



Time Separation of Events: An Inverse Method

Emmanuelle Encrenaz¹ Laurent Fribourg^{a,2}

^a *LSV, ENS de Cachan & CNRS
Paris, France*

Abstract

The problem of “time separation” can be stated as follows: Given a system made of several connected components, each one entailing a local delay known with uncertainty, what is the maximum time for traversing the global system? This problem is useful, e.g. in the domain of digital circuits, for determining the global traversal time of a signal from the knowledge of bounds on the component propagation delays. The uncertainty on each component delay is given under the form of an interval. The general problem is NP-complete. We focus here on the *inverse problem*: we seek intervals for component delays for which the global traversal time is guaranteed to be no greater than a specified maximum. We give a polynomial time method to solve it. As a typical application, we show how to use the method in order to relax some specified local delays while preserving the maximum traversal time. This is especially useful, in the area of digital circuits, for optimizing “setup” timings of input signals (minimum timings required for stability).

Keywords: Execution time analysis, Timing constraints, Time separation problem, Inverse method

1 Introduction

As said in [5]: “The behavior of asynchronous and concurrent systems is naturally described in terms of events and their interactions. A fundamental problem in analyzing such systems is to determine bounds on the time separation of events. Stated informally, we seek answers to questions such as: “How late can event i occur after event j ?” for arbitrary events i and j . The problem of computing time separation bounds is compounded in practice by statistical variations in manufacturing and operating conditions that introduce uncertainties in component delays. Consequently, finding bounds on time separation of events in the presence of uncertain component delays is an important practical problem.”

The uncertainty on each component delay is given under the form of an interval. The general problem of finding the exact bound on the time separation between two

¹ Email: emmanuelle.encrenaz@lsv.ens-cachan.fr

² Email: fribourg@lsv.ens-cachan.fr

given events i and j is NP-complete ([12]). We focus here on the *inverse problem*: we seek intervals for the component delays, so that the time separation between i and j is guaranteed to be no greater than a specified bound. We give a polynomial time method to solve it, and explain how it is useful for relaxing some bounds associated to some component delays while preserving the global separation time from i to j .

Related Work.

The direct problem of time separation of events has received considerable attention in the literature (see, e.g. [5] for an extensive survey). Even in the case of *acyclic* timing constraint graphs, the direct problem is NP-complete [12]. Many researchers have thus proposed polynomial-time *approximating* algorithms, which give an upper approximation of the maximal separation time. In contrast here, the time generated will be guaranteed to be the *exact* maximal time (for a possibly restricted domain of component delays, however). Other researchers have replaced some of the bound values by *parameters*, then have computed the exact solution space using exponential-time procedures (e.g., Fourier-Motzkin elimination or Presburger-based procedures [2])). A variant of this approach has been proposed using additional techniques of abstract interpretation [9] or parametric reachability analysis [8].

In this paper we focus on the *inverse problem*. To our knowledge, such an inverse problem has been rarely tackled in the literature. An exception is [6]. Its goal is there to compute *safe* bounds on some timing constraints (typically, setup timings), in the sense that, when satisfied, these constraints guarantee “correct” operation of the circuit. In their context, “correct” means that the circuit has no *hazard*, i.e., roughly speaking, that the output signal changes at most once. Such a guarantee is complementary to ours, since what we guarantee, in this context, is that the first change of the output signal occurs before a specified lapse of time. Also, their analysis does not take place in the framework of separation of events, but in the framework of “multi-value signal algebra”, which is well-suited to the problem of hazard-detection. Another work, which can be viewed as an inverse method, is by [4]: they pose the problem as a *min-max linear programming* problem and use it for computing optimal clock schedules in synchronous circuits, but their method is exponential.

2 Time Separation of Events

The Direct Problem.

Formally, the system can be represented under the form of a “timing constraint graph” [7]. The graph is oriented: vertices (or nodes) represent events and directed edges represent causal dependencies between them. Let $\{0, \dots, N - 1\}$ be the set of vertices. Each event is labeled with a *min*, *max* or *delay* operator specifying how the time of occurrence of event q , denoted t_q , depends on those of its predecessors in the timing constraint graph. A delay node q is either a source event (no incoming edge), or has a unique predecessor, say p , in the graph, and the edge from event p

to event q is labeled with symbol $\delta_{p,q}$ representing the (non-negative) delay in the propagation of event p to event q . For any vertex q , let $\text{preds}(q)$ denote the set of its predecessor vertices. We have the system³:

- for each delay-node q , distinct from the source, of (unique) predecessor p :

$$t_q = t_p + \delta_{p,q} \quad (1)$$

- for each min-node q :

$$t_q = \min_{p \in \text{preds}(q)} t_p \quad (2)$$

- for each max-node q :

$$t_q = \max_{p \in \text{preds}(q)} t_p \quad (3)$$

In the following, we restrict our analysis to *acyclic* timing constraint graphs, with a unique source (event with no incoming edge). The *delay-edges* are the edges linking a node p to a delay-node q . Note that, for a given vector of values \bar{d} for $\bar{\delta}$, system (1)-(2)-(3) has a unique solution in t_0, \dots, t_{N-1} .

For all delay-edge from p to q , the associated delay $\delta_{p,q}$ is constrained to fall within fixed (nonnegative) lower and upper bounds $l_{p,q}$ and $u_{p,q}$. That is:

$$l_{p,q} \leq \delta_{p,q} \leq u_{p,q}.$$

Let $\bar{\delta}$ be the vector of the $\delta_{p,q}$ s, where q is a delay-node (distinct from the source). Let \bar{l} and \bar{u} be the vectors of $l_{p,q}$ s and $u_{p,q}$ s respectively. Note that all these vectors have $r-1$ components, where r is the number of delay-nodes (including the source).⁴ Let $\Delta = (\bar{l}, \bar{u})$ be the zone delimited by intervals $(l_{p,q}, u_{p,q})$. Henceforth, the set of inequations $\{l_{p,q} \leq \delta_{p,q} \leq u_{p,q}\}$ will appear under the form $\bar{l} \leq \bar{\delta} \leq \bar{u}$ or $\bar{\delta} \in \Delta$. The *direct problem* is posed as one of finding the maximum achievable *separation* $\max(t_j - t_i)$ between two events i and j under the considered system of timing constraints. We will assume here that the time of source event is null, and coincides with vertex i ($t_i = 0$). In the direct problem, one seeks therefore the maximal value of t_j , denoted by MAX_j :

$$MAX_j = \max_{\bar{\delta} \in (\bar{l}, \bar{u})} t_j.$$

As usual, a *path* π_p from the source to event p is a sequence of adjacent edges going from the source to p . The *delay of a path* π is the sum of the delays of its edge components, $\sum_{(p,q) \in \pi} \delta_{p,q}$, which will be abbreviated as $\sum_{\pi} \bar{\delta}$.

