b, other from T_j get data in b.

The writings and readings on buffers generate precedence constraints between the operations. The limitation of the size of the buffers generates an other set of precedence constraints between them and circuits in this precedence graph may appear. In this case, there is no feasible schedule for the operations. The aim is to find the size of each buffer $\Delta(b), b \in \mathcal{B}$ such that $\sum_{b \in \mathcal{B}} w_b \Delta(b)$ is minimum and there is no circuit in the precedence graph.

We prove that this problem is polynomial for 2 tasks using a flow algorithm. We also prove that it is NP-hard in the strong sense for 3 tasks.

1 Introduction

An embedded system is the association of an hardware and a software system to realize a given application : the synthesis of such a system is divided into several phases. In this paper, we consider that the allocation of the tasks to the processors is given. These tasks communicate using buffers. The model of communication that we consider is knowned as Kahn Process Network [4], and suppose that the capacity of the buffers is not bounded.

The problem is that to realize the system on a chip, the size of the buffers has to be bounded : moreover, the cost of a buffer is proportionnal to the size of the data that can be stored by it. So, the problem tackle in this paper is to limit the size of the buffers such that the application can be realize and the global surface of the buffers is minimum.

In the litterature, few studies are dedicated to this field. Authors usually arbitrarily bound the capacities of the buffers and simulate to show if no deadlocks appear [3]. In [5], the author has develop a greedy heuristic that increase the capacity of a buffer everytime a deadlock appears.

This paper is organized as follows : in the section 2, we define the problem and we show how the communications with buffers can be modelled using a precedence graph. In section 3, we prove that the problem is polynomial for 2 sequentiel tasks using a flow algorithm. In section 4, we show that the associated decision problem is NP-complete in the strong sense for 3 sequential tasks. We give some ideas for further research in section 5.

2 Problem definition

Let us consider a set of n tasks denoted by $\mathcal{T} = \{T_1, \dots, T_n\}$ and corresponding to the load of n different processors. These tasks may exchange data using a given set of buffers $\mathcal{B}$ managed as a FIFO (First In/First Out) structures. Each buffer $b \in \mathcal{B}$ is defined by a couple of tasks (T_i, T_j) : T_i (*resp.* T_j) is the task allowed to write (*resp.* read) data in b. Moreover, the values stored by a buffer are all of the same type : $\forall b \in \mathcal{B}$, w_b is the size of a value that may be stored by b.

The Kahn process network of an application is an oriented graph where the vertices correspond to the tasks and the arcs to the buffers.

More precisely, each task T_i may be decomposed into n_i sequential operations $o_i^1, \dots, o_i^{n_i}$ which write or read a value in a buffer. Let us suppose that task T_i sends data to T_j. $\mathcal{B}_{ij}$ is the set of buffers from T_i to T_j dedicated to these communications.

Each operation o_i^p is associated with a buffer denoted by $b(o_i^p)$. If o_i^p write a data in $b(o_i^p)$, then $b(o_i^p) \in \cup_{j \in \mathcal{T}} \mathcal{B}_{ij}$, else, $b(o_i^p) \in \cup_{j \in \mathcal{T}} \mathcal{B}_{ji}$.

The capacity of a buffer is the maximal number of values that can be stored simultaneously. It is denoted by $\Delta(b), b \in \mathcal{B}$ and is a variable of the problem. The problem is that writing in a full buffer is not possible : in this case, the next writing operation has to wait until a reading operation

Table 1: $N(o_i^p)$, $i = 1, 2$, $p = 1, \dots, 7$

i	1	2	3	4	5	6	7
$N(o_1^i)$	0	1	0	1	2	0	1
$N(o_2^i)$	0	1	0	1	0	1	2

empties the buffer. Deadlocks may occur if the size of some buffers is too small.

Now, the surface needed in a component for b is proportional to $w_b \Delta(b)$. The aim here is to minimize this global surface of the buffers noted by $C_\Sigma = \sum_{b \in \mathcal{B}} w_b \Delta(b)$.

Definition of the precedence graph We prove here that the communication with buffers can be modeled by a precedence graph on operations. Let us define, for every operation o_i^p, the value $N(o_i^p)$ as :

$$N(o_i^p) = |\{q \in \{1, \dots, p-1\}, b(o_i^q) = b(o_i^p)\}|$$

- If o_i^p is a writing operation, then $N(o_i^p)$ is the number of values written in the buffer $b(o_i^p)$ before the execution of o_i^p.
- Else, o_i^p is a reading operation and $N(o_i^p)$ is the number of values read in the buffer $b(o_i^p)$ before the execution of o_i^p.

For example, let us consider two tasks T_1, T_2, each of them composed by 7 operations and 3 buffers b, b' and b'' with $\mathcal{B}_{12} = \{b\}$ and $\mathcal{B}_{21} = \{b', b''\}$. Moreover, $b = b(o_1^1) = b(o_1^2) = b(o_1^5) = b(o_2^5) = b(o_2^6) = b(o_2^7)$, $b' = b(o_1^3) = b(o_1^4) = b(o_2^3) = b(o_2^4)$ and $b'' = b(o_2^1) = b(o_2^2) = b(o_1^6) = b(o_1^7)$. N is presented by table 1.

