A min-max problem for the computation of the cycle time lower bound in interval-based Time Petri Nets*

Simona Bernardi[1] and Javier Campos[2]

[1] Centro Universitario de la Defensa, Academia General Militar, Zaragoza
[2] Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza
{simonab, jcampos}@unizar.es

Abstract. The time Petri net with firing frequency intervals (TPNF) is a modeling formalism used to specify system behavior under timing and frequency constraints. Efficient techniques exist to evaluate the performance of TPNF models based on the computation of bounds of performance metrics (e.g., transition throughput, place marking). In this paper, we propose a min-max problem to compute the cycle time of a transition under optimistic assumptions. That is, we are interested in computing the lower bound. We will demonstrate that such a problem is related to a maximization linear programming problem (LP-max) previously stated in the literature, to compute the throughput upper bound of the transition. The main advantage of the min-max problem compared to the LP-max is that, besides the optimal value, the optimal solutions provide useful feedback to the analyst on the system behavior (e.g., performance bottlenecks). We have implemented two solution algorithms, using *CPLEX* APIs, to solve the min-max problem, and have compared their performance using a benchmark of TPNF models, several of these being case studies. Finally, we have applied the min-max technique for the vulnerability analysis of a critical infrastructure, i.e., the Saudi Arabian crude-oil distribution network.

1 Introduction

Nowadays, there is increasing demand for large scale and distributed systems that are required to fulfil their mission in a timely manner, despite the presence of both accidental and malicious faults. The assessment of timing and performance requirements is a key issue in the development of such systems and often the analysis techniques used in the validation process are based on modeling.

The Time Petri Net with firing frequency intervals (TPNF) [1] is a modeling formalism that enables the system behavior to be specified under timing constraints, related to the duration of activity, and frequency constraints, related to

alternative behaviors. Such constraints are expressed by firing time/frequency intervals associated to the transitions of the TPNF model.

TPNF can be applied in different application domain contexts [1, 2], e.g., flexible manufacturing systems, embedded real-time domain, and, in general, wherever timing and performance evaluation is a primary concern in the validation process.

The TPNF analysis techniques basically consist in formulating (and solving) proper linear programming problems [1] where the objective function —to be either maximized or minimized— represents a metric in the Petri net context (e.g., a transition throughput, a place marking), and the set of constraints includes marking reachability and routing restrictions, considering the net topology, and enabling operational law constraints.

In this paper, we propose a convex optimization problem for the computation of the lower bound cycle time of a transition t of a TPNF model, that is the time duration between two consecutive firings of t under the best performance case assumption. Such a problem is a min-max problem [3], where the objective function is quadratic and characterized by two kinds of variable vectors: the $\mathbf{y}$s, related to places, that enable the identification of the subset of the TPNF model where the weighted sum of tokens is constant (i.e., place invariant), and the $\mathbf{v}$s, the visit ratios associated to transitions (i.e., relative throughputs). The set of linear constraints can be partitioned into two subsets, each one involving one variable vector, and the optimum value is computed by maximizing with respect to the place variables $\mathbf{y}$s and minimizing with respect to the transition variables $\mathbf{v}$s.

The paper is organized as follows. The rest of this section explains the contribution of the paper and describes related works. In Section 2 the basic notions of TPNF and the associated LP-max for the computation of the throughput upper bound of a transition are recalled. Section 3 introduces the min-max problem by reasoning on the net behavior. The solution algorithms and the analysis of their performance, using the TPNF benchmark, are described in Section 4. Section 5 describes the case study and, finally, Section 6 concludes the paper. In Appendix A, we formally derive the min-max problem from the LP-max.

1.1 Paper contribution

The main contribution of this paper is the statement of a new optimization problem. This is an alternative to the LP-max problem, which was introduced in [1] to compute the throughput upper bound of a transition t of a TPNF model. We also prove formally (Appendix A) that the two problems are related; in particular, the proposed problem can be derived from the LP-max problem and the optimal value of the min-max (i.e., the cycle time lower bound of transition t) is the inverse of the optimal value of the LP-max (i.e., throughput upper bound of t).

The advantage of the proposed min-max problem compared to the LP-max is that, besides the optimal value, the optimal solution vectors $\mathbf{y}^*$ and $\mathbf{v}^*$ provide useful feedback to the analyst on the system behavior. In particular, the $\mathbf{y}^*$s

enable the slowest subnet(s) of the model to be identified, where the cycle time of the transitions in such subnet(s) corresponds to the optimal value. Furthermore, the optimal visit ratios $\mathbf{v}^*$ indicate the frequencies of alternative system behaviors that minimize the cost function, i.e., the cycle time.

It is worth noticing that the min-max problem provides a generalization of a linear programming problem (LPP), previously stated in literature [4], for a particular structural class of Timed Petri Nets, namely freely-related T-semiflows (FRT). Indeed, for FRT Timed Petri Nets, the firing delays and firing frequencies associated to transitions are fixed values (rather than intervals) and there is a single visit ratio vector that can be computed —in polynomial time— by solving a homogenous system of linear equations. Then, when the min-max problem is applied to FRT Timed Petri Nets we obtain the LPP stated in [4].

A second contribution of this work is the comparative performance analysis of two solution algorithms for the min-max problem. The solution algorithms have been implemented using the *CPLEX Callable Library* [5]: one is based on the sub-gradient method [3] and thus provides an approximate solution; the other computes instead the exact solution. It requires, however, the computation of the optimal solution of the related LP-max.

The analysis of the algorithms has been carried out with the support of a benchmark of models that has been built for this purpose. We have gathered 40 TPNF models, several of them being case studies from the literature [1, 6–8, 2, 9–14].

Finally, we show the applicability of the min-max technique in the vulnerability analysis of a critical infrastructure, i.e., the crude-oil distribution network of Saudi Arabia [15], a terrorist target in 2006. In particular, we have studied the effect of different attack plans, where up to three resources of the network are attacked in order to reduce the network throughput to half of the current overall capacity (under normal behavior). The optimal solution $\mathbf{y}^*$ of the min-max problem allowed us to identify the critical parts of the network which should be protected against coordinated attacks. In addition, the optimal solution $\mathbf{v}^*$ provided suggestions for improving the survivability of the system, i.e. on how to distribute the oil-flow between alternative paths in order to reduce the economic loss due to an attack.

1.2   Related work

Time Petri Nets [16] have been extensively used for the validation of timing requirements. Most of the proposed solution techniques, e.g., [17], [18], [19], are enumerative, i.e., based on the construction of the state space of the TPN model. Vicario [17] computes tight bounds on the maximum and minimum execution time of feasible traces. Wang et al. [18] verify timing requirements using on-the-fly techniques to avoid the complete generation of the state space. Xu et al. [19] propose a compositional technique to manage the complexity of the schedulability analysis.

On the one hand, extensions of the TPN formalism have been proposed to support the modeling of activity preemption [20] as well as performance analy-

sis [21]. On the other hand, different classes of Petri Nets have been extended with timing interval specifications to verify end-to-end time constraints, e.g., in context-aware systems [22] and in workflow management systems [23]. However, the solution techniques remain basically enumerative: Lime and Roux [20] define Scheduling Extended Time Petri Nets (SETPN) and provide an approximation method for computing the state space of a SETPN model as a stopwatch automaton. Stochastic preemptive Time Petri Nets [21] are proposed for the performance analysis of real-time systems, based on the generation of a stochastic class graph. Han and Yong [22] introduce interval time Colored Petri Nets, where time stamps are associated to colored tokens and timing intervals are assigned to transition output arcs. Wang and Zeng [23] extend Work-Flow nets by characterizing each place with a minimum and a maximum duration time. Places model workflow activities. The timeline analysis in [22] and [23] relies on the construction of the states reachable from the initial state, via the considered firing sequence.

Linear programming techniques have been applied to the performance analysis of Petri Net models (e.g., [1, 4, 24–26]) mainly to overcome the state explosion problem, which is often suffered by enumerative methods. Beside the bound computation, linear programming techniques and, in general, optimization methods are powerful techniques that enable the identification of those subnets of a Petri Net model which are *critical* with respect to a given system non-functional requirement, such as performance, timing or dependability.