Henceforth, we assume given a Time Separation of Events system with an acyclic timing constraint graph with a unique source (node i), a set of vertices $\{0, \dots, N-1\}$, a domain $\Delta = (\bar{l}, \bar{u})$, and an associated set of equations (1)-(2)-(3) (together with implicit equation $t_i = 0$).

Example 2.1 This example concerns SPSMALL, an embedded memory designed by STMicroelectronics which has been presented in [8]. The constraint timing graph

³ In the original formulation of the problem, there is no delay-nodes, but one allows min-nodes of the form $t_q = \min(t_p + \delta_{p,q}, t_r + \delta_{r,q})$ (and similarly for max-nodes). Such a timing constraint graph can be transformed into an equivalent form as above, at the price of introducing new intermediate nodes on each edge. This adds to the original graph at most $n \times K$ new vertices, where n is the number of vertices of the original graph, and K the maximum number of predecessors of any vertex. (Usually K is a small constant, and in any case bounded by n .)

⁴ It is convenient to assume that all the other edges (to min/max nodes q) have delays $\delta_{p,q}$ which are null (i.e., $l_{p,q} = u_{p,q} = 0$ for min/max node q).

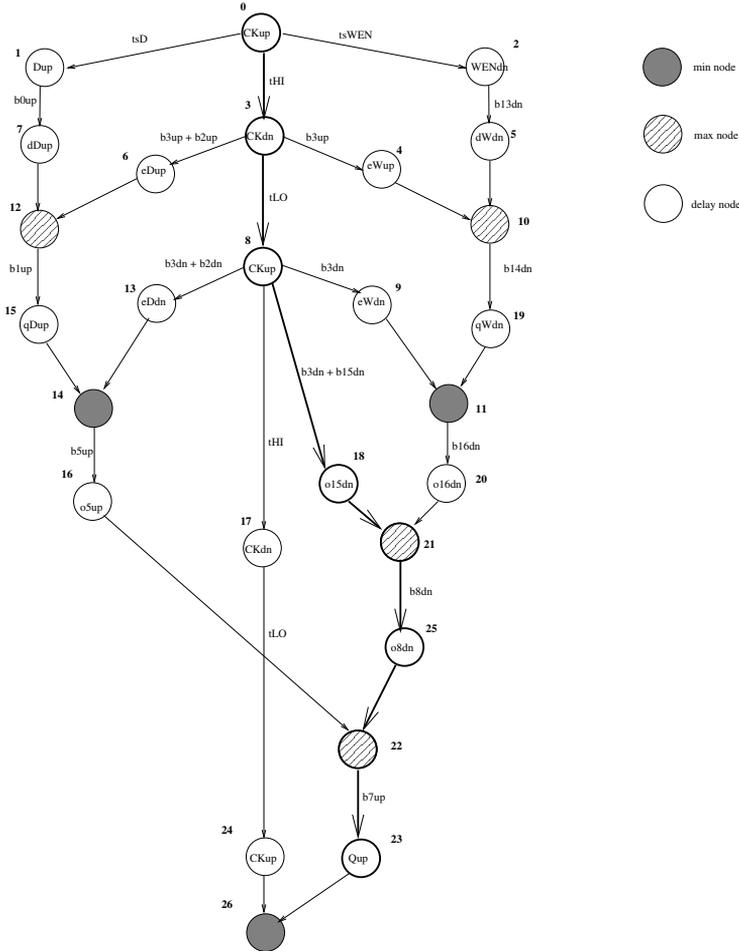


Fig. 1. Timing Constraint Graph for SPSMALL ($\delta_{0,1}$ stands for $\delta_{HI} + \delta_{LO} - \delta_{setup_D}$, and $\delta_{0,2}$ stands for $\delta_{HI} + \delta_{LO} - \delta_{setup_W}$)

(for the write operation) is made of 27 event nodes (see Fig. 1). For the sake of homogeneity with presentation of [8], we proceed to cosmetic changes with respect to the presentation of timing constraint graphs given above. Thus, events are not only numbered but also denoted by symbols corresponding to those used in [8]; for example event 20 is also denoted by σ_{16}^\dagger .⁵ Also, delays $\delta_{p,q}$ between events p and q are allowed to appear as an algebraic sum of more elementary delays: for example, $\delta_{0,1}$ corresponds to $\delta_{HI} + \delta_{LO} - \delta_{setup_D}$, $\delta_{3,6}$ to $\delta_3^\dagger + \delta_2^\dagger, \dots$. The separation time of interest is here $t_{Q^\dagger} (\equiv t_{23})$. The delay δ_{HI} (resp. δ_{LO}) corresponds to the high-level (resp. low-level) period of the clock of the circuit. These delays are specific, as they are assumed to be fixed: the bounds l_{HI} and u_{HI} of δ_{HI} are equal to a common value denoted by d_{HI} , and similarly for δ_{LO} . Henceforth, δ_{HI} and δ_{LO} will appear under the form d_{HI} and d_{LO} respectively. The expression $\bar{\delta} \in (\bar{l}, \bar{u})$ (where \bar{l} and \bar{u} correspond to implementation SP1 of [8]) is given componentwise as follows:

⁵ The symbols \dagger and \downarrow indicate the rising or falling nature of the edges of the corresponding signal.

$\delta_0^\uparrow \in (94, 95)$, $\delta_0^\downarrow \in (65, 66)$, $\delta_1^\uparrow \in (13, 14)$, $\delta_1^\downarrow \in (17, 18)$, $\delta_2^\uparrow \in (23, 24)$, $\delta_2^\downarrow \in (29, 30)$, $\delta_3^\uparrow \in (5, 5)$, $\delta_3^\downarrow \in (2, 2)$, $\delta_5^\uparrow \in (21, 22)$, $\delta_5^\downarrow \in (44, 45)$, $\delta_7^\uparrow \in (20, 21)$, $\delta_7^\downarrow \in (19, 20)$, $\delta_8^\uparrow \in (0, 0)$, $\delta_8^\downarrow \in (21, 22)$, $\delta_{13}^\uparrow \in (10, 11)$, $\delta_{13}^\downarrow \in (8, 8)$, $\delta_{14}^\uparrow \in (21, 22)$, $\delta_{14}^\downarrow \in (21, 22)$, $\delta_{15}^\uparrow \in (13, 14)$, $\delta_{15}^\downarrow \in (10, 11)$, $\delta_{16}^\uparrow \in (23, 24)$, $\delta_{16}^\downarrow \in (0, 0)$, $d_{HI} = 36$, $d_{LO} = 74$, $\delta_{setup_D} \in (108, ?)$, $\delta_{setup_W} \in (48, ?)$.⁶

Besides: $MAX_{23} \equiv MAX_{Q\uparrow} = 56$ ($+d_{HI} + d_{LO}$). All these values have been found by the manufacturer via electrical simulation (at the transistor level). Decreasing values $l_{setup_D} = 108$ and $l_{setup_W} = 48$ would lower the cost of the circuit, but this should not alter the maximal response time $MAX_{23} = 56$ (see [8]). Our inverse method will show that this can be safely done.