Let us suppose that, $\forall b \in \mathcal{B}$, the capacity of the buffers is denoted by $\Delta(b), b \in \mathcal{B}$. Then, the constraints between the operations can be modeled by a precedence graph $G(\Delta) = (V, E(\Delta))$ built as follows :

- The vertices of the graph are the operations o_i^p, $i \in \{1, \dots, n\}$, $p \in \{1, \dots, n_i\}$.
- Precedence relations $E(\Delta)$ can be split into 3 classes E_1, E_2 and $E_3(\Delta)$ defined as follows :

1. The operations of any task T_i are performed sequentially.

$$E_1 = \bigcup_{i=1}^{n} \{(o_i^p, o_i^{p+1}), p = 1, \ldots, n_i - 1\}$$

2. Let us suppose that o_i^p is a writing operation, o_j^q is a reading operation and that they used the same buffer ($i.e$ $b(o_i^p) = b(o_j^q) = b$). Then, o_j^q get the value written in b by o_i^p if and only if $N(o_i^p) = N(o_i^q)$. An arc $(o_i^p, o_j^q) \in E_2$ iff $N(o_i^p) = N(o_i^q)$.
3. Using the same notations as the previous case, o_i^p has to wait for the execution of o_j^q if :
   (a) Before the execution of o_j^q, there is no enough room for o_i^p to write an additional value in b, so $N(o_i^p) + 1 - N(o_j^q) > \Delta(b)$.
   (b) There is enough room in b after the execution of o_j^q, so $N(o_i^p) + 1 - (N(o_j^q) + 1) \leq \Delta(b)$.

   Then, an arc $(o_j^q, o_i^p) \in E_3(\Delta)$ iff $N(o_i^p) - N(o_j^q) = \Delta(b)$.

We also denote by $G(\infty)$ the precedence graph with no limitations on the capacity of the buffers : we get $E(\infty) = E_1 \cup E_2$. In the following, we suppose that $G(\infty)$ has no circuit : in the opposite, there is a mistake in the application. The graph $G(\infty)$ of our example is pictured by figure 1.

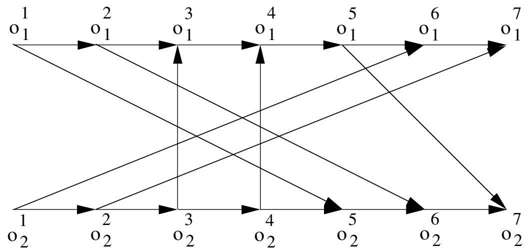

Figure 1: $G(\infty)$ of our example

If the size of some buffers is too small, some circuit may occur in $G(\Delta)$. For example, if $\Delta(b'') = 1$, we obtain a circuit between the operations o_1^6, o_2^2, o_2^3 and o_1^3.

The problem is to find $\Delta(b)$ for every $b \in \mathcal{B}$ such that the sum $C_\Sigma = \sum_{b \in \mathcal{B}} w_b \Delta(b)$ is minimum and that $G(\Delta)$ has no circuit.

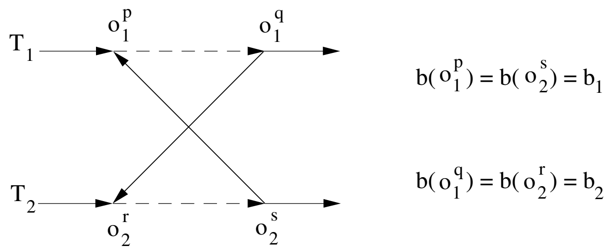

Figure 2: Structure of a circuit

3 A polynomial algorithm for two tasks

Let us consider two tasks T_1 and T_2 which communicate using a set of buffers $\mathcal{B}$. We study firtly the structure of the circuit in the precedence graph. Then, we prove that the presence of circuits can be described as a set of clauses of size less than or equal to 2. Lastly, we show that the problem is equivalent with the decision problem associated with the minimum weighted vertex cover of a bipartite graph, which is knwoned to be polynomial [1].

3.1 Structure of the circuits

We suppose here that the capacity Δ of the buffers is fixed. The following theorem characterizes the structure of the circuits of $G(\Delta)$ (if it exists).

Theorem 1. *Let c be a circuit of $G(\Delta)$. Then, there are 4 operations o_1^p, o_1^q, o_2^r and o_2^s with $p < q$ and $r < s$ such that (o_2^s, o_1^p) and (o_1^q, o_2^r) are in $G(\Delta)$ (see figure 2).*

Proof
Let o_1^q ($resp. o_2^s$) be the operation of $T_1 \cap c$ ($resp. T_2 \cap c$) with a maximum q ($resp. s$). Since o_1^q and o_2^s are in c, there exist in $G(\Delta)$ the arcs (o_1^q, o_2^r) with $r \leq s$ and (o_2^s, o_1^p) with $p \leq q$.

We prove now by contradiction that $r < s$ and $p < q$. Indeed, if $r = s$, then $o_2^s = o_2^r$ and $b = b(o_2^s) = b(o_1^p) = b(o_1^q)$.

1. If o_1^q is a writing operation, b is a buffer from T_1 to T_2.

   - The arc $(o_1^q, o_2^s) \in E_2$, so $N(o_1^q) = N(o_2^s)$.

- The arc $(o_2^s, o_1^p) \in E_3(\Delta)$, so $N(o_1^p) - N(o_2^s) = \Delta(b)$.

Since $N(o_1^p) \leq N(o_1^q)$, we get $\Delta(b) \leq 0$, which is impossible.

2. Now, if o_1^q is a reading operation, b is a buffer from T_2 to T_1.

   - The arc $(o_1^q, o_2^s) \in E_3(\Delta)$ so $N(o_2^s) - N(o_1^q) = \Delta(b)$.
   - The arc $(o_2^s, o_1^p) \in E_2$ so $N(o_1^p) = N(o_2^s)$.

   We get $N(o_1^p) - N(o_1^q) = \Delta(b)$. As $N(o_1^p) \leq N(o_1^q)$ we obtain again $\Delta(b) \leq 0$. $\square$

3.2 Conditions on the capacities of the buffers

We consider here a couple of buffers $(b_1, b_2) \in \mathcal{B}$. A basic circuit c associated with a couple of buffers $(b_1, b_2) \in \mathcal{B}$ is composed by 4 operations o_1^p, o_1^q, o_2^r and o_2^s with $p < q$ and $r < s$ and by the paths $\nu_1 = (o_1^p, o_1^{p+1}, \ldots o_1^q)$, $\nu_2 = (o_2^r, o_2^{r+1}, \ldots o_2^s)$ and the arcs (o_1^q, o_2^r) and (o_2^s, o_1^p). Moreover, we have $b_2 = b(o_1^q) = b(o_2^r)$ and $b_1 = b(o_1^p) = b(o_2^s)$.