Rodríguez and Julvez [27] propose an iterative algorithm to compute tight performance bounds for stochastic Marked Graphs (a structural PN subclass). The algorithm produces also the performance bottleneck subnet, which is determined through the solution of a linear programming problem associated to the Petri Net model.

In [2], we applied TPN bound techniques for the assessment of timing requirements in software design. Similarly to [27], the solution of the LPP associated to a TPN model enables the location of the bottleneck subnets. The latter correspond to risk causes in the design (i.e., highest demanded hw/sw components, time consuming interaction paths) and their identification provides useful feedback to the software engineer.

In the dependability analysis context, Ramírez et al. [28] consider the diagnosability problem of permanent and operational faults in discrete event systems and use interpreted Petri Nets to model the normal and faulty behaviour of the system. An efficient algorithm is provided to check whether a model is diagnosable and, in such a case, to identify the influence areas of a fault. The latter are subnets defined by P-semiflows and are located by exploiting linear programming techniques.

Ohl [29] defines optimization problems similar to the one proposed in this paper. He considers timed PNs (not intervals, but fixed durations) in the particular case of the FRT subclass. The stated min-max problem is then transformed into a LPP that includes as many constraints as the number of P-semiflows (Y), which grows exponentially as a function of the net size. In this paper, we consider

4

instead interval time PNs for arbitrary net subclasses, and propose a problem with a linear number of variables and constraints on the net size.

Optimization problems for particular net subclasses have also been proposed in the context of max-plus [30] and min-plus algebras [31]. In [30] linear systems are considered that are equivalent to PN marked graphs, and optimization problems are defined to express situations in which some of the system coefficients may vary within certain intervals. In our proposal, we include similar interval variations in the time durations and in the firing frequencies of transitions, but these are not restricted to marked graphs.

The work in [31] states a throughput optimization problem, but minimizing other criteria such as the work in process, in the framework of min-plus algebras for the particular case of marked graphs and weighted T-systems.

Finally, El Amraoui et al. [32] address related problems in a different setting, i.e., the domain of the Hoist Scheduling Problem (HSP). The work considers the cyclic time optimisation of handling devices in electroplating facilities with time-window constraints and uses techniques like branch and bound or genetic algorithms to solve proper mixed linear programming problems.

2   Background

In this section, we introduce the basic definitions of interval-based Time Petri Nets and the associated linear programming problem (LPP) —stated in [1]— for the computation of transition throughput upper bounds.

2.1   Time Petri Net

A Time Petri Net (TPN) [16] is a bipartite directed graph $\mathcal{T} = (P, T, \mathbf{B}, \mathbf{F}, \mathbf{a}, \mathbf{b}, \mathbf{M_0})$, where P and T are the disjoint sets of nodes, namely *places* and *transitions*. The former, signified by circles, are used to model conditions; the latter, graphically depicted by bars, represent events/activities that may occur in the system. Immediate transitions are drawn with thin black bars, while timed ones are drawn with thick white bars (Figure 1).

The directed arcs, shown by arrows, describe which places are pre- or post-condition for which transitions. Weights are associated to arcs and they are defined by the pre- and post-incidence matrices $\mathbf{B}$ and $\mathbf{F}$ ($|T| \times |P|$ sized). The two $|T|$ sized, non negative rational valued vectors, $\mathbf{a}$ and $\mathbf{b}$ specify, respectively, the *static earliest and latest firing times* of each transition (in the Figure, they are shown by an interval $[a_i, b_i]$ near to the transition T_i). Places may contain tokens (drawn as black dots), and a token distribution over the set of places is called marking. $\mathbf{M_0}$ is a $|P|$ sized, natural valued vector that defines the *initial marking* of the TPN.

The TPN dynamic is governed by the transition enabling and firing rules. A transition t is enabled in a marking $\mathbf{M}$ if in each of its pre-condition places there are at least as many tokens as the weights of the corresponding arcs. The earliest and latest firing times $[a, b]$ of t are relative to the instant at which the
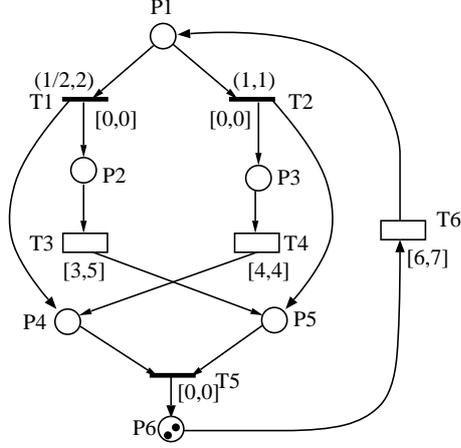
Fig. 1. A simple TPN model

transition was last enabled. Hence, if t has been last enabled at time τ then it may not fire before $\tau + a$ and it must fire before or at $\tau + b$, unless it is disabled previously by the firing of a conflicting transition. The firing itself is immediate and produces a new marking $\mathbf{M'} = \mathbf{M} + \mathbf{F}[t, \cdot] - \mathbf{B}[t, \cdot]$.

2.2 TPN subnets

A TPN subnet $\mathcal{T}'$ of $\mathcal{T}$, defined by $P' \subseteq P$ and $T' \subseteq T$, is characterized by the pre- and post-incidence matrices: $\mathbf{B'} = \mathbf{B}[T', P']$ and $\mathbf{F'} = \mathbf{F}[T', P']$. Its initial marking and the timing specification of its transitions are those of $\mathcal{T}$ restricted to the subset of places P' and transitions T', respectively ($\mathbf{M'_0} = \mathbf{M_0}[P']$, $\mathbf{a'} = \mathbf{a}[T']$ and $\mathbf{b'} = \mathbf{b}[T']$). Subnets defined by a subset of places (transitions), with all their adjacent transitions (places), are called P- (T-) subnets.

2.3 Boundedness and liveness

Boundedness and liveness are among the desirable Petri Net (PN) system properties. A PN system is *bounded* when every place is bounded, e.g., its token content is less than some bound, at every reachable marking. A PN system is *live* when every transition is live, e.g., it can ultimately occur from every reachable marking.

The annullers of the incidence matrix $\mathbf{C^T} = (\mathbf{F} - \mathbf{B})^\mathbf{T}$ and its transpose, $\mathbf{C}$, play an important role in PN theory since they induce invariant relations which are useful for reasoning about PN properties such as boundedness and liveness. In particular, *semiflows* are integer and non negative annullers of $\mathbf{C^T}$ and $\mathbf{C}$: they are called T- and P-semiflows, respectively. A semiflow is called *minimal* when its support $\|\mathbf{x}\|$, i.e., the set of the non-zero components of the annuller $\mathbf{x}$,

is not a proper superset of the support of any other semiflow, and the greatest common divisor (GCD) of its elements is one.

Qualitative analysis of Petri nets is out the scope of this paper, therefore we will just assume in the following that proper functional properties, such as boundedness and liveness, hold and have been tested for the models considered. Notice that other stronger model properties such as home state existence or reversibility are not needed since the goal of this paper is not to compute an average behaviour of the state (marking process) of the model, but just to bound the maximum transition throughput considering all possible model behaviours (possibly, different livelocks) or, equivalently, the lower bound for a transition cycle time (we define cycle time of a transition as the inverse of its throughput).

2.4 Time Petri Net with firing frequency intervals

Throughout the paper, we will consider a subclass of TPN, namely Time Petri Nets with firing frequency intervals (TPNF). A TPNF is a pair $\mathcal{TF} = (\mathcal{T}, \mathcal{R})$, where $\mathcal{T}$ is the underlying TPN model and $R : T \hookrightarrow \mathbb{R}^+ \times \mathbb{R}^+$ is the partial function that assigns an interval of firing frequencies $(\underline{r}^t, \bar{r}^t)$, $\underline{r}^t \leq \bar{r}^t$, to each transition in equal conflict relation.