Interest of the Inverse Problem.

Before giving the inverse method, let us explain why the resolution of the inverse problem can help to *relax* the bounds assigned to the component delays of the system.

In the direct problem, we have:

- as an input: $\Delta = (\bar{l}, \bar{u})$,
- as an output: MAX_j , a value such that $t_j \leq MAX_j$ for all $\bar{\delta} \in \Delta$.

In the inverse problem, we will have:

- as an input: a specific value \bar{d}^0 for $\bar{\delta}$,
- as outputs: a domain $\Delta^* = (\bar{l}^*, \bar{u}^*)$, and a value MAX_j^* , such that $t_j \leq MAX_j^*$ for all $\bar{\delta} \in \Delta^*$.

In a “good situation”, we will have for MAX_j^* and Δ^* the following:

1. $MAX_j^* \leq MAX_j$,
2. $\Delta^* = (\bar{l}^*, \bar{u}^*) \supset \Delta = (\bar{l}, \bar{u})$ (i.e: $\bar{l}^* \leq \bar{l}$ and $\bar{u} \leq \bar{u}^*$).

In this case, we can safely *relax* the original bounds \bar{l} and \bar{u} to \bar{l}^* and \bar{u}^* , and still guarantee an upper bound (no greater than) MAX_j . As illustrated in Example 1, the method is useful, in the area of digital circuits, for optimizing “setup” timings of input signals (i.e., for minimizing the stability period of an input signal required before the circuit clock changes). Even if the second item does not hold for all its components, i.e.: $l_k^* > l_k$ or $u_k > u_k^*$ for some events k , the knowledge of Δ^* may be instructive: it allows in particular to identify certain sets of “key” parameters k , which, when ranging over a restricted interval, enables the relaxation of the bounds of the other parameters (see Appendix 2).

3 Inverse Method

Let us now describe our inverse method.

Definition 3.1 We say that \mathcal{O} is a *canonical order on the events* if \mathcal{O} is of the form $t_{k_1} \leq \dots \leq t_{k_r}$, where k_1, \dots, k_r are the delay-nodes of the timing constraint graph, and k_1 corresponds to source node i (hence: $t_{k_1} = 0$).

⁶ The maximal values of δ_{setup_D} and δ_{setup_W} are unspecified. (The values of interest are the minimal values, which correspond to *optimal* behavior of the circuit.)

Given two delay-nodes p_1 and p_2 , we say that p_1 is \mathcal{O} -earlier (resp. \mathcal{O} -later) than p_2 if $t_{p_1} \leq t_{p_2}$ (resp. $t_{p_2} \leq t_{p_1}$) according to \mathcal{O} .

Definition 3.2 Let $\mathcal{O} \equiv t_{k_1} \leq \dots \leq t_{k_r}$ be a canonical order. For all event q , the representative delay-node of q (or the delay-node representing q) under \mathcal{O} , denoted by q^* , is defined recursively as follows:

- if q is a delay-node then $q^* = q$.
- if q is a min-node (resp. max-node) then q^* is the \mathcal{O} -earliest (resp. \mathcal{O} -latest) node among the delay-nodes representing the predecessors of q (i.e. among $\{p^* \mid p \in \text{preds}(q)\}$).

For all event q of representative delay-node q^* , it follows from (1)-(2)-(3) that $t_q = t_{q^*}$.

Consider a vector of (nonnegative real) values \bar{d} for the delay-edges. When \bar{d} is assigned to $\bar{\delta}$, each time t_p ($0 \leq p \leq N - 1$) takes a unique value, say D_p , according to (1)-(2)-(3). Let us order the delay-nodes by non-decreasing values of D (in case there are two delay-nodes p and q with $D_p = D_q$, we order them in an arbitrary way). This yields a non-decreasing sequence of the form $t_{k_1} \leq t_{k_2} \leq \dots \leq t_{k_r}$. Such an order is said to be a canonical order induced by \bar{d} .

Example 3.3 Consider Example 2.1. As for particular value \bar{d}^0 of $\bar{\delta}$, we take the maximal values \bar{u} of the specification. (Note that, for $\delta_{0,1}$ and $\delta_{0,2}$, this corresponds to the minimal values of δ_{setup_D} and δ_{setup_W} respectively, since $\delta_{0,1} = d_{HI} + d_{LO} - \delta_{\text{setup}_D}$, and similarly for $\delta_{0,2}$.) For \bar{d}^0 , the time of each event is: $t_0 = 0$, $t_1 = 2$, $t_3 = 36$, $t_4 = 41$, $t_2 = 62$, $t_6 = 65$, $t_5 = t_{10} = 70$, $t_{11} = t_{19} = t_{20} = 92$, $t_7 = t_{12} = 97$, $t_8 = 110$, $t_{14} = t_{15} = 111$, $t_9 = 112$, $t_{18} = t_{21} = 123$, $t_{16} = 133$, $t_{13} = 142$, $t_{22} = t_{25} = 145$, $t_{17} = 146$, $t_{23} = 166$, $t_{24} = t_{26} = 220$. This induces the following order \mathcal{O}_0 :

$$t_0 \leq t_1 \leq t_3 \leq t_4 \leq t_2 \leq t_6 \leq t_5 \leq t_{19} \leq t_{20} \leq t_7 \leq t_8 \leq t_{15} \leq t_9 \leq t_{18} \leq t_{16} \leq t_{13} \leq t_{25} \leq t_{17} \leq t_{23} \leq t_{24}.$$

The representative nodes q^* of min/max nodes q under \mathcal{O}_0 are given by:

$$10^* = 5, 11^* = 19, 12^* = 7, 14^* = 15, 21^* = 18, 22^* = 25 \text{ and } 26^* = 23.$$

Definition 3.4 Let \mathcal{O} be a canonical order of the form $t_{k_1} \leq \dots \leq t_{k_r}$. For all event q , the representative path to q associated to \mathcal{O} , and denoted by π_q^* , is recursively defined as follows:

- if q is the source node, then π_q^* is the empty path.
- if q is a delay-node (distinct from the source) then $\pi_q^* = \pi_p^* \cup (p, q)$, where p is the unique predecessor of q , and π_p^* the representative path of p for \mathcal{O} .
- if q is a min-node (resp. max-node) then $\pi_q^* = \pi_{q^*}^*$, where q^* is the delay-node representing q .

Example 3.5 For ordering \mathcal{O}_0 (see Example 3.3), the representative path π_{23}^* to event 23 ($\equiv Q^\dagger$) is $\{(0, 3), (3, 8)(8, 18)(21, 25), (22, 23)\}$, whose support is drawn in bold font on Fig. 1.