In this section, we fix a couple of buffers $(b_1, b_2) \in \mathcal{B}^2$, $b_1 \neq b_2$, and we express necessary and sufficient conditions on the capacities $\Delta(b_1)$ and $\Delta(b_2)$ to avoid basic circuits associated with this couple of buffers. We have 3 cases to consider :

Case 1 If $(b_1, b_2) \in \mathcal{B}_{21} \times \mathcal{B}_{12}$, then arcs (o_1^q, o_2^r) and (o_2^s, o_1^p) are both in E_2. The lemma 1 follows :

Lemma 1. *If $(b_1, b_2) \in \mathcal{B}_{21} \times \mathcal{B}_{12}$ and $G(\infty)$ has no circuit, then there is no basic circuit in $G(\Delta)$ associated with these two buffers.*

Case 2 If $(b_1, b_2) \in \mathcal{B}_{12}^2 \cup \mathcal{B}_{21}^2$, then we consider two subcases :

1. If $(b_1, b_2) \in \mathcal{B}_{12}^2$, then $(o_1^q, o_2^r) \in E_2$ and $(o_1^p, o_2^s) \in E_3(\Delta)$. So, the only way to eliminate this circuit is to increase $\Delta(b_1)$.

   Let x_1 be the minimum capacity of b_1 such that there is no basic circuit associated with the couple (b_1, b_2) in the corresponding precedence graph. We set $y_1 = 1$. Then, a precedence graph $G(\Delta)$ has no basic circuit associated with (b_1, b_2) iff $\Delta(b_1) \geq x_1 \wedge \Delta(b_2) \geq y_1$.

2. By symmetry, we get the same property for the case $(b_1, b_2) \in \mathcal{B}_{21}^2$.

Lemma 2. *If $(b_1, b_2) \in \mathcal{B}_{12}^2 \cup \mathcal{B}_{21}^2$, there exists two integers x_1 and y_1 such that $G(\Delta)$ has no circuit associated with (b_1, b_2) if and only if $\Delta(b_1) \geq x_1 \wedge \Delta(b_2) \geq y_1$.*

Case 3 If $(b_1, b_2) \in \mathcal{B}_{12} \times \mathcal{B}_{21}$, then arcs (o_1^q, o_2^r) and (o_2^s, o_1^p) are both in $E_3(\Delta)$. We define a set of points $u_i = (x_i, y_i), i = 1, \ldots, l$ as follows :

1. $y_1 = 1$ and x_1 is the minimum value of the capacity of b_1 such that, if a capacity Δ' verifies $\Delta'(b_1) = x_1$ and $\Delta'(b_2) = 1$, then there is no basic circuit associated with the couple of buffers (b_1, b_2) in $G(\Delta')$. $u_1 = (x_1, y_1)$.
2. Let us assume that $u_i, i \geq 1$ has been defined.
   - If $x_i = 1$, we stop the recurrence and we set $l = i$.
   - Else, we set y_{i+1} the smallest value strictly greater than y_i such that if $\Delta'(b_2) = y_{i+1}$, then minimum value needed for $\Delta'(b_1)$ to avoid basic circuits associated with (b_{1,b_2}) in $G(\Delta')$ is denoted by x_{i+1} and verifies $x_{i+1} < x_i$. We get then $u_{i+1} = (x_{i+1}, y_{i+1})$.

Notice that the value l is bounded by the number n of operations. Indeed, if $\Delta'(b_2) = n$, there is no associated precedence constraint in $E_3(\Delta')$ and we can set $\Delta'(b_1) = 1$.

Lemma 3. *If $(b_1, b_2) \in \mathcal{B}_{12} \times \mathcal{B}_{21}$, there exists a sequence $u_i = (x_i, y_i), i = 1, \ldots, l$ such that $G(\Delta)$ has no basic circuit associated with the couple (b_1, b_2) if and only if $\exists i \in \{1, \ldots, l\}$ with $\Delta(b_1) \geq x_i$ and $\Delta(b_2) \geq y_i$.*

Proof

1. $A \Rightarrow B$: let us suppose that, $\forall i \in \{1, \ldots, l\}$, $\Delta(b_1) < x_i$ or $\Delta(b_2) < y_i$.
   - If $\Delta(b_2) \geq y_l$, then $\forall i \in \{1, \ldots, l\}, \Delta(b_1) < x_i$. As $x_l = 1$, this is impossible.
   - So, there exists $k \in \{1, \ldots, l-1\}$ with $y_k \leq \Delta(b_2) < y_{k+1}$. By hypothesis, we get $\Delta(b_1) < x_i$ for $i \in \{1, \ldots, k\}$.

Now, by definition of the sequences u_i, if $y_k \leq \Delta(b_2) < y_{k+1}$, then the minimum value for $\Delta(b_1)$ to avoid basic circuits associated with (b_1, b_2) is x_k. So, $G(\Delta)$ has a basic circuit associated with (b_1, b_2).

2. $B \Rightarrow A$: If $\exists i \in \{1, \ldots, l\}$ with $\Delta(b_1) \geq x_i$ and $\Delta(b_2) \geq y_i$, then, by definition of u_i, $G(\Delta)$ has no circuit. □

For any $(b_1, b_2) \in \mathcal{B}^2$, we set

$$C(b_1, b_2) = \bigvee_{i=1}^{l} (\Delta(b_1) \geq x_i \wedge \Delta(b_2) \geq y_i)$$

For the case 1, we set $l = 1$ and $x_1 = y_1 = 1$. We prove the following theorem :

Theorem 2. *The graph $G(\Delta)$ has no circuit if and only if, for every couple of buffers $(b_1, b_2) \in \mathcal{B}$ with $b_1 \neq b_2$, $C(b_1, b_2)$ is true.*