The equal conflict relation [33]:

$$t_i \; EQ \; t_j \;\; \text{iff} \;\; \mathbf{B^T}[t_i, P] = \mathbf{B^T}[t_j, P] \neq \mathbf{0}.$$

is an equivalence relation that partitions the set of transitions of the net into equivalence classes ECS_j, called *equal conflict sets*. For each equal conflict set $ECS \subseteq T$, the function R satisfies the following constraint: there exists a transition $t_0 \in ECS$, where $R(t_0) = (1, 1)$. Without loss of generality, we assume that the transitions belonging to an equal conflict set are immediate (i.e., $a[t] = b[t] = 0$) [6].

When the ECS of a TPNF is enabled (i.e., all the transitions $t \in ECS$ are enabled in a given marking $\mathbf{M}$), the conflict among the conflicting transitions is resolved in a probabilistic manner by a discrete random variable (d.r.v.) from the family $\mathcal{F} = \{X_{\mathbf{p}}\}_{\mathbf{p} \in \mathcal{R}}$, where each d.r.v. $X_{\mathbf{p}}$ specifies which transition $t_j \in ECS$ will fire once enabled, as follows:

$$Pr\{X_{\mathbf{p}} = t_j \mid ECS \text{ enabled}\} = \begin{cases} p_0 > 0 & \text{if } t_j = t_0, \\ p_j & \text{otherwise.} \end{cases}$$

Since $X_{\mathbf{p}}$ is a d.r.v. then $\sum_{j=0}^{n} p_j = 1$. Moreover, the following relations hold between the firing probability and the firing frequency intervals of such transitions:

$$\mathcal{R} = \{\mathbf{p} = (p_0, \dots, p_n) \in [0, 1]^n \subseteq \mathbb{R}^n : \quad \underline{r}^j \leq \frac{p_j}{p_0} \leq \bar{r}^j, t_j \in ECS\}.$$

Observe that the selection of the d.r.v. from the family $\mathcal{F}$ occurs when the ECS becomes enabled and is not deterministic.

In Figure 1, there is one equal conflict between transitions T_1 and T_2; according to the firing frequency specification (i.e., the intervals in round-brackets, near to the transitions), the firing ratio of T_1, with respect to T_2, ranges between half the firing ratio of T_2 and double the latter.

TPNF can be used to model dynamic systems. Efficient techniques exist to evaluate the performance of these modeled systems, based on the computation of performance metric bounds. In [1], a linear programming problem (LPP) was proposed to compute the throughput upper bound of a transition t. The following proposition recalls the result:

Proposition 1 *Let $\mathcal{TF} = (\mathcal{T}, \mathcal{R})$ be a live and bounded TPNF system. A throughput upper bound $x[t_1]$ of a transition $t_1 \in T$, can be computed by solving the LPP:*

$$
\begin{aligned}
P_0 \;=\; & \text{maximize } x[t_1] && (1)\\
& s.t. \; \mathbf{M} = \mathbf{M_0} + \mathbf{C^T}\sigma \\
& \quad \sum_{t \in {}^\bullet p} x[t]F[t,p] = \sum_{t \in p^\bullet} x[t]B[t,p], \forall p \in P \\
& \quad M[p] \geq x[t]a[t]B[t,p], \forall t \in T \text{ and } \forall p \in {}^\bullet t \\
& \quad \underline{r}^j x[t_k] \leq \overline{r}^k x[t_j], \; \underline{r}^k x[t_j] \leq \overline{r}^j x[t_k], \forall t_j, t_k \in ECS \\
& \quad \mathbf{x}, \sigma \geq \mathbf{0}_T, \; \mathbf{M} \geq \mathbf{0}_P
\end{aligned}
$$

where $\mathbf{0}_T$ and $\mathbf{0}_P$ are, respectively, the T-column and P-column null vectors.

The first set of constraints is derived from the net structure and the initial marking; these represent the marking reachability equations. The second set of constraints represents the token flow equations that hold for each bounded place. The third set of constraints is derived by applying the utilization law for queueing network systems [34] to each transition of the TPNF, considering their earliest firing times. The routing inequalities are determined by the firing frequency intervals assigned to the immediate transitions belonging to the same ECS. Finally, all the problem variables (i.e., transition throughput $\mathbf{x}$ and firing count σ vectors, place marking vector $\mathbf{M}$) are non negative.

By applying the LPP to compute, for example, the throughput upper bound of the transition T_6, in Figure 1, we get $\frac{7}{27} \approx 0.26$. Observe that the aforementioned LPP, though useful for calculating the transition throughput upper bound, it does not enable the slowest subnet to be identified (i.e., performance bottleneck).

3 The min-max problem

In this section, we define a problem for computing the cycle time lower bound of a transition $t_1 \in T$ of a live and bounded TPNF model. The transition cycle time is defined as the time duration between two consecutive firings of t_1 or, equivalently,

the inverse of its throughput. First, we introduce the problem variables that have a precise interpretation at the Petri net level, provided that they satisfy a set of constraints. Subsequently, we state the min-max problem assuming that the timed transitions are persistent (i.e., once enabled they eventually fire). The min-max problem can be derived from the LP-max problem P_0 (1), stated in Section 2. The formal proof is given in Appendix A. Herein, we provide instead the rationale behind the objective function and its optimization, which is based on an implicit decomposition of the net into the P-subnets generated by P-semiflows. In particular, we follow a reasoning similar to that used in [4], where a LPP was proposed to compute the cycle time lower bound of transitions in a *Freely Related T-semiflow* Timed Petri Net. An exemplification of the min-max problem is also given by using the running example of Figure 1. Finally, we give a proof sketch of the equivalence between the LP-max P_0 (1) and the min-max problem.

3.1 The problem variables and constraints

We consider two types of non negative variables $y[i](i = 1, \ldots, P)$ and $v[j](j = 1, \ldots, T)$. The former are associated to places and they enable us to identify the P-subnets of the TPNF model where the weighted sum of tokens is constant in every marking $\mathbf{M}$ reachable from the initial marking $\mathbf{M_0}$. Such variables are solutions to the linear system of equation $\mathbf{Cy} = \mathbf{0}_T$; indeed, from the reachability constraints of the LPP P_0 (1), we get $\mathbf{y^T M} = \mathbf{y^T M_0}$. Figure 2 shows two P-subnets of the net in Figure 1, defined by the sets $||\mathbf{y_1}||$ and $||\mathbf{y_2}||$, respectively, where $\mathbf{y_1} = [1, 1, 0, 0, 1, 1]$ and $\mathbf{y_2} = [1, 0, 1, 1, 0, 1]$ are the minimal P-semiflows of the net.
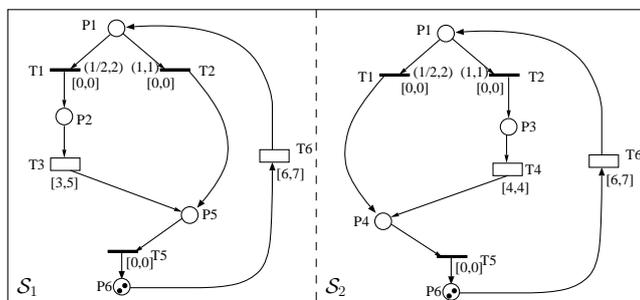

Fig. 2. Net decomposition into P-subnets.

The second type of variables corresponds to the transition visit ratios, that is the transition throughputs relative to the reference transition t_1:

$$v[t_j] = \frac{x[t_j]}{x[t_1]} = \Gamma[t_1]x[t_j] \tag{2}$$

9

where $\Gamma[t_1] = \frac{1}{x[t_1]}$ is the cycle time of t_1. Obviously, the visit ratio of the reference transition is one (i.e., $v[t_1] = 1$). Like the throughput variables, the visit ratios satisfy the token flow constraints and the routing constraints defined in LPP P_0 (1). We rewrite such constraints in matrix form. Thus, the visit ratios are solutions to both the linear system of equations $\mathbf{C^T v} = \mathbf{0}_P$ (token flow) and the linear system of inequalities $\mathbf{R v} \leq \mathbf{0}_K$ (routing). In particular, $\mathbf{R}$ is the routing matrix, $K \times |T|$ sized positive real valued —where $K = \sum_{k=1}^{n} 2(|ECS_k| - 1)$ and n is the number of equal conflict sets (see definition in Appendix A).