Definition 3.6 Let $\mathcal{O} \equiv t_{k_1} \leq \dots \leq t_{k_r}$ be a canonical order for the considered

system, and $\pi_{k_s}^*$ the representative path of k_s ($1 \leq s \leq r$). We call δ -constraints associated to \mathcal{O} the set $\mathcal{I}^* = \{I_{k_1}, \dots, I_{k_r}\}$ with, for all $1 \leq s \leq r$:

$$I_{k_s} \equiv \sum_{\pi'_s} \bar{\delta} \leq \sum_{\pi'_{s+1}} \bar{\delta},$$

where π'_s and π'_{s+1} are the disjoint parts of $\pi_{k_s}^*$ and $\pi_{k_{s+1}}^*$ respectively (i.e. $\pi'_s = \pi_{k_s}^* \setminus \pi_{k_{s+1}}^*$ and $\pi'_{s+1} = \pi_{k_{s+1}}^* \setminus \pi_{k_s}^*$).

Example 3.7 Consider in \mathcal{O}_0 inequation $t_{17} \leq t_{23}$, and let us give the associated inequality, say I_{18} . For path π_{23}^* associated to event 23, we have (see Example 3.5):

$$\sum_{\pi_{23}^*} \bar{\delta} = d_{HI} + d_{LO} + \delta_3^\downarrow + \delta_{15}^\downarrow + \delta_8^\downarrow + \delta_7^\uparrow.$$

Likewise, for event 17 $\equiv ck^\downarrow$, the representative path π_{17}^* goes from 0 to 17 through vertices 3 and 8. We have:

$$\sum_{\pi_{17}^*} \bar{\delta} = 2d_{HI} + d_{LO}.$$

Therefore $t_{17} \leq t_{23}$ corresponds (after elimination of the delay $d_{HI} + d_{LO}$ associated to the common sub-path of π_{17}^* and π_{23}^*) to

$$I_{18} \equiv \sum_{\pi_{17}^*} \bar{\delta} \leq \sum_{\pi_{23}^*} \bar{\delta}, \text{ i.e.: } d_{HI} \leq \delta_3^\downarrow + \delta_{15}^\downarrow + \delta_8^\downarrow + \delta_7^\uparrow.$$

The other inequalities of \mathcal{I}_0^* associated to \mathcal{O}_0 are given in Appendix 1.

Proposition 3.8 Let $\mathcal{O} \equiv t_{k_1} \leq \dots \leq t_{k_r}$ be a canonical order, π_p^* the representative path of event p ($0 \leq p \leq N - 1$) associated to \mathcal{O} , and $\mathcal{I}^* \equiv \{I_{k_1}, \dots, I_{k_r}\}$ the set of δ -constraints associated to \mathcal{O} .

If \bar{d} is a value of $\bar{\delta}$ satisfying \mathcal{I}^* , then we have (under (1)-(2)-(3)), for $\bar{\delta} = \bar{d}$:

$$t_p = \sum_{\pi_p^*} \bar{d}, \text{ for all event } p.$$

Proof. Suppose that \bar{d} is a value of $\bar{\delta}$ satisfying \mathcal{I}^* . Let us prove: $t_p = \sum_{\pi_p^*} \bar{d}$, for all event p . The proof proceeds by induction on the depth of event p in the timing constraint graph. The proof of the base case is trivial. Let us prove the induction step. Node p is either a min/max node or a delay-node. Suppose first that p is a min/max-node, say a min-node (the case of a max-node is similar). Then p is the sink of a “min/max subgraph”, i.e. the smallest (directed) subgraph of sink p having only delay-nodes, say p_1, \dots, p_m , as sources (All the non-source nodes are min/max nodes). By induction hypothesis on p_k ($1 \leq k \leq m$), we have: $t_{p_k} = \sum_{\pi_{p_k}^*} \bar{d}$. Furthermore, since \bar{d} is a solution of \mathcal{I}^* , the values $(\sum_{\pi_{p_k}^*} \bar{d})_{1 \leq k \leq m}$ are ordered as $(t_{p_k})_{1 \leq k \leq m}$ in \mathcal{O} . It follows that all the nodes v of the min/max subgraph are such that $t_v = \sum_{\pi_v^*} \bar{d}$. In particular for the sink p : $t_p = \sum_{\pi_p^*} \bar{d}$.

Suppose now that p is a delay-node. Then p has p_1 as a unique predecessor and $t_p = \delta_{p_1,p} + t_{p_1}$. Now, by induction hypothesis, $t_{p_1} = \sum_{\pi_{p_1}^*} \bar{d}$. Therefore: $t_p = \delta_{p_1,p} + \sum_{\pi_{p_1}^*} \bar{d} = \sum_{\pi_p^*} \bar{d}$.

In both cases, $t_p = \sum_{\pi_p^*} \bar{d}$. This achieves the proof of the induction step. \square

Definition 3.9 Let $\mathcal{O} \equiv t_{k_1} \leq \dots \leq t_{k_r}$ be a canonical order, and $\pi_{k_s}^*$ the representative path of π_{k_s} ($1 \leq s \leq r$). Let $\mathcal{I}^* \equiv \{I_{k_1}, \dots, I_{k_r}\}$, be the set of δ -constraints associated to \mathcal{O} . We call $y \leq x$ -constraints associated to \mathcal{O} the set $\mathcal{J}^* = \{J_{k_1}, \dots, J_{k_r}\}$, where J_{k_s} ($1 \leq s \leq r$) is obtained by replacing every $\delta_{p,q}$ of the left (resp. right) side of inequality I_{k_s} by a new variable $y_{p,q}$ (resp. $x_{p,q}$), i.e.:

$$J_{k_s} \equiv \sum \pi'_s \bar{y} \leq \sum \pi'_{s+1} \bar{x},$$

where π'_s and π'_{s+1} are the disjoint parts of $\pi_{k_s}^*$ and $\pi_{k_{s+1}}^*$ respectively.