Proof

- $A \Rightarrow B$: If there is a couple of buffer for which $C(b_1, b_2)$ is false, then, by lemmas 2 and 3, $G(\Delta)$ has a circuit.
- $B \Rightarrow A$: if $G(\Delta)$ has a circuit c, then, using theorem 1, we can build a basic circuit associated with a couple of buffers (b_1, b_2). Using lemmas 2 and 3, we know that $C(b_1, b_2)$ is false. □

3.3 Description of the polynomial algorithm for 2 tasks

The following lemma is a rewriting of $C(b_1, b_2)$,

Lemma 4. $\forall (b_1, b_2) \in \mathcal{B}$,

$$C(b_1, b_2) = (\Delta(b_1) \geq x_l) \wedge (\Delta(b_2) \geq y_1) \wedge \bigwedge_{i=1}^{l-1} (\Delta(b_1) \geq x_i \vee \Delta(b_2) \geq y_{i+1})$$

Proof
We prove it by recurrence on l :

- The lemma is trivially true for logical expressions with $l = 1$.
- Let us suppose now that the lemma is true for l and that the sequences associated with $C(b_1, b_2)$ have $l+1$ terms. Then

$$C(b_1, b_2) = (\Delta(b_1) \geq x_{l+1} \wedge \Delta(b_2) \geq y_{l+1}) \\ \vee (\Delta(b_1) \geq x_l \wedge \Delta(b_2) \geq y_1 \wedge \bigwedge_{i=1}^{l-1} (\Delta(b_1) \geq x_i \vee \Delta(b_2) \geq y_{i+1}))$$

  Using the distributivity of the logical operators, we get for $C(b_1, b_2)$ the following clauses :

  1. $\Delta(b_1) \geq x_l \vee \Delta(b_1) \geq x_{l+1} = (\Delta(b_1) \geq x_{l+1})$;
  2. $\Delta(b_1) \geq x_{l+1} \vee \Delta(b_2) \geq y_1$ is always true because of the first clause and can be suppressed;
  3. $\forall i \in \{1, \ldots, l-1\}$, $\Delta(b_1) \geq x_{l+1} \vee (\Delta(b_1) \geq x_i \vee \Delta(b_2) \geq y_{i+1})$ is always true since the first clause is included in these terms;
  4. $\Delta(b_2) \geq y_{l+1} \vee \Delta(b_1) \geq x_l$;
  5. $\Delta(b_2) \geq y_{l+1} \vee \Delta(b_2) \geq y_1 = \Delta(b_2) \geq y_1$;
  6. $\forall i \in \{1, \ldots, l-1\}$, $\Delta(b_2) \geq y_{l+1} \vee (\Delta(b_1) \geq x_i \vee \Delta(b_2) \geq y_{i+1}) = (\Delta(b_1) \geq x_i \vee \Delta(b_2) \geq y_{i+1})$;

  If we consider all these clauses except the second and the third ones, we get the lemma. □

Lemma 5. *Let I be an instance of the problem with two tasks. $\forall (b_1, b_2) \in \mathcal{B}^2$, $C(b_1, b_2)$ may be computed in $O(n^2)$.*

Proof
Let n be the number of operations of each task. In the case 3, $l \leq |n|$ and at each step, the presence of circuits must be tested. □

MINIMUM WEIGHTED VERTEX COVER OF A BIPARTITE GRAPH

- Instance: an undirected bipartite graph $X = (A \cup B, E)$, and a function $v : A \cup B \rightarrow I\!N$.

- Question: is there a cover D of the vertices of X ($i.e$, a subset $D \subset A \cup B$ such that, for every edge $e = \{u, w\} \in E$, at leat $u \in D$ or $w \in D$) such that $\sum_{u \in D} v(u)$ is minimum ?

Let us consider an instance I of our problem with two tasks. For every couple $(b_1, b_2) \in \mathcal{B}^2$, we can associate a logical expression $C(b_1, b_2)$. Using notations of lemma 4, we define the following sequences to describe the extremum values of the capacity of the buffers :

$$x_{min}(C(b_1, b_2)) = x_l \text{ and } x_{max}(C(b_1, b_2)) = x_1$$

$$y_{min}(C(b_1, b_2)) = y_1 \text{ and } y_{max}(C(b_1, b_2)) = y_l$$

Then, we get for every $b \in \mathcal{B}$ the minimum and maximum feasible values for the capacity are $\Delta_{min}(b) = \max_{b' \in \mathcal{B}_{12}} x_{min}(C(b, b'))$ and $\Delta_{max}(b) = \max\{\max_{b' \in \mathcal{B}} x_{max}(C(b, b')), \max_{b' \in \mathcal{B}} y_{max}(C(b', b))\}$.

For example, let us consider 2 tasks T_1 and T_2 and the buffers $\mathcal{B}_{12} = \{b, b'\}$ and $\mathcal{B}_{21} = \{b''\}$. We suppose that $w_b = 1$, $w_{b'} = 4$ and $w_{b''} = 2$. We also consider the following logical expressions : $C(b, b') = \Delta(b) \geq 3 \wedge \Delta(b') \geq 1$, $C(b', b) = \Delta(b') \geq 1 \wedge \Delta(b) \geq 1$, $C(b, b'') = \Delta(b) \geq 2 \wedge \Delta(b'') \geq 1 \wedge (\Delta(b) \geq 5 \vee \Delta(b'') \geq 7) \wedge (\Delta(b) \geq 7 \vee \Delta(b'') \geq 4)$, $C(b'', b) = \Delta(b'') \geq 1 \wedge \Delta(b) \geq 1$, $C(b', b'') = \Delta(b') \geq 2 \wedge \Delta(b'') \geq 1 \wedge (\Delta(b') \geq 3 \wedge \Delta(b'') \geq 3)$, and $C(b'', b') = \Delta(b'') \geq 1 \wedge \Delta(b') \geq 1$. The extremum values are $\Delta_{min}(b) = 3$, $\Delta_{min}(b') = 2$, $\Delta_{min}(b'') = 1$, and $\Delta_{max}(b) = 7$, $\Delta_{max}(b') = 3$ and $\Delta_{max}(b'') = 7$.