The visit ratio vectors of the net in Figure 1 have the general form $\mathbf{v} = [v_1, v_2, v_1, v_2, v_1 + v_2, v_1 + v_2]$ where, $v_1, v_2 \geq 0$ (i.e., they are non negative right-annullers of the incidence matrix). Moreover, the visit ratios of the conflicting transitions T_1, T_2 satisfy the routing constraints: $\frac{1}{2} v_2 \leq v_1 \leq 2v_2$. If, for example, we are interested in computing the cycle time of transition T_6, then its visit ratio should be set to one (which implies: $v_1 + v_2 = 1$).

3.2 Problem statement

The following proposition defines the min-max problem in order to compute the cycle time lower bound of a transition t_1. The problem can be derived from the LPP-max P_0, defined in Section 2, as formally proved in Appendix A.

Proposition 2 *Let $\mathcal{TF} = (\mathcal{T}, \mathcal{R})$ be a live and bounded TPNF system where all the timed transitions are persistent (that is, once enabled they eventually fire). A cycle time lower bound of a transition t_1 (i.e., the inverse of its throughput upper bound) can be computed by solving the min-max problem:*

$$P_1 = \min_{\mathbf{v} \in D_v} \ \max_{\mathbf{y} \in D_y} \ \mathbf{y^T}(\mathbf{B^T} \odot \mathbf{a}) \ \mathbf{v} \tag{3}$$

$$s.t. \ D_y : \left\{ \mathbf{Cy} = \mathbf{0}_T, \mathbf{M_0^T y} = 1, \mathbf{y} \geq \mathbf{0}_P \right\}$$

$$D_v : \left\{ \mathbf{Rv} \leq \mathbf{0}_K, \mathbf{C^T v} = \mathbf{0}_P, v[t_1] = 1, \mathbf{v} \geq \mathbf{0}_T \right\}$$

Rationale behind the objective function The problem is quadratic. In particular, the objective function $F(\mathbf{y}, \mathbf{v}) = \mathbf{y^T}(\mathbf{B^T} \odot \mathbf{a})\mathbf{v}$ is a function of the two types of variable vectors previously introduced. The latter must satisfy two disjonted sets of constraints: D_y and D_v. Set D_y identifies the P-semiflows $\mathbf{y}$ of the net, subject to the normalization constraint $\mathbf{M_0^T y} = 1$, and D_v identifies the possible visit ratios $\mathbf{v}$ of the net, normalized to the visit ratio of transition t_1. Given a P-semiflow $\mathbf{y} \in D_y$, the objective function defines a cycle time lower bound of the transitions of the P-subnet generated by $\mathbf{y}$. Such quantity is the weighted sum of the earliest firing time delays (vector $\mathbf{a}$) associated to the timed transitions of the P-subnet, where the weights are the corresponding visit ratios, divided by the total number of tokens circulating in the subnet (which is a constant).

In the running example (Figure 1), the net is characterized by two P-semiflows that satisfy the normalization constraint $\mathbf{M_0^T y} = 1$: $\mathbf{y_1} = [\frac{1}{2}, \frac{1}{2}, 0, 0, \frac{1}{2}, \frac{1}{2}]$ and

$\mathbf{y_2} = [\frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}]$. The subnets $\mathcal{S}_1$ and $\mathcal{S}_2$ generated by the two P-semiflows are shown in Figure 2. A lower bound cycle time of the transitions belonging to the subnet $\mathcal{S}_1$ is equal to $F(\mathbf{y_1}, \mathbf{v}) = (3v_3 + 6v_6)/2$, where v_3, v_6 are the visit ratio variables associated to the timed transitions T_3 and T_6, with earliest firing time delays equal to 3 and 6, respectively. Similarly, a lower bound cycle time of the transitions belonging to the subnet $\mathcal{S}_2$ is equal to $F(\mathbf{y_2}, \mathbf{v}) = (4v_4 + 6v_6)/2$. Observe that the total number of tokens circulating in both the subnets is equal to 2.

Rationale behind the function optimization Let us start by focusing on the net behavior at the finer-grained level, that is we consider the subnets of the TPNF model defined by a single place together with its ouput transitions.

We apply Little's law [34] to obtain inequalities that include the cycle time of a transition t_1. According to Little's formula, each place $p \in P$ satisfies the equation:

$$M[p] = (\mathbf{B^T}[T, p] \cdot \mathbf{x})\, r[p], \quad \forall p \in P$$

where $r[p]$ is the average time spent by the tokens in p, $M[p]$ is the mean number of tokens in p, and $\mathbf{B^T}[T, p] \cdot \mathbf{x}$ is the output rate of the tokens from place p. Considering the relationship between the transition throughputs and visit ratios (i.e., $\mathbf{v} = \Gamma[t_1]\mathbf{x}$, Eq. 2), we can rewrite the above equation in terms of the cycle time of the reference transition t_1:

$$\Gamma[t_1]M[p] = (\mathbf{B^T}[T, p] \cdot \mathbf{v})\, r[p], \quad \forall p \in P.$$

Due to the persistency assumption, place p can have at most one timed transition t as an output transition, hence the residence time of tokens in p is greater than (or equal to) the earliest firing time delay $a[t]$. Let us consider the simple net in Figure 3, where the transition t is an output transition for both places p_1 and p_2, and it is not enabled in the given marking. The token residence time in place p_1 (p_2) takes into account the time elapsed from the token entrance in the place until the transition enabling, as well as the time delay from the instant of enabling until the firing, which is greater or equal to the earliest firing time delay a.
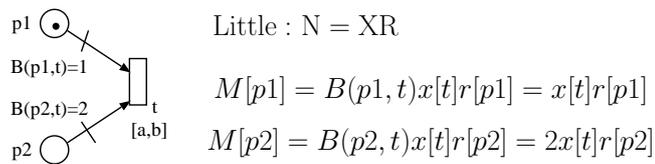

Fig. 3. Application of Little's formula to places in a TPNF model.

Then, the following inequality holds (also in the case of several immediate transitions —i.e., $a[t] = 0$— sharing the input place p), for every place $p \in P$

and every transition $t \in T : B(p,t) > 0$:

$$\Gamma[t_1]M[p] = (\mathbf{B^T}[T,p] \cdot \mathbf{v}) \ r[p] \geq (\mathbf{B^T}[T,p] \cdot \mathbf{v}) \ a[t]. \qquad (4)$$

The inequalities (4), written for each place p of the net, relate the cycle time of transition t_1 to unknown quantities, that is the transition visit ratios (vector $\mathbf{v}$) and the mean number of tokens in p ($M[p]$) which are also variables. A possible approach to make such inequalities explicit with respect to the interested metric (i.e., $\Gamma[t_1]$) is to write a linear combination of the inequalities such that the $M[p]$ variables can be removed. We consider then the P-subnets defined by the set of places where the weighted sum of tokens is constant (and greater than zero) in every marking $\mathbf{M}$ reachable from the initial marking $\mathbf{M_0}$. Such subnets are identified by non negative variables $\mathbf{y}$ that, being the annullers of the matrix $\mathbf{C}$, satisfy the equation $\mathbf{y^T M} = \mathbf{y^T M_0}$.