Example 3.10 Let us consider \mathcal{I}_0^* associated to the ordering \mathcal{O}_0 for SPSMALL Example (see Appendix 1). By substituting each $\delta_{p,q}$ of the left (resp. right) side of \mathcal{I}_0^* by $y_{p,q}$ (resp. $x_{p,q}$), we obtain the following system \mathcal{J}_0^* (after having discarded trivial inequalities $0 \leq \delta_3^\uparrow$ and $0 \leq \delta_k^\downarrow$ for $k = 14, 15, 16$):

1. $y_{setup_D} \leq d_{HI} + d_{LO}$
2. $d_{LO} \leq x_{setup_D}$
3. $y_3^\uparrow + y_{setup_W} \leq d_{LO}$
4. $d_{LO} \leq x_3^\uparrow + x_2^\uparrow + x_{setup_W}$
5. $y_3^\uparrow + y_2^\uparrow + y_{setup_W} \leq d_{LO} + x_{13}^\downarrow$
6. $y_{13}^\downarrow + y_{14}^\downarrow + y_{16}^\downarrow + y_{setup_D} \leq x_{setup_W} + x_0^\uparrow$
7. $y_0^\uparrow \leq x_{setup_D}$
8. $y_{setup_D} \leq x_0^\uparrow + x_1^\uparrow$
9. $y_0^\uparrow + y_1^\uparrow \leq x_{setup_D} + x_3^\downarrow$
10. $y_3^\downarrow + y_{15}^\downarrow + y_{setup_D} \leq x_0^\uparrow + x_1^\uparrow + x_5^\uparrow$
11. $y_0^\uparrow + y_1^\uparrow + y_5^\uparrow \leq x_3^\downarrow + x_2^\downarrow + x_{setup_D}$
12. $y_2^\downarrow \leq x_{15}^\downarrow + x_8^\downarrow$
13. $y_3^\downarrow + y_{15}^\downarrow + y_8^\downarrow \leq d_{HI}$
14. $d_{HI} \leq x_3^\downarrow + x_{15}^\downarrow + x_8^\downarrow + x_7^\uparrow$
15. $y_3^\downarrow + y_{15}^\downarrow + y_8^\downarrow + y_7^\uparrow \leq d_{HI} + d_{LO}$

Theorem 3.11 Let \mathcal{O} be a canonical order of events, π_j^* the representative path of event j , and \mathcal{J}^* the associated set of $y \leq x$ -constraints.

Let \bar{l}^* and \bar{u}^* be solutions of \mathcal{J}^* for \bar{x} and \bar{y} respectively, and let $\Delta^* = (\bar{l}^*, \bar{u}^*)$.

Then, under (1)-(2)-(3), the representative path π_j^* is such that:

$$MAX_j^* \equiv \max_{\bar{\delta} \in \Delta^*} t_j = \sum \pi_j^* \bar{u}^*.$$

Proof. Since \bar{l}^* and \bar{u}^* are solutions of \mathcal{J}^* , we have: $\sum \pi'_{k_s} \bar{u}^* \leq \sum \pi'_{k_{s+1}} \bar{l}^*$, for all $1 \leq s \leq r$. Hence, for all \bar{d} with $\bar{l}^* \leq \bar{d} \leq \bar{u}^*$: $\sum \pi'_{k_s} \bar{d} \leq \sum \pi'_{k_{s+1}} \bar{d}$. It follows that, for all \bar{d} with $\bar{l}^* \leq \bar{d} \leq \bar{u}^*$, \bar{d} is a solution of \mathcal{I}^* . Therefore, under (1)-(2)-(3), for all event p associated to $\bar{\delta} = \bar{d}$, we have by Prop. 3.8: $t_p = \sum \pi_p^* \bar{d} \leq \sum \pi_p^* \bar{u}^*$. In particular, for $\bar{\delta} = \bar{u}^*$, we have: $t_p = \sum \pi_p^* \bar{u}^*$. It follows: $MAX_p^* \equiv \max_{\bar{\delta} \in \Delta^*} t_p = \sum \pi_p^* \bar{u}^*$. This is true in particular for $p = j$. □

The system \mathcal{J}^* is a linear system of r inequalities with r unknowns $x_{p,q}$ and r unknowns $y_{p,q}$, where r is the number of delay-nodes (r is less than or equal to the number N of events). The problem now reduces to find solutions \bar{l}^* and \bar{u}^* which delimit a domain of interest Δ^* as large as possible. This problem can be seen as an optimization problem in linear programming, which is solvable in polynomial time in N .

Example 3.12 Let us consider system \mathcal{J}^* generated for SPSMALL system (see Example 3.10). In order to find optimal values for $x_{setup_D}, y_{setup_D}, x_{setup_W}, y_{setup_W}$,

we instantiate the other components of \bar{x} and \bar{y} with the corresponding values of \bar{l} and \bar{u} respectively (see Example 2.1). In other terms, we have $\bar{l}^* = \bar{l}$ and $\bar{u}^* = \bar{u}$, for each component distinct from $setup_D$ and $setup_W$. \mathcal{J}^* then reduces to:

$$\begin{aligned}
 y_{setup_D} &\leq 110 && \text{(from: } y_{setup_D} \leq d_{HI} + d_{LO}) \\
 107 &\leq x_{setup_D} && \text{(from: } y_0^\uparrow + y_1^\uparrow \leq x_{setup_D} + x_3^\downarrow) \\
 y_{setup_W} &\leq 53 && \text{(from: } y_3^\uparrow + y_2^\uparrow + y_{setup_W} \leq d_{LO} + x_{13}^\downarrow) \\
 46 &\leq x_{setup_W} && \text{(from: } d_{LO} \leq x_3^\uparrow + x_2^\uparrow + x_{setup_W}) \\
 y_{setup_D} &\leq x_{setup_W} + 64 && \text{(from: } y_{13}^\downarrow + y_{14}^\downarrow + y_{16}^\downarrow + y_{setup_D} \leq x_{setup_W} + x_0^\uparrow)
 \end{aligned}$$

Therefore, optimal (i.e., least) solutions are:

$$l_{setup_D}^* = 107, \quad u_{setup_D}^* = 110, \quad l_{setup_W}^* = 46, \quad u_{setup_W}^* = 53.$$

On Δ^* , we have: $MAX_{Q^\uparrow}^* \equiv \max_{\bar{\delta} \in \Delta^*} t_{Q^\uparrow} = (d_{HI} + d_{LO} +) u_3^\downarrow + u_{15}^\downarrow + u_8^\downarrow + u_7^\uparrow$.

This gives: $MAX_{Q^\uparrow}^* = 56$ ($+d_{HI} + d_{LO}$). Therefore: $MAX_{Q^\uparrow}^* = MAX_{Q^\uparrow}$, where MAX_{Q^\uparrow} is the original value of the specification. Besides, the value 107 of $l_{setup_D}^*$ is (slightly) relaxed with respect to the value $l_{setup_D} = 108$ of the specification, as well as the value 46 of $l_{setup_W}^*$ with respect to $l_{setup_W} = 48$. Therefore $\Delta^* \supset \Delta$. We are thus in the “good situation” described in Sect. 4. The lower bound $l_{setup_D}^*$ can however be further optimized (from 107 to 100), using an enhancement of the method described in Sect. 4.

4 Complexity and Enhancements

The inverse method described in Sect. 3 can be recapitulated as follows:

- find an order \mathcal{O} induced by a specific point \bar{d}_0
- construct the associated set of δ -constraints \mathcal{J}^*
- find an optimal solution (\bar{l}^*, \bar{u}^*) of the associated system \mathcal{J}^* .