- For every buffer $b \in \mathcal{B}$, we define the sequence $\Delta^i(b), i = 0, \ldots, k(b)$ of feasible values of $\Delta(b)$ as follows :

  1. $\Delta^0(b) = \Delta_{min}(b)$;
  2. For $i > 0$, if there exists a minimum value $x > \Delta^{i-1}(b)$ such that there is a logical expression $C(b, b')$ or $C(b', b)$ with the inequality $\Delta(b) \geq x$, then set $\Delta^i(b) = x$ and continue with $i+1$. Else, we get $\Delta^i(b) = \Delta_{max}(b)$ then we set $k(b) = i$ and we stop the recurrence.

We build an instance $f(I)$ of the minimum weighted cover problem of a bipartite graph as follows :

- For every buffer $b \in \mathcal{B}$, we associate $k(b)$ vertices $(b, 1), \ldots, (b, k(b))$ with the weight $v(b, i) = w_b(\Delta^i(b) - \Delta^{i-1}(b))$, $i = 1, \ldots, k(b)$.

  We denote by $A = \{(b, 1), \ldots, (b, k(b)), b \in \mathcal{B}_{12}\}$ and by $B = \{(b, 1), \ldots, (b, k(b)), b \in \mathcal{B}_{21}\}$

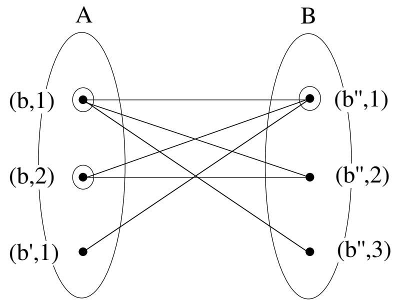

Figure 3: Graph $X = (A \cup B, E)$ and a cover D

- For every couple $(b_1, b_2) \in \mathcal{B}_{12} \times \mathcal{B}_{21}$, we build an edge between the vertices (b_1, k_1) and (b_2, k_2) if there exists a term $(\Delta(b_1) \geq x_i \vee \Delta(b_2) \geq y_{i+1})$ in $C(b_1, b_2)$ with $\Delta^{k_1}(b_1) \leq x_i$ and $\Delta^{k_2}(b_2) \leq y_{i+1}$.

  Let E be the set of edges. The graph $X = (A \cup B, E)$ is bipartite.

For our former example, we get :

- $k(b) = 2$, $\Delta^0(b) = 3$, $\Delta^1(b) = 5$ and $\Delta^2(b) = 7$. $v(b,1) = 2$ and $v(b,2) = 2$.
- $k(b') = 1$ and $v(b',1) = 4$.
- $k(b'') = 3$, $\Delta^0(b'') = 1$, $\Delta^1(b'') = 3$, $\Delta^2(b'') = 4$ and $\Delta^1(b'') = 7$. $v(b'',1) = 4$, $v(b'',2) = 2$ and $v(b'',3) = 6$.

The corresponding graph is pictured by figure 3.

Lemma 6. *f is a polynomial transformation.*

Proof
Let us denote by m the total number of buffers and by L the maximum number of inequalities of a logical expression $C(b, b')$. Then, $|A| + |B| \leq mL$. So, f is polynomial. $\Box$

Lemma 7. *Let Δ^* be a solution of an intance I. We can build a cover D for the corresponding instance $f(I)$. Moreover,*

$$\sum_{b \in \mathcal{B}_{12} \cup \mathcal{B}_{21}} w_b \Delta^*(b) = \sum_{b \in \mathcal{B}_{12} \cup \mathcal{B}_{21}} w_b \Delta^0(b) + \sum_{i \in D} v(i)$$

Proof
For every buffer $b \in \mathcal{B}_{12} \cup \mathcal{B}_{21}$, we defined $\alpha^*(b)$ as $\Delta^{\alpha^*(b)}(b) = \Delta^*(b)$. We set $D = \{(b, i), b \in \mathcal{B}_{12} \cup \mathcal{B}_{21}, 1 \leq i \leq \alpha^*(b)\}$.

We prove that D is a cover of X : for every edge $\{(b_1, k_1), (b_2, k_2)\}$ in the graph X, there exists a term $(\Delta(b_1) \geq x_i \vee \Delta(b_2) \geq y_{i+1})$ of $C(b_1, b_2)$ with $\Delta^{k_1}(b_1) \leq x_i$ and $\Delta^{k_2}(b_2) \leq y_{i+1}$.

Since Δ^* is a solution of I, we get $(\Delta^*(b_1) \geq x_i \vee \Delta^*(b_2) \geq y_{i+1})$. If $\Delta^*(b_1) \geq x_i$, then we get $\Delta^{k_1}(b_1) \leq \Delta^*(b_1)$, so $k_1 \leq \alpha^*(b_1)$ and $(b_1, k_1) \in D$. On the same way, if $\Delta^*(b_2) \geq y_{i+1}$, $(b_2, k_2) \in D$. So, D is a feasible solution of $f(I)$.

Now, $\sum_{i \in D} v(i) = \sum_{b \in \mathcal{B}_{12} \cup \mathcal{B}_{21}} w_b(\Delta^{\alpha^*(b)}(b) - \Delta^0(b)) = \sum_{b \in \mathcal{B}_{12} \cup \mathcal{B}_{21}} w_b(\Delta^*(b) - \Delta^0(b))$, so the equality holds. □

Four our example, $\Delta^*(b) = 7$, $\Delta^*(b') = 2$ and $\Delta^*(b'') = 4$ is a solution to I. For $f(I)$, we obtain $D = \{(b, 1), (b, 2), (b'', 1)\}$. The equality trivially holds.

Lemma 8. *Let D be a solution for $f(I)$. Then, we can build a solution Δ of I. Moreover, the previous equality between the two criteria holds.*

Proof

Let D be a solution for $f(I)$. We prove that, $\forall b \in \mathcal{B}_{12} \cup \mathcal{B}_{21}$, there exists a maximum value $\alpha(b) \in \{1, \ldots, k(b)\}$ such that all the vertices $(b, 1), \ldots,$ $(b, \alpha(b))$ are in D.