For each P-subnet $\mathcal{T}'_{\|\mathbf{y}\|}$, we can write a linear combination of the inequalities (4) by multiplying each one with the corresponding p-entry of $\mathbf{y}$:

$$\Gamma[t_1] \geq \frac{\mathbf{y^T}(\mathbf{B^T} \odot \mathbf{a}) \ \mathbf{v}}{\mathbf{y^T M_0}}, \quad \forall \ \mathcal{T}'_{\|\mathbf{y}\|} \qquad (5)$$

where $\mathbf{B^T} \odot \mathbf{a}$ is the $P \times T$ matrix obtained by multiplying the t-column of $\mathbf{B^T}$ by the scalar $a[t]$. A lower bound for the cycle time of t_1 is obtained by taking the maximum over the P-subnets:

$$\Gamma[t_1] \geq \max_{\mathcal{T}'_{\|\mathbf{y}\|}} \ \frac{\mathbf{y^T}(\mathbf{B^T} \odot \mathbf{a}) \ \mathbf{v}}{\mathbf{y^T M_0}}. \qquad (6)$$

Since the TPNF is live, the weighted sum of tokens in the initial marking is greater than zero ($\mathbf{y^T M_0} = q > 0$). Considering the change of variables $\bar{\mathbf{y}}^{\mathbf{T}} = \mathbf{y^T}/q$, the lower bound can be rewritten as:

$$\Gamma[t_1] \geq \overbrace{\max_{\bar{\mathbf{y}} \in D_y} \ \underbrace{\bar{\mathbf{y}}^{\mathbf{T}}(\mathbf{B^T} \odot \mathbf{a}) \ \mathbf{v}}_{F(\bar{\mathbf{y}},\mathbf{v})}}^{\Gamma[t_1](\mathbf{v})} \qquad (7)$$

where $D_y = \left\{ \mathbf{Cy} = \mathbf{0}_T, \mathbf{y^T M_0} = 1, \mathbf{y} \geq \mathbf{0}_P \right\}$.

Observe that, if there exists a single visit ratio vector $\mathbf{v}$, then the inequality (7) boils down to the LPP stated in [4] for *Freely Related T-semiflow* Timed Petri Nets, where the earliest firing time delays (vector $\mathbf{a}$) are the mean service times of the timed transitions.

However, when TPNF are considered, the right member of the inequality (7) is a function of the visit ratio vectors, i.e., $\Gamma[t_1](\mathbf{v})$. On the other hand, each function $F(\bar{\mathbf{y}}, \mathbf{v})$, where $\bar{\mathbf{y}} \in D_y$ is a P-semiflow, provides a lower bound for the cycle time of the transitions in the P-subnet $\mathcal{T}'_{\|\mathbf{y}\|}$. Then, the best lower bound for the cycle time of the transition t_1 is given by the visit ratios that minimize the cycle time of the transitions in the slowest P-subnets.

12

Graphical representation of the min-max problem The min-max problem associated to the running example can be solved graphically. Indeed, the TPNF can be decomposed into the two subnets in Figure 2, generated by the P-semiflows $\mathbf{y_1} = [\frac{1}{2}, \frac{1}{2}, 0, 0, \frac{1}{2}, \frac{1}{2}]$ and $\mathbf{y_2} = [\frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}]$, respectively. The cycle time lower bound of the transitions of the P-subnets can be rewritten as functions of the visit ratio of transition T_2:

$$F(\mathbf{y_1}, v_2) = \frac{3}{2} - \frac{3}{2}v_2 + 3 \qquad \text{and} \qquad F(\mathbf{y_2}, v_2) = 2v_2 + 3,$$

where $\frac{1}{3} \leq v_2 \leq \frac{2}{3}$. The two curves are plotted in Figure 4 versus the visit
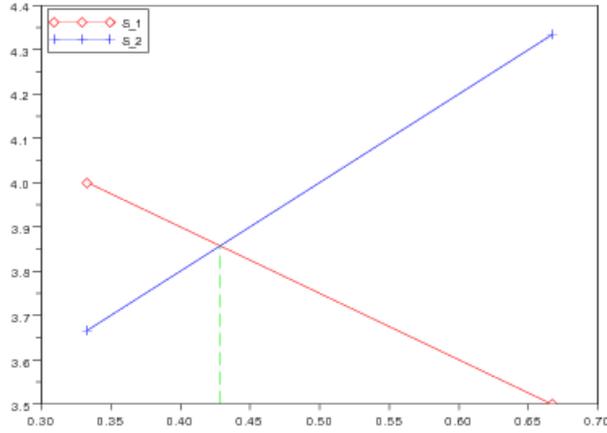

Fig. 4. Graphical resolution of the min-max problem for the running example.

ratio of transition T_2: the slowest subnet is $\mathcal{S}_1$ (red line), when $v_2 \in [\frac{1}{3}, \frac{3}{7})$; when instead $v_2 \in (\frac{3}{7}, \frac{2}{3}]$, the slowest subnet is $\mathcal{S}_2$ (blue line). The minimum of the function $\Gamma[T_6](v_2) = max\{F(\mathbf{y_1}, v_2), F(\mathbf{y_2}, v_2)\}$ is reached at $v_2 = \frac{3}{7}$ and is equal to $\frac{27}{7}$. Then, the lower bound cycle time of T_6 is equal to $\frac{27}{7}$ and the optimal visit ratio vector is $v^* = [\frac{4}{7}, \frac{3}{7}, \frac{4}{7}, \frac{3}{7}, 1, 1]$. Observe that the point $(\frac{3}{7}, \frac{27}{7})$ is the intersection of the two curves; this means that the transitions of both the P-subnets $\mathcal{S}_1$ and $\mathcal{S}_2$ are characterized by the same lower bound cycle time. Indeed, the min-max problem for the running example has two optimal solutions $y_1^* = [\frac{1}{2}, \frac{1}{2}, 0, 0, \frac{1}{2}, \frac{1}{2}]$ and $y_2^* = [\frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}]$, whose supports $||y_1^*||$ and $||y_2^*||$ define the two P-subnets $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively.

3.3 Equivalence with the LP-max problem

The min-max problem P_1 (3) can be derived from the LPP P_0 (1) stated in Prop. 1, Section 2. Moreover, an optimal solution of the min-max is equal to the inverse of the optimal solution of the initial LPP. In Appendix A, we formally prove the equivalence of the two problems. Here we provide a the general idea of the proof.

Proposition 3 *Let $\mathcal{TF} = (\mathcal{T}, \mathcal{R})$ be a live, bounded and persistent TPNF system. The following relation holds between the optimal values of the min-max problem P_1 (3) and of the LPP P_0 (1):*

$$P_1 = \frac{1}{P_0}.$$

Proof (Proof sketch). Firstly, by considering the inverse relationship between the transition throughput and the transition cycle time, the initial LPP P_0 is transformed into an equivalent LPP $Q = \frac{1}{D(P_0)}$, where $D(P_0)$ is the dual of P_0. Secondly, a Lagrangian relaxation is applied to the LPP Q. Thirdly, by reasoning on the meaning of the problem variables, the domain of the Lagrangian multipliers is restricted, then leading to the min-max problem P_1 which provides an upper-bound of the minimum transition cycle time (i.e., $P_1 \geq Q$). Afterwards, we apply the same reasoning to the dual of Q, i.e., $D(Q)$, and we obtain a max-min problem P_2 (where $D(Q) \geq P_2$). Finally, we prove that P_2 and P_1 are equivalent, that is an optimal solution of P_2 is an optimal solution of P_1 (and viceversa).

4 Performance analysis of the solution techniques

4.1 Solution algorithms

We have used the *CPLEX Callable Library* [5] to solve the problems P_0 and P_1. The implementation of the former is straightforward since it is an LP problem. On the other hand, the min-max problem is actually a convex optimization problem. We have implemented two algorithms that solve it.

Algorithm 1 is based on the sub-gradient method [3] and thus provides an approximate solution. Firstly, an initial visit ratio vector $\mathbf{v_0}$ is computed (line 1). After initialization of the variables (line 2), the loop (lines 3-17) is executed until either the maximum number of iterations is reached or the optimal solution is found. At each iteration k, the Lagrangian multiplier vector $\mathbf{v_k}$ is fixed and the sub-problem P_{max} is solved (line 4). The optimal solution $\mathbf{y_k}$ of P_{max} is used to compute the sub-gradient in $\mathbf{v_k}$ (line 5). If the sub-gradient is not the null vector, we calculate the step δ (line 9), where α is an input parameter, LB is a lower bound for the problem P_1[3] and the denominator is the square of the length of the sub-gradient. If δ is not negligible (line 10), then we set a new

---

[3] We have set $\alpha = 1$ and $LB = 1/P_0$ in the experiments.