We have already mentioned that the last step can be seen as an optimization problem of linear programming, and, as such, is polynomial in N . It is easy to see that the two first steps are also polynomial in N (actually, respectively quadratic and linear). It follows:

Proposition 4.1 *Given a system of time separation of events with N events, with an acyclic timing constraint graph of unique source and j an event of interest, the complexity of finding the domain $\Delta^* = (l^*, u^*)$ and the maximal separation time MAX_j^* by the inverse method is polynomial in N .*

Let us recall that the number N of edges of the timing constraint graph is at most quadratic in the number n of vertices of the original graph (see footnote of Sect. 2).

Let us now focus on a possible enhancement of the method. In order to have a domain Δ^* as large as possible, it is of great interest to be allowed to *remove* inequalities from \mathcal{I}^* , hence from \mathcal{J}^* , as much as possible. Intuitively, in order to guarantee that the representative path π_j^* is still critical, we need to check that the removal of an inequality of \mathcal{I}^* , does not entail any new path to j of separation time greater than $\sum_{\pi_j^*} \bar{\delta}^*$. Formally:

Proposition 4.2 Let \mathcal{O} be a canonical order of events, π_j^* the representative path of j , \mathcal{I}^* and \mathcal{J}^* the associated sets of δ -constraints and $y \leq x$ -constraints respectively. Let I be an inequation of \mathcal{I}^* , and J its counterpart in system \mathcal{J}^* .

Suppose that the system

$$\mathcal{I}^* - \{I\} \cup \{(1-2-3)\} \cup \{t_j > \sum \pi_j^* \bar{\delta}\} \tag{4}$$

has no solution in $\bar{\delta}$. Then, for all solutions $\bar{l}^{**}, \bar{u}^{**}$ of $\mathcal{J}^* - \{J\}$ in \bar{x}, \bar{y} resp., and $\Delta^{**} = (\bar{l}^{**}, \bar{u}^{**})$, we have: $MAX_j^{**} \equiv \max_{\bar{\delta} \in \Delta^{**}} t_j = \sum \pi_j^* \bar{u}^{**}$.

Note that checking the unsatisfiability of system (4) is *exponential* in the number N of events, due to the presence of min and max constraints in (2)-(3) (The general problem is indeed NP-complete; see, e.g. [4]). We loose therefore the polynomial-time complexity result of the basic method. Still, this enhancement turns out to be useful in practice.

Example 4.3 In order to decrease the lower bound of δ_{setup_D} , we have to remove the “restrictive” inequality, say J , of \mathcal{J}_0^* : $y_0^\uparrow + y_1^\uparrow \leq x_{setup_D} + x_3^\downarrow$. This inequality originates itself from inequality, say I , of \mathcal{I}_0^* : $\delta_0^\uparrow + \delta_1^\uparrow \leq \delta_{setup_D} + \delta_3^\downarrow$. Let us consider system $\mathcal{I}_0^* - \{I\} \cup \{(1-2-3)\} \cup \{t_{23} > \sum \pi_{23}^* \bar{\delta} \equiv d_{HI} + d_{LO} + \delta_3^\downarrow + \delta_{15}^\downarrow + \delta_8^\downarrow + \delta_7^\uparrow\}$.

It can be checked that such a system has no solution in $\bar{\delta}$. Hence J can be safely removed from \mathcal{J}_0^* . After instantiation of \bar{x}, \bar{y} with \bar{l}, \bar{u} respectively, for components distinct from $setup_D$ and $setup_W$, $\mathcal{J}_0^* - \{J\}$ then reduces to:

$$\begin{array}{ll} y_{setup_D} \leq 110 & \text{(from: } y_{setup_D} \leq d_{HI} + d_{LO} \text{)} \\ 100 \leq x_{setup_D} & \text{(from: } y_0^\uparrow + y_1^\uparrow + y_5^\downarrow \leq x_3^\downarrow + x_2^\downarrow + x_{setup_D} \text{)} \\ y_{setup_W} \leq 53 & \text{(from: } y_3^\uparrow + y_2^\uparrow + y_{setup_W} \leq d_{LO} + x_{13}^\downarrow \text{)} \\ 46 \leq x_{setup_W} & \text{(from: } d_{LO} \leq x_3^\uparrow + x_2^\uparrow + x_{setup_W} \text{)} \\ y_{setup_D} \leq x_{setup_W} + 64 & \text{(from: } y_{13}^\downarrow + y_{14}^\downarrow + y_{16}^\downarrow + y_{setup_D} \leq x_{setup_W} + x_0^\uparrow \text{)} \end{array}$$

Accordingly, solutions are now:

$$l_{setup_D}^{**} = 100, \quad u_{setup_D}^{**} = 110, \quad l_{setup_W}^{**} = 46, \quad u_{setup_W}^{**} = 53,$$

for which $MAX_{23}^{**} \equiv \max_{\bar{\delta} \in \Delta^{**}} t_{23}$ is still equal to 56 (+ $d_{HI} + d_{LO}$).

Similar results were obtained in [8] by addressing the direct problem, but it required the decomposition of the system into three smaller parts, and heavily relied on heuristics (via the repeated integration of the negation of “suspect” inequalities).

In case the space of solutions S of system (4) (see Prop. 4.2) is *non* empty, one can still remove (the counterpart of) inequality I : This requires the integration of the negation of an inequality delimiting the convex hull of S (cf. Appendix 2).

Implementation. Rather than a direct implementation of the inverse method, we transform the given timing constraint graph into a synchronized product of timed automata [1], then used the facilities of HYTECH [11]. Basically, each node of the timing constraint graph corresponds to a timed automaton and each edge as a synchronized transition. It is then easy to infer the order \mathcal{O} induced by a particular point \bar{d}^0 , then to generate, via parametric reachability analysis, the associated set \mathcal{I}^* of δ -constraints. The test of unsatisfiability for removing inequalities is also done using HYTECH. Apart from the two examples given here, we tried other examples

(SP2 from [8], buffer from [7]). They all took less than 5 minutes (on 1GHz PowerPC G4 with 512 MB of memory).

5 Final Remarks

The main advantage of the inverse method, compared with the direct method, is that the method gives an *exact* maximal separation time, which can be computed in a *polynomial* time (at least without the enhancement of inequality removal). A drawback is that the range of some component delays may have to be tightened with respect to the original interval. On the other hand, many other intervals may be relaxed substantially. Results obtained with the inverse method thus give a useful complementary information.