By contradiction, let us suppose that there exists a vertex $(b, \alpha) \in D$ with $(b, \alpha - 1) \notin D$. By construction of X, if $\Gamma(b, \alpha)$ denotes the set of adjacent vertices of (b, α), we get $\Gamma(b, \alpha) \subset \Gamma(b, \alpha - 1)$. Since $(b, \alpha - 1) \notin D$, then $\Gamma(b, \alpha - 1) \subset D$. So, (b, α) can be removed from D; D is not optimal, the contradiction.

For every $b \in \mathcal{B}_{12} \cup \mathcal{B}_{21}$, we set $\Delta^*(b) = \Delta^{\alpha(b)}(b)$. For any couple $(b_1, b_2) \in \mathcal{B}_{12} \times \mathcal{B}_{21}$ and for any term $(\Delta(b_1) \geq x_i \vee \Delta(b_2) \geq y_{i+1})$ from $C(b_1, b_2)$, we define k_1 and k_2 such that $\Delta^{k_1}(b) = x_i$ and $\Delta^{k_2}(b) = y_{i+1}$. Then, since there is an edge $\{(b_1, k_1), (b_2, k_2)\} \in E$, $(b_1, k_1) \in D$ or $(b_2, k_2) \in D$. So $\alpha(b_1) \geq k_1$ or $\alpha(b_2) \geq k_2$. We get $\Delta^*(b_1) \geq x_i$ or $\Delta^*(b_2) \geq y_{i+1}$. □

Theorem 3. *The minimization of the buffers for two tasks is a polynomial problem.*

The construction of an instance $f(I)$ is polynomial. Moreover, the minimum weighted cover problem of a bipartite graph is solved polynomially by a flow algorithm. So, the theorem holds. $\Box$

4 Complexity for 3 tasks

In this section, we prove that the decision problem associated with the minimzation of the buffers is NP-complete in the strong sense for 3 tasks using a transformation from 3-SAT [2].

3-TASKS MIN WEIGHTED BUFFERS

- Instance: 3 tasks T_1, T_2, T_3, a set of buffers $\mathcal{B}$, an execution graphs G and a value K.
- Question: is there a feasible size function $\Delta : \mathcal{B} \rightarrow N$ such that $\sum_{i \in \mathcal{B}} w_b \Delta(b) \leq K$?

For $i \in I\!N^*$ and the buffers $(b_1, b_2, b_3) \in \mathcal{B}_{21} \times \mathcal{B}_{13} \times \mathcal{B}_{32}$ we consider 16 operations defined as follows :

- The operations are noted $o_1^{4(i-1)+1}$, $o_1^{4(i-1)+2}$, $o_1^{4(i-1)+3}$, $o_1^{4(i-1)+4}$, $o_2^{4(i-1)+1}$, $o_2^{4(i-1)+2}$, $o_2^{4(i-1)+3}$, $o_2^{4(i-1)+4}$, $o_3^{4(i-1)+1}$, $o_3^{4(i-1)+2}$, $o_3^{4(i-1)+3}$, $o_3^{4(i-1)+4}$
- Their associated buffer are :
    1. $b(o_1^{4(i-1)+1}) = b(o_1^{4(i-1)+2}) = b(o_3^{4(i-1)+3}) = b(o_3^{4(i-1)+4}) = b_2$
    2. $b(o_2^{4(i-1)+1}) = b(o_2^{4(i-1)+2}) = b(o_1^{4(i-1)+3}) = b(o_1^{4(i-1)+4}) = b_1$
    3. $b(o_3^{4(i-1)+1}) = b(o_3^{4(i-1)+2}) = b(o_2^{4(i-1)+3}) = b(o_2^{4(i-1)+4}) = b_3$

$g_i(b_1, b_2, b_3)(\infty)$ (*resp.*$g_i(b_1, b_2, b_3)(\Delta)$) is the associated precedence graph for an unlimited values on the size of the buffers (*resp.*. for a capacity function Δ).

Lemma 9. *Let us consider a capacity function $\Delta : \mathcal{B} \rightarrow N^*$ and the precedence graph $G(\Delta) = \cup_{i=1}^{p} g_i(\alpha_1^i, \alpha_2^i, \alpha_3^i)(\Delta)$ with $(\alpha_1^i, \alpha_2^i, \alpha_3^i) \in \mathcal{B}_{21} \times \mathcal{B}_{13} \times \mathcal{B}_{32}$. $G(\Delta)$ has no circuit if and only if $\Delta(\alpha_1^i) > 1 \vee \Delta(\alpha_2^i) > 1 \vee \Delta(\alpha_3^i) > 1$.*

Proof

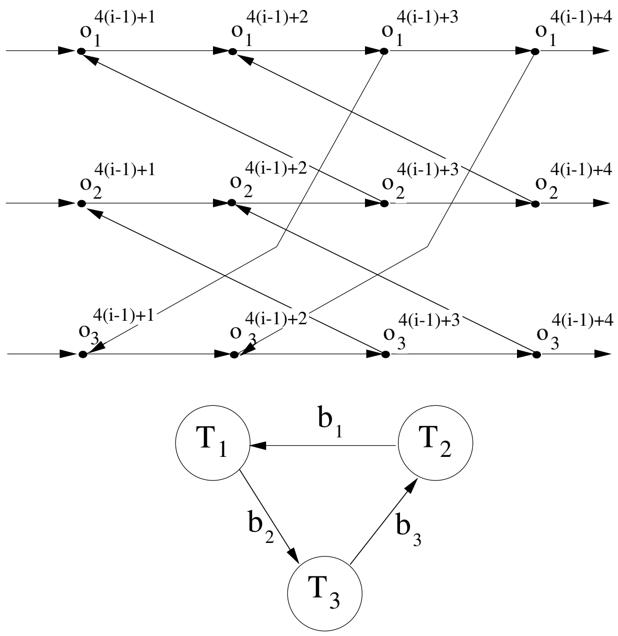

Figure 4: Graph $g_i(b_1, b_2, b_3)(\infty)$ and the Kahn subgraph for buffers b_1, b_2 and b_3