Algorithm 1 MinMax (sub-gradient method)

Require: $\mathbf{B^T}, \mathbf{C^T}, \mathbf{a}, \mathbf{M_0}, \mathbf{R}, t_1$ { *reference transition* } , max_it, α, LB;
Ensure: $P_1(\tilde{\mathbf{y}}, \tilde{\mathbf{v}})$ { *approximate* }
1: Find $\mathbf{v_0} \in D_v$
2: k = 0; found = **false**;
3: **while** $k < max\_it$ **and not** found **do**
4:      Compute $P_{max}(\mathbf{y_k}) = max_{\mathbf{y} \in D_y} \ \mathbf{y^T}(\mathbf{B^T} \odot \mathbf{a}) \ \mathbf{v_k}$
5:      Compute $\mathbf{s_k} = \mathbf{y_k^T} \ (\mathbf{B^T} \odot \mathbf{a})$
6:      **if** $\mathbf{s_k} = \mathbf{0}$ **then**
7:        found = **true**
8:      **else**
9:        Compute $\delta = \alpha \frac{P_{max}(\mathbf{y_k}) - LB}{||\mathbf{s_k}||^2}$
10:        **if** $\delta \ll 0$ **then**
11:          found = **true**
12:        **else**
13:          $\mathbf{v_{k+1}} = \mathbf{v_k} - \delta \ \mathbf{s_k}$ { *Set any negative component* $\mathbf{v_{k+1}}[i]$ *to zero* }
14:        **end if**
15:      **end if**
16:      k = k + 1
17: **end while**
18: **if** $\mathbf{v_k} \neq \mathbf{v_0}$ **then**
19:      Compute projection $\tilde{\mathbf{v}} = \Pi_{D_v}(\mathbf{v_k})$
20:      Compute $P_{max}(\tilde{\mathbf{y}}) = max_{\mathbf{y} \in D_y} \ \mathbf{y^T}(\mathbf{B^T} \odot \mathbf{a}) \ \tilde{\mathbf{v}}$
21: **end if**

Lagrangian multiplier by moving a step in the opposite direction of the sub-gradient (line 13). We are looking for a multiplier that minimizes the objective function from the current direction $\mathbf{v_k}$. Observe that, the multipliers (line 13) are not necessarily visit ratios. Having exited from the loop, the multiplier vector $\mathbf{v_k}(\neq \mathbf{v_0})$ is projected onto the set of feasible solutions (line 19) and a post-evaluation is carried out by solving P_{max} with the projected vector (line 20).

Algorithm 2 provides an exact solution. However, it requires the computation of the optimal solution of the LP-max problem P_0 (line 1). Indeed, as the relation between the visit ratio and the throughput vectors is known (line 2), it is possible to obtain the optimal visit ratios from the solution of P_0 and then solve P_{max} to obtain the optimal values $\mathbf{y}^*$ (line 3).

Algorithm 2 MinMax (direct method)

Require: $\mathbf{B^T}, \mathbf{C^T}, \mathbf{a}, \mathbf{M_0}, \mathbf{R}, t_1$ { *reference transition* };
Ensure: $P_1(\mathbf{y}^*, \mathbf{v}^*)$
1: Compute LP-max $P_0(\mathbf{x}^*, \sigma^*)$
2: $\mathbf{v}^* = \mathbf{x}^*/x_1^*$
3: Compute $P_{max}(\mathbf{y}^*) = max_{\mathbf{y} \in D_y} \ \mathbf{y^T}\mathbf{B^T} \odot \mathbf{a} \ \mathbf{v}^*$

4.2   TPNF benchmark

We have collected 40 TPNF models (Table 1) that belong to different structural PN classes defined in the literature (i.e., monoT-semiflow, free-related T-semiflow, deterministic systems of sequential processes and free-choice) [4].

Table 1. The TPNF benchmark

Net_id	Modeled system	Metric	Size (P,T)
1	flexible manufacturing system [1]	production time	(37,34)
2	communication protocol [6]	turnaround time	(12,11)
3	job shop [7]	cycle time	(22,13)
4	alternating bit protocol [7]	cycle time	(16,12)
5	flexible manufacturing system [8]	production time	(25,22)
6	software execution multi-scenario	response time	(167,109)
7	computer assisted braking system [2]	response time	(72,53)
8	software execution scenario [2]	response time	(36,26)
9	example [7]	cycle time	(6,6)
10	example	cycle time	(9,9)
11	e-health system failure scenario [9]	response time	(89,88)
12	example [7]	cycle time	(11,10)
13	data-flow graph [7]	execution time	(26,28)
14	example	cycle time	(13,12)
15	example	cycle time	(36,29)
16	example	cycle time	(53,33)
17	example	cycle time	(28,28)
18	example	cycle time	(39,29)
19	example	cycle time	(16,15)
20	example	cycle time	(21,14)
21	Ada tasking system [7]	execution time	(13,10)
22	producer-consumer system [7]	response time	(11,8)
23	example	cycle time	(16,13)
24	example	cycle time	(13,12)
25	example [7]	cycle time	(14,14)
26	sw retrieval system (e-commerce) [10]	response time	(48,40)
27	example	cycle time	(12,9)
28	example	cycle time	(30,30)
29	example	cycle time	(13,13)
30	example	cycle time	(10,10)
31	example	cycle time	(14,16)
32	assembly line (push strategy) [11]	production time	(37,20)
33	assembly line (on-demand strategy) [11]	production time	(35,19)
34	assembly line (Kanban strategy) [11]	production time	(31,20)
35	flexible manufacturing system [11]	production time (prod. B)	(86,74)
36	oil pipeline network (Sect. 5)	distribution time	(140,110)
37	oil pipeline network under attack (Sect. 5)	distribution time	(150,117)
38	Universal Control Hub architecture [12]	response time	(21,16)
39	web-service application [13]	response time	(82, 63)
40	mobile agent application [14]	response time	(63, 55)

There are several models of case studies, including that illustrated in Section 5. The others are examples that do not represent particular systems. The metric of interest is basically a cycle time, which for the TPNF modelling case studies has been interpreted in the modelled system context. The last column of Table 1 shows the size of the TPNF models in terms of number of places and transitions, respectively.

4.3 Performance results

The LP-max and min-max solvers have been run on an Intel Core Duo laptop with 2,4GHz CPU. The TPNF benchmark has been used to validate the solvers as well as to evaluate their performance. The execution time has been considered as a performance metric. Table 2 shows the results of the analysis. Columns 2-4 are related to the solver of the LP-max problem P_0; they indicate, respectively, the size of the reduced LPP[4] in terms of number of rows (R), columns (C) and non zero entries in the LPP matrices, the optimal value (i.e., transition throughput) and the solver execution time (ET). Columns 5-7 show the data of

---

[4] Indeed, the CPLEX pre-solver is called on to reduce the size of the problem before executing the simplex algorithm.

Algorithm 1, that is the (approximated) optimal value of the min-max problem P_1, the solver execution time and the number of iterations (k) required to obtain the optimal value. Finally, the last two columns shows the results of the Algorithm 2, i.e., the optimal value of P_1 and the solver execution time.

Table 2. Performance results on the TPNF benchmark.