The inverse method basically relies on the choice of the input point \bar{d}^0 of the component delays. Intuitively, this point corresponds to a point of “good behavior”, and the method infers a (rectangular) domain for points that behave similarly. Such a method is similar in spirit to a common engineering practice, where the parameters of the systems are tuned around a typical point of good behavior. Note also that the “critical path” (the representative path to the output event j) is often known in the area of digital circuits, when the system is designed in a top-down manner by assembling portions of circuit around this crucial path (see, e.g., [10]). Such a critical path induces a *partial* order on events that may be useful to infer the complete order \mathcal{O} (instead of starting from \bar{d}^0).

Finally, let us note that timing constraint graphs correspond to *unconditional* systems like “free-choice” Petri nets [3]. The inverse method can be extended to more general systems (timed Petri nets or timed automata) in a natural manner.

References

- [1] R. Alur and D.L. Dill. A theory of timed automata. *TCS* 126, 1994.
- [2] T. Amon, G. Borriello, T. Hu, and J. Liu. Symbolic timing verification of timing diagrams using Presburger formulas. In *Proc. 34th DAC*, 1997.
- [3] W. Belluomini and C. Myers. Timed state space exploration using posets. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 19(5), 2000.
- [4] T. Burks and K. Sakallah. Min-max linear programming and the timing analysis of digital circuits. In *Proc. of ICCAD*, pages 152–155, 1993.
- [5] S. Chakraborty, D. Dill, and K. Yun. Efficient algorithms for approximate time separation of events. *Sadhana*, 27(2):129–162, 2002.
- [6] S. Chakraborty, D. Dill, K. Yun, and K. Chang. Practical timing analysis of asynchronous circuits using time separation of events. In *Proc. of IEEE Custom Integrated Circuits Conference, 1998*, 1998.
- [7] S. Chakraborty, P. Subrahmanyam, and D. Dill. Approximate time separation of events in practice. In *Proc. of 5th ACM/IEEE Int. Workshop TAU*, 1997.
- [8] R. Chevallier, E. Encrenaz-Tiphène, L. Fribourg, and W. Xu. Verification of the generic architecture of a memory circuit using parametric timed automata. In *Proc. FORMATS’06*, volume 4202 of *LNCS*, pages 113–127. Springer, 2006.
- [9] R. Clariso and J. Cortadella. Verification of timed circuits with symbolic delays. In *Proc. ASP-DAC*, pages 628–633, 2004.

- [10] J. Cortadella, M. Kishinevsky, S. Burns, A. Kondratyev, L. Lavagno, K. Stevens, A. Taubin, and A. Yakovlev. Lazy transition systems and asynchronous circuit synthesis with relative timing assumptions. *IEEE Trans. on CADICS* 21:2, 2002.
- [11] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. A User Guide to HYTECH. In *TACAS'95*, volume 1019 of *LNCIS*, pages 41–71. Springer, 1995.
- [12] K. McMillan and D. Dill. Algorithms for interface timing specification. In *Proc. of the IEEE Int. Conf. on Computer Design (ICCD)*, pages 48–51, 1992.

Appendix 1: System \mathcal{I}_0^* associated to \mathcal{O}_0 for SPSMALL

- (i) $0 \leq d_{HI} + d_{LO} - \delta_{setup_D}$ (t0 \leq t1)
- (ii) $d_{LO} - \delta_{setup_D} \leq 0$ (t1 \leq t3 after elim. of d_{HI})
- (iii) $0 \leq \delta_3^\uparrow$ (t3 \leq t4 after elim. of d_{HI})
- (iv) $\delta_3^\uparrow \leq d_{LO} - \delta_{setup_W}$ (t4 \leq t2 after elim. of d_{HI})
- (v) $d_{LO} - \delta_{setup_W} \leq \delta_3^\uparrow + \delta_2^\uparrow$ (t2 \leq t6 after elim. of d_{HI})
- (vi) $\delta_3^\uparrow + \delta_2^\uparrow \leq d_{LO} - \delta_{setup_W} + \delta_{13}^\downarrow$ (t6 \leq t5 after elim. of d_{HI})
- (vii) $0 \leq \delta_{14}^\downarrow$ (t5 \leq t19 after elim. of $d_{HI} + d_{LO} - \delta_{setup_W}$)
- (viii) $0 \leq \delta_{16}^\downarrow$ (t19 \leq t20 after elim. of $d_{HI} + d_{LO} - \delta_{setup_W} + \delta_{13}^\downarrow + \delta_{14}^\downarrow$)
- (ix) $-\delta_{setup_W} + \delta_{13}^\downarrow + \delta_{14}^\downarrow + \delta_{16}^\downarrow \leq -\delta_{setup_D} + \delta_0^\uparrow$ (t20 \leq t7 after elim. of $d_{HI} + d_{LO}$)
- (x) $-\delta_{setup_D} + \delta_0^\uparrow \leq 0$ (t7 \leq t8 after elim. of $d_{HI} + d_{LO}$)
- (xi) $0 \leq -\delta_{setup_D} + \delta_0^\uparrow + \delta_1^\uparrow$ (t8 \leq t15 after elim. of $d_{HI} + d_{LO}$)
- (xii) $[-\delta_{setup_D} + \delta_0^\uparrow + \delta_1^\uparrow \leq \delta_3^\downarrow]$ (t15 \leq t9 after elim. of $d_{HI} + d_{LO}$)
- (xiii) $0 \leq \delta_{15}^\downarrow$ (t9 \leq t18 after elim. of $d_{HI} + d_{LO} + \delta_3^\downarrow$)
- (xiv) $\delta_3^\downarrow + \delta_{15}^\downarrow \leq -\delta_{setup_D} + \delta_0^\uparrow + \delta_1^\uparrow + \delta_5^\uparrow$ (t18 \leq t16 after elim. of $d_{HI} + d_{LO}$)
- (xv) $-\delta_{setup_D} + \delta_0^\uparrow + \delta_1^\uparrow + \delta_5^\uparrow \leq \delta_3^\downarrow + \delta_2^\downarrow$ (t16 \leq t13 after elim. of $d_{HI} + d_{LO}$)
- (xvi) $\delta_2^\downarrow \leq \delta_{15}^\downarrow + \delta_8^\downarrow$ (t13 \leq t25 after elim. of $d_{HI} + d_{LO} + \delta_3^\downarrow$)
- (xvii) $\delta_3^\downarrow + \delta_{15}^\downarrow + \delta_8^\downarrow \leq d_{HI}$ (t25 \leq t17 after elim. of $d_{HI} + d_{LO}$)
- (xviii) $d_{HI} \leq \delta_3^\downarrow + \delta_{15}^\downarrow + \delta_8^\downarrow + \delta_7^\uparrow$ (t17 \leq t23 after elim. of $d_{HI} + d_{LO}$)
- (xix) $\delta_3^\downarrow + \delta_{15}^\downarrow + \delta_8^\downarrow + \delta_7^\uparrow \leq d_{HI} + d_{LO}$ (t23 \leq t24 after elim. of $d_{HI} + d_{LO}$)