- $A \Rightarrow B$: if there exists $i \in \{1, \ldots, p\}$ such that $\Delta(\alpha_1^i) = \Delta(\alpha_2^i) = \Delta(\alpha_3^i) = 1$ then there is the circuit $c = (o_1^{4(i-1)+2}, o_1^{4(i-1)+3}, o_2^{4(i-1)+2}, o_2^{4(i-1)+3}, o_3^{4(i-1)+2}, o_3^{4(i-1)+3}, o_1^{4(i-1)+2})$ in $G(\Delta)$, so Δ is not feasible.
- $B \Rightarrow A$: Let us suppose now that Δ is not a feasible solution. Then, there is a circuit in $G(\Delta)$. A circuit in $G(\Delta)$ can't involve several subgraphs g_i, even if all the size of the buffers is set to 1. So, there exists $i \in \{1, \ldots, p\}$ such that $g_i(\alpha_1^i, \alpha_2^i, \alpha_3^i)(\Delta)$ has a circuit. The only way to get this circuit is to set $\Delta(\alpha_1^i) = \Delta(\alpha_2^i) = \Delta(\alpha_3^i) = 1$. $\square$

3-SAT

- Instance: set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$.
- Question: Is there a satisfying truth assignment of C ?

Let us consider an instance I of 3-SAT given by a set of of variables $U = \{u_1, \ldots, u_k\}$ and a set of clauses $C = \{C_1, \ldots, C_q\}$. We build and instance $f(I)$ of 3-TASKS MIN WEIGHTED BUFFERS as follows :

The set of buffers $\mathcal{B}$ is composed by 3 sets $\mathcal{B}_{21}$, $\mathcal{B}_{32}$ and $\mathcal{B}_{13}$ defined as follows :

- 3 buffers $b_1 \in \mathcal{B}_{21}$, $b_2 \in \mathcal{B}_{32}$ and $b_3 \in \mathcal{B}_{13}$ with $w_{b_1} = w_{b_2} = w_{b_3} = 3k+1$.

  The sizes of these buffers is important to oblige their capacity to be 1.
- For every variable $u_i \in U$, we associate 6 buffers $(b_i^1, \bar{b}_i^1) \in \mathcal{B}_{21}$, $(b_i^2, \bar{b}_i^2) \in \mathcal{B}_{32}$ and $(b_i^3, \bar{b}_i^3) \in \mathcal{B}_{13}$ with $w_{b_i^p} = w_{\bar{b}_i^p} = 1$ for $p = 1, 2, 3$.

The precedence graph $G(\infty)$ is defined as the union of a sequences of $6k + q$ graphs $g_i^*, i = 1, \ldots, 6k + q$ as follows :

- For every variable $u_i \in U, i = 1, \ldots, k$, we would like to express that every feasible capacity Δ verifies $\Delta(b_i^1) = \Delta(b_i^2) = \Delta(b_i^3) \in \{1, 2\}$ $\Delta(\bar{b}_i^1) = \Delta(\bar{b}_i^2) = \Delta(\bar{b}_i^3) \in \{1, 2\}$ and that these two values are different. For that, we build the following sequences of graphs :
  1. $g_{6(i-1)+1}^* = g_{6(i-1)+1}(b_i^1, \bar{b}_i^2, b_3)(\infty)$
  2. $g_{6(i-1)+2}^* = g_{6(i-1)+2}(b_i^1, b_2, \bar{b}_i^3)(\infty)$

3. $g^*_{6(i-1)+3} = g_{6(i-1)+3}(\bar{b}^1_i, b^2_i, b_3)(\infty)$

4. $g^*_{6(i-1)+4} = g_{6(i-1)+4}(\bar{b}^1_i, b_2, b^3_i)(\infty)$

5. $g^*_{6(i-1)+5} = g_{6(i-1)+5}(b_1, \bar{b}^2_i, b^3_i)(\infty)$

6. $g^*_{6(i-1)+6} = g_{6(i-1)+6}(b_1, b^2_i, \bar{b}^3_i)(\infty)$.

- For every clause $C_j = c^1_j \vee c^2_j \vee c^3_j$, $j = 1, \dots, q$, we associate a sub-graph $g^*_{6k+j} = g_{6k+j}(\alpha^1_j, \alpha^2_j, \alpha^3_j)$ with the buffers $\alpha^l_j, l = 1, 2, 3$ defined as follows : if $c^l_j = u_i$, we set $\alpha^l_j = b^l_i$. Otherwise, $c^l_j = \bar{u}_i$ and we set $\alpha^l_j = \bar{b}^l_i$.

The question is : is there a capacity function $\Delta : \mathcal{B} \to I\!N^*$ such that $\sum_{b \in \mathcal{B}} w_b \Delta(b) \leq 18k + 3$?

Lemma 10. *For every instance I of 3-SAT, the determination of $f(I)$ is polynomial.*

For every instance I with k variables and q clauses, the execution graph of the instance $f(I)$ has 3 tasks and $4(6k + q)$ operations. □

Lemma 11. *f is a polynomial tranformation from 3-SAT to 3-TASKS MIN WEIGHTED BUFFERS .*

Proof

Let us assume that the answer of an intance I of 3-SAT is "yes", then we build a solution to the corresponding instance $f(I)$ as follows :

- $\Delta(b_1) = \Delta(b_2) = \Delta(b_3) = 1$.
- Let us define a function $v : U \to \{0, 1\}$ as $v(u_i) = 1$ if u_i is true, 0 otherwise. Then, we set $\Delta(b^1_i) = \Delta(b^2_i) = \Delta(b^3_i) = 1 + v(u_i)$ and $\Delta(\bar{b}^1_i) = \Delta(\bar{b}^2_i) = \Delta(\bar{b}^3_i) = 2 - v(u_i)$.