Net_id	LPP solver			Algorithm 1			Algorithm 2	
	LPP size (R, C, NZ)	Value (P_0)	ET (ms.)	Value (P_1)	ET (ms.)	k	Value (P_1)	ET (ms.)
1	(13, 12, 38)	0.277741	5.826	3.600480	4.567	1	3.600480	6.667
2	(5, 3, 10)	0.173010	5.185	5.780000	3.535	1	5.780000	7.297
3	(18,10, 54)	0.111111	4.646	9.000000	5.296	1	9.000000	10.046
4	(9, 6, 27)	3.857839	4.246	0.259212	3.593	1	0.259212	7.095
5	(17, 15, 45)	0.500000	4.736	2.000000	4.460	1	2.000000	6.969
6	(98, 58, 333)	0.028196	11.341	35.841376	30.664	2	35.466631	15.004
7	(38, 23, 111)	0.241935	8.754	4.133333	6.379	1	4.133333	11.886
8	(22, 14, 63)	0.049505	8.319	20.200000	8.031	1	20.200000	8.503
9	(3, 2, 6)	0.259259	4.029	3.989011	7.469	2	3.857143	7.736
10	(8, 6, 25)	0.196078	4.570	5.100000	6.095	1	5.100000	7.589
11	(67, 57, 208)	0.000257	8.433	3891.567783	18.672	2	3890.956548	11.477
12	(8, 6, 20)	1.125000	4.781	0.888889	5.924	1	0.888889	6.590
13	(18, 15, 61)	0.086022	5.532	13.734375	13.434	3	11.625000	7.347
14	(11, 7, 32)	1.166667	5.085	0.964286	10.058	2	0.857143	6.885
15	(20, 13, 59)	0.109375	6.526	9.142857	6.662	1	9.142857	7.788
16	(35, 18, 100)	0.083333	5.650	12.000000	4.833	1	12.000000	7.883
17	(21, 16, 83)	0.618558	6.362	1.616664	4.486	1	1.616664	7.952
18	(28, 19, 90)	0.211641	5.546	4.799884	11.468	2	4.724985	7.545
19	(10, 7, 34)	0.250000	4.811	4.000000	4.370	1	4.000000	6.715
20	(17,10, 54)	0.666667	5.533	1.710000	10.549	2	1.500000	6.903
21	(5, 4, 15)	0.250000	5.324	4.000000	3.352	1	4.000000	5.863
22	(4, 4, 12)	0.142857	4.132	7.000000	3.535	1	7.000000	6.392
23	(8,7, 25)	0.333333	2.369	3.000000	3.576	1	3.000000	6.363
24	(7, 4, 14)	0.333333	4.791	3.570000	8.487	2	3.000000	5.751
25	(14, 10, 36)	0.428571	4.635	2.589160	9.163	2	2.333333	6.176
26	(36, 28, 132)	0.004470	6.208	224.811869	13.036	2	223.727483	8.913
27	(5, 5, 15)	0.090909	4.294	11.000000	3.890	1	11.000000	5.690
28	(24, 17, 83)	0.128713	5.113	7.769341	6.141	1	7.769341	7.728
29	(10, 8, 35)	0.230769	4.572	4.592469	9.188	2	4.333333	6.199
30	(9, 5, 26)	0.250000	4.550	4.000000	3.789	1	4.000000	6.933
31	(21, 16, 73)	0.142857	4.743	7.000000	4.499	1	7.000000	5.804
32	(23, 12, 64)	0.222222	4.986	4.500000	3.859	1	4.500000	6.757
33	(23, 12, 64)	0.222222	4.712	4.500000	4.200	1	4.500000	6.670
34	(14, 11, 38)	0.285714	4.520	3.500000	3.727	1	3.500000	6.198
35	(58, 45, 221)	0.266667	7.320	3.750000	7.396	1	3.750000	10.657
36	(67, 63, 206)	1.098901	9.426	1.030714	22.637	2	0.910000	11.742
37	(76, 66, 228)	0.523560	9.971	2.039953	22.876	2	1.910000	13.956
38	(12, 9, 29)	0.019608	5.525	51.000000	4.740	1	51.000000	8.218
39	(49, 33, 131)	0.467513	6.663	2.138980	5.615	1	2.138980	9.422
40	(36, 28, 108)	0.014200	6.162	83.065165	13.034	2	70.421333	9.018

From the results of Algorithm 1, observe that for most of the TPNFs the algorithm requires just one iteration and, in this case, the exact optimal value and solution are calculated. On the other hand, when more iterations are needed (denoted in the Table by grey rows) the algorithm simply provides an approximate value (and solution).

Concerning the performance of the solvers, the results show that they are all fast (a few milliseconds are needed to execute them), including for TPNFs with large sizes in terms of places and transitions (e.g., the net_id 6). The ETs of the LPP solver and Algorithm 1, for a given TPNF, are similar when just one iteration is executed by the latter. As expected, the ET of Algorithm 1 becomes greater in the case of more than one iteration. Algorithm 2 performs less well than the LPP solver and Algorithm 1 (in the case of one iteration), since it always requires the solution of two LPPs (see Algorithm 2, lines 1 and 3). Nevertheless, it provides better performance than Algorithm 1 when more than one iteration is carried out by the latter and, more importantly, it computes the exact optimal solution.

Of the two min-max solvers, indeed Algorithm 2 is certainly preferable, even though it implies solving the LPP P_0 first, since knowledge of the optimal solution vectors $\mathbf{v}^*$ and $\mathbf{y}^*$ enables the critical part of the TPNF (and hence of the modelled system) to be identified.

5 Vulnerability analysis of an oil pipeline network

As a case study, we consider the Saudi Arabian oil pipeline network that was a terrorist target in 2006 [35]. Figure 5 shows a map of Middle Eastern countries, including Saudi Arabia, where the oil and gas pipeline infrastructures are highlighted with green and red lines, respectively.

A previous vulnerability analysis of the network was conducted in [15]. Herein, we build a TPNF model of the network taking into consideration this previous analysis [15] and the report pulished in [35]. Afterwards, we apply the min-max technique to the TPNF model in order to analyze the impact of a coordinated attack on the network throughput. In particular, we show that the optimal solution of the min-max problem enables identification of critical parts of the network which should be protected against attacks.

5.1 The TPNF model

The Saudi Arabian oil pipeline network has been modelled with a TPNF, as shown in Figure 6. The *sources* place represents different oil fields that jointly produce 8-9 mmbbl/day[5] of crude oil. Saudi Arabia has three primary oil seaport terminals: Ras Tanura (on the Arabian Gulf, with 6 mmbbl capacity), Yanbu (on the Red Sea, with 4.5 mmbbl capacity) and Al-Ju'aymah (on the Persian Gulf, with 3 mmbbl capacity). Together with two minor terminals (Ras Al-Khafji and Jubail), these are represented by places in the TPNF model. The oil produced by the fields is distributed to the main pathways leading to the terminals, according to the capacity of the latter (this is taken into account by the multiplicities of the output arcs of transition *start*). The oil distribution network consists of pipelines and junctions with limited capacities, these being
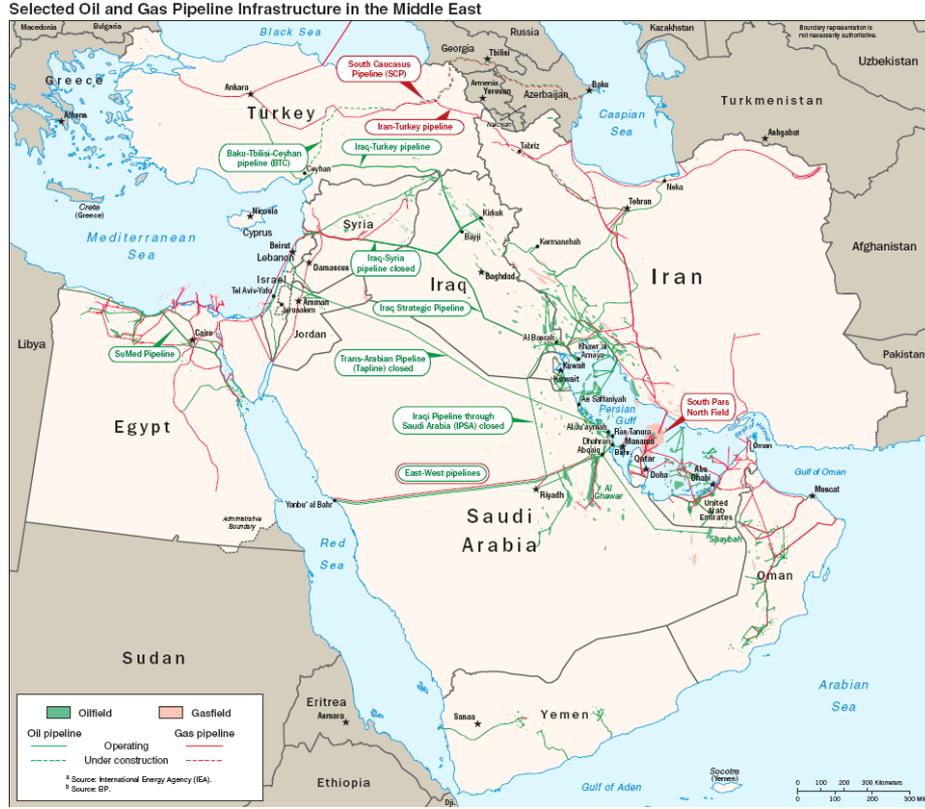
---

[5] millions of barrels per day.

Fig. 5. Oil and Gas Pipeline Infrastructure in the Middle East [35]

target of the attackers. They are modelled by the marked places p_i (pipes) and j_k (junctions). The oil flow from the sources to the terminals takes time and the delays are represented by the timed transitions $D_l p_i$ (pipe delays) and $D_l j_k$ (junction delays). For a given terminal, several alternative paths (modelled by free-choice transitions) can be taken by the oil flow. The choice of a path among the possible ones is constrained by the interval firing frequencies associated to the corresponding free-choice transitions. The *end* transition synchronizes the different paths and its throughput corresponds to the throughput of the network. We have computed the upper bound of the distribution network throughput by solving the problem P_0 associated to the TPNF model in Figure 6. The network throughput is 1.098901 (net_id 36, Table 2) and corresponds to 8.791208 mmbbl/day[6].

The goal of the terrorists is to decrease the network throughput to half of the current capacity, in order to cause worldwide economic distress. To achieve

---

[6] The quantity is given by number of oil barrels produced daily, 8 mmbbl/day, multiplied by the throughput.
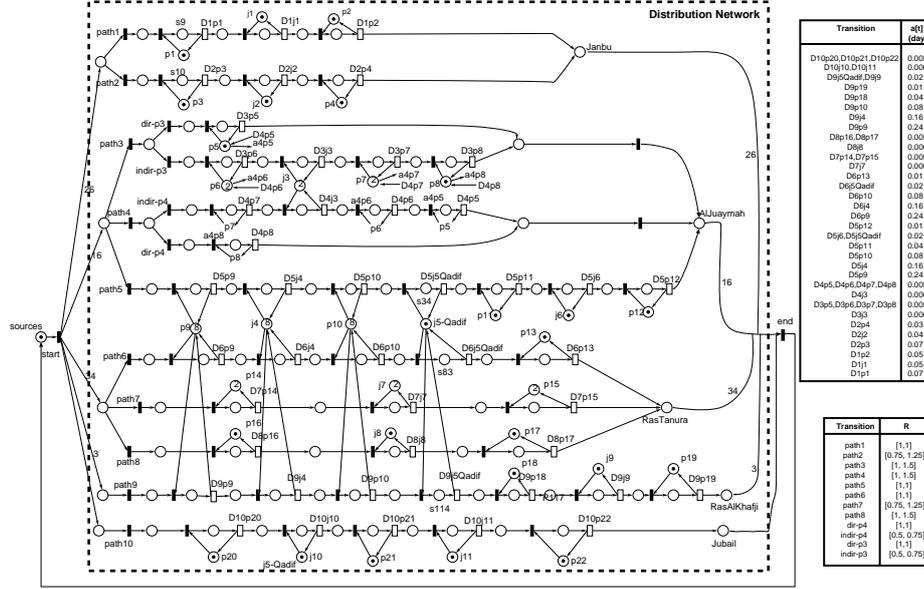
Fig. 6. TPNF model of the oil pipeline network.

their goal, terrorists draw up a coordinated attack that involve damaging as few facilities as possible. In particular, their objectives are the two pipes p_1 and p_3, leading to the Janbu terminal, and j_5, the Qadif junction [15]. As a consequence of an attack, the damaged facilities need to be repaired. Figure 7(A) sketches the TPNF model of the distribution network under a coordinated attack, where the two *clouds* represent TPNF subnets: the distribution network subnet (rounded by the dotted rectangle in Figure 6) and a coordinated attack & repair subnet. The latter models a coordinated attack on several targets and the subsequent repair work. For example, Figure 7(B) shows the *attack & repair* model in the case of a coordinated attack on three targets: the two pipes and the Qadif junction. In particular, the places `p1,p3` and `j5-Qadif` are shared with the distribution network subnet and the timed transitions Rp_1, Rp_2 and Rj_5 model the repair work. The metric of interest is the cycle time of the transition *start* (or, equivalently, its *end*) that corresponds to the traversal time of the token from the firing of the *start* until the firing of the *end*. Observe that, since the two subnets are concurrently enabled, after the firing of the *start*, the traversal time takes account of the delay(s) caused by the coordinated attack.

5.2 Application of the min-max technique

We aim to identify those facilities that mainly affect the network throughput in case of a coordinated attack. We have applied the min-max technique to different TPNF models representing different attack plans. The structure is shown in
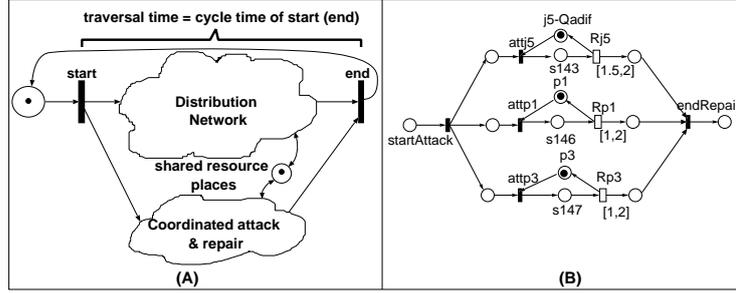
Fig. 7. (A) TPNF model of the attacked network. (B) TPNF subnet of a coordinated attack on three targets & repairs.

Figure 7(A). In particular, we consider that from one to three facilities can be concurrently attacked, so 6 TPNF models have been built (i.e., one per each possible attack plan) that are characterized by the same distribution network subnet and differ in the *attack & repair* subnet.

Table 3. Attack plans

Facilities	Throughput	$||\mathbf{y}^*||$	Path ratios		
			To Janbu	To Al-Ju'aymah	To Ras Tanura
			$\frac{v^*_{path_i}}{v^*_{path_1}}, i=1,2$	$\frac{v^*_{path_i}}{v^*_{path_5}}, i=3,4,5$	$\frac{v^*_{path_i}}{v^*_{path_6}}, i=6,7,8$
j_5	0.549048	$\{j_5, s34, s83, s114, s145\}$	$\{1; 0.75\}$	$\{1.5; 1.5; 1\}$	$\{1; 1.25; 1.5\}$
p_1	0.552826	$\{p_1, s9, s146\}$	$\{1; 1.25\}$	$\{1.5; 1.5; 1\}$	$\{1; 1.25; 1.5\}$
p_3	0.561798	$\{p_3, s10, s147\}$	$\{1; 0.75\}$	$\{1.5; 1.5; 1\}$	$\{1; 1.25; 1.5\}$
p_1, p_3	0.523560	$\{p_1, s9, s146\}$	$\{1; 1\}$	$\{1.5; 1.5; 1\}$	$\{1; 1.25; 1.5\}$
j_5, p_1	0.549048	$\{j_5, s34, s83, s114, s145\}$	$\{1; 1.25\}$	$\{1.5; 1.5; 1\}$	$\{1; 1.25; 1.5\}$
j_5, p_3	0.549048	$\{j_5, s34, s83, s114, s145\}$	$\{1; 0.75\}$	$\{1.5; 1.5; 1\}$	$\{1; 1.25; 1.5\}$
j_5, p_1, p_3	0.523560	$\{p_1, s9, s146\}$	$\{1; 1\}$	$\{1.5; 1.5; 1\}$	$\{1; 1.25; 1.5\}$

Table 3 summarizes the results of the analysis (each row refers to a different TPNF model), where the first and the second columns indicate the attacked facilities and