Appendix 2: A system of concurrent processors

In [7], a multiprocessor system is evaluated by solving the direct problem. The timing constraint graph is depicted in Fig.2. The separation time of interest is here t_{19} . The set of inequations $\bar{\delta} \in (\bar{l}, \bar{u})$ is here of the form: $\delta_{0,1} \in (5, 10)$, $\delta_{0,2} \in (5, 10)$, $\delta_{4,5} \in (3, 5)$, $\delta_{7,8} \in (3, 5)$, $\delta_{6,9} \in (6, 12)$, $\delta_{8,10} \in (5, 10)$, $\delta_{8,11} \in (5, 10)$, $\delta_{9,15} \in (6, 12)$, $\delta_{13,14} \in (3, 5)$, $\delta_{15,18} \in (6, 12)$, $\delta_{16,17} \in (3, 5)$, $\delta_{18,19} \in (6, 12)$. As for \bar{d}^0 , take the middle of each interval (i.e.: $\delta_{pq} = \lfloor \frac{u_{pq} - l_{pq}}{2} \rfloor$). This yields \mathcal{O} :

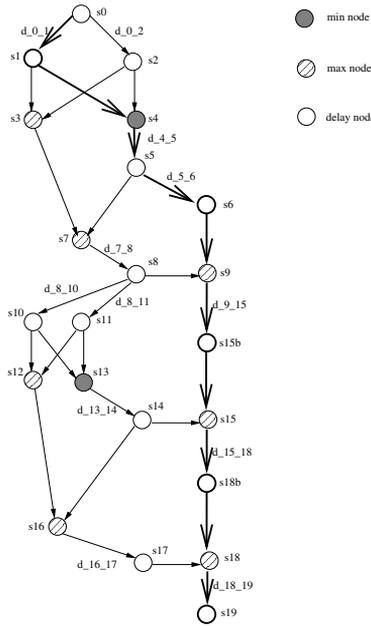


Fig. 2. Timing constraint graph for the system of concurrent processors

$$t_0 \leq t_1 \leq t_2 \leq t_5 \leq t_6 \leq t_8 \leq t_{10} \leq t_{11} \leq t_{14} \leq t_{15} \leq t_{17} \leq t_{18} \leq t_{19}$$

The delay-nodes representing the min/max nodes are given by:

$$3^* = 2, 4^* = 1, 7^* = 5, 9^* = 6, 12^* = 11, 13^* = 10, 15'^* = 15, 16^* = 14, 18'^* = 18.$$

The support of the representative path to 19 is in bold font in Fig. 2. The system \mathcal{I}^* is:

- (i) $\delta_{0,1} \leq \delta_{0,2}$ (t1 ≤ t2)
- (ii) $\delta_{0,2} \leq \delta_{0,1} + \delta_{4,5}$ (t2 ≤ t5)
- (iii) $\delta_{7,8} \leq \delta_{5,6}$ (t8 ≤ t9)
- (iv) $[\delta_{5,6} \leq \delta_{7,8} + \delta_{8,10}]$ (t9 ≤ t10)
- (v) $\delta_{8,10} \leq \delta_{8,11}$ (t10 ≤ t11)
- (vi) $[\delta_{8,11} \leq \delta_{8,10} + \delta_{13,14}]$ (t11 ≤ t14)
- (vii) $\delta_{7,8} + \delta_{8,10} + \delta_{13,14} \leq \delta_{5,6} + \delta_{9,15}$ (t14 ≤ t15)
- (viii) $[\delta_{5,6} + \delta_{9,15} \leq \delta_{7,8} + \delta_{8,10} + \delta_{13,14} + \delta_{16,17}]$ (t15 ≤ t17)
- (ix) $\delta_{7,8} + \delta_{8,10} + \delta_{13,14} + \delta_{16,17} \leq \delta_{5,6} + \delta_{9,15} + \delta_{15,18}$ (t17 ≤ t18)

We can remove the 4th, 6th and 8th inequalities (under brackets), using the enhancement of removal of inequalities (see Sect. 4). Further restrictive inequalities $\delta_{0,1} \leq \delta_{0,2}$ and $\delta_{8,10} \leq \delta_{8,11}$ can be also removed, but this requires the addition of new inequalities (obtained by negating constraints delimiting the hull of solutions of system (4)). This yields the set \mathcal{I}^{**} :

- (i) $\delta_{0,2} \leq \delta_{0,1} + \delta_{4,5}$
- (ii) $\delta_{0,1} \leq \delta_{0,2} + \delta_{4,5}$ (added for removing $\delta_{0,1} \leq \delta_{0,2}$)

$$(iii) \quad \delta_{7,8} \leq \delta_{5,6}$$

$$(iv) \quad \delta_{7,8} + \delta_{8,10} + \delta_{13,14} \leq \delta_{5,6} + \delta_{9,15}$$

$$(v) \quad \delta_{7,8} + \delta_{8,11} + \delta_{13,14} \leq \delta_{5,6} + \delta_{9,15} \quad (\text{added for removing } \delta_{8,10} \leq \delta_{8,11})$$

$$(vi) \quad \delta_{7,8} + \delta_{8,10} + \delta_{13,14} + \delta_{16,17} \leq \delta_{5,6} + \delta_{9,15} + \delta_{15,18}$$

$$(vii) \quad \delta_{7,8} + \delta_{8,11} + \delta_{13,14} + \delta_{16,17} \leq \delta_{5,6} + \delta_{9,15} + \delta_{15,18} \quad (\text{added for removing } \delta_{8,10} \leq \delta_{8,11})$$

Possible solutions of the associated system \mathcal{J}^{**} are:

$$l_{5,6}^{**} = l_{9,15}^{**} = 10, \quad l_{4,5}^{**} = 5, \quad l_{0,1}^{**} = l_{0,2}^{**} = 5, \quad l_{15,18}^{**} = 5, \quad l_{p,q}^{**} = 0,$$

for all other (p, q) .

The lower bounds of $\delta_{5,6}$ and $\delta_{9,15}$ are thus *increased* (from 6 to 10), as well as the lower bound of $\delta_{4,5}$ (from 3 to 5). This is a *restriction* of their original domain. The set $\{\delta_{4,5}, \delta_{5,6}, \delta_{9,15}\}$ is thus a set of “key-parameters” whose tightening allows to *extend* the domain of all the other $\delta_{p,q}$ s (smaller lower bounds). Besides, we have:

$$\begin{aligned} MAX_{19}^{**} &= \max_{\bar{\delta} \in \Delta^{**}} t_{19} = u_{0,1} + u_{4,5} + u_{5,6} + u_{9,15} + u_{15,18} + u_{18,19} \\ &= 10 + 5 + 12 + 12 + 12 + 12 = 63. \end{aligned}$$

This coincides with the upper bound value obtained in [7].