Clearly, $\sum_{b \in \mathcal{B}} w_b \Delta(b) = 3(3k + 1) + 9k = 18k + 3$. For every variable $u_i \in U$, the 6 corresponding sub-graphs $g^*_{6(i-1)+1}, \dots g^*_{6(i-1)+6}$ have all at least one buffer which size is strictly greater than 1. Lastly, since each clause C_j is true, the associated graph g^*_{6k+j} has also at least one buffer with a size strictly greater than 1. By lemma 9, Δ is a solution of $f(I)$ and the answer to $f(I)$ is "yes".

Conversely, let us assume that Δ is a solution to $f(I)$. Then, we first prove that $\Delta(b_1) = \Delta(b_2) = \Delta(b_3) = 1$. Indeed, if $\Delta(b_1) \geq 2$, then

$$\sum_{b \in \mathcal{B}} w_b \Delta(b) \geq 4(3k+1) + 6k = 18k + 4$$

So, from the same reasonning with b_2 and b_3, we deduce that $\Delta(b_1) = \Delta(b_2) = \Delta(b_3) = 1$.

For every variable u_i, the graphs $g^*_{6(i-1)+\alpha}, \alpha = 1, \ldots 6$ have all at least one buffer whose capacity is strictly superior than 2. So, by lemma 9 we get necessarily : $\Delta(b_i^1) + \Delta(\bar{b}_i^2) \geq 3$, $\Delta(b_i^1) + \Delta(\bar{b}_i^3) \geq 3$, $\Delta(\bar{b}_i^1) + \Delta(b_i^2) \geq 3$, $\Delta(\bar{b}_i^1) + \Delta(b_i^3) \geq 3$, $\Delta(\bar{b}_i^2) + \Delta(b_i^3) \geq 3$, and $\Delta(b_i^2) + \Delta(\bar{b}_i^3) \geq 3$. By summing these inequalities, we get :

$$\Delta(b_i^1) + \Delta(\bar{b}_i^1) + \Delta(b_i^2) + \Delta(\bar{b}_i^2) + \Delta(b_i^3) + \Delta(\bar{b}_i^3) \geq 9$$

Now, $18k + 3 - w_{b_1}\Delta(b_1) - w_{b_2}\Delta(b_2) - w_{b_3}\Delta(b_3) = 9k$. So, we have $\Delta(b_i^1) + \Delta(\bar{b}_i^1) + \Delta(b_i^2) + \Delta(\bar{b}_i^2) + \Delta(b_i^3) + \Delta(\bar{b}_i^3) = 9$ for every variable u_i.

Now, by contradiction, let us suppose that there is a variable $u_i \in U$ with $\Delta(b_i^1) = \Delta(\bar{b}_i^2) = 2$. Then, since there is no circuit in $g^*_{6(i-1)+3}$, $g^*_{6(i-1)+4}$ and $g^*_{6(i-1)+6}$ we get :

$$\Delta(\bar{b}_i^1) \geq 2 \vee \Delta(b_i^2) \geq 2$$
$$\Delta(\bar{b}_i^1) \geq 2 \vee \Delta(b_i^3) \geq 2$$
$$\Delta(b_i^2) \geq 2 \vee \Delta(\bar{b}_i^3) \geq 2$$

A minimum solution is $\Delta(\bar{b}_i^1) = \Delta(b_i^2) = 2$ and $\Delta(b_i^3) = \Delta(\bar{b}_i^3) = 1$. So,

$$\Delta(b_i^1) + \Delta(\bar{b}_i^1) + \Delta(b_i^2) + \Delta(\bar{b}_i^2) + \Delta(b_i^3) + \Delta(\bar{b}_i^3) \geq 10$$

which is impossible. So, $\Delta(b_i^1) = \Delta(b_i^2) = \Delta(b_i^3)$ and $\Delta(\bar{b}_i^1) = \Delta(\bar{b}_i^2) = \Delta(\bar{b}_i^3)$.

We can set $u_i = (\Delta(b_i^1) \geq 2)$. Notice that the equality $\bar{u}_i = (\Delta(\bar{b}_i^1) \geq 2)$ is true. We prove now that every clause $C_j \in C$ is true. Indeed, by lemma 9, the buffers $\alpha_j^p, p = 1, 2, 3$ verify

$$\Delta(\alpha_j^1) + \Delta(\alpha_j^2) + \Delta(\alpha_j^3) \geq 4$$

So, $\exists p \in \{1, 2, 3\}$ such that $\Delta(\alpha_j^p) \geq 2$. Now, if $c_j^p = u_i$, then $\alpha_j^p = b_i^p$. So, u_i is true and C_j is true. Else, $c_j^p = \bar{u}_i$ and then $\alpha_j^p = \bar{b}_i^p$. So, $\bar{u}_i$ is true, and C_j is true.

So, the answer for the instance I of 3-SAT is "yes". $\square$

Theorem 4. *3-TASKS MIN WEIGHTED BUFFERS is NP-complete in the strong sense for* 3 *tasks.*

5 Conclusion

We proved in this paper that the communication with buffers can be modeled using a precedence graph and two complexity results on the problem. Further research will consists on develop exact and approximate methods to solve the efficiently the general problem.

Acknowlegments We wish to thanks Frederic Pétrot and Emmanuelle Encrenaz for their help in the project LIP6-Buffalo.

References

[1] R.K Ahuja, T.L Magnanti, and J.B Orlin. *Network Flows.* Prentice Hall, 1993.

[2] M Garey and D Johnson. *Computers and Intractability.* Freeman, 1979.

[3] D Hommais. *Une méthode d'évaluation et de synthèse des communications dans les systèmes intégrés matériel-logiciel.* PhD thesis, Université Paris 6, 2001.

[4] G Kahn and D.B MacQueen. Coroutines and networks of parallel processors. In *Info. Proc.*, pages 993–998, 1977.

[5] T Parks. *Bounded Scheduling of Process Netwotks.* PhD thesis, Princeton University, 1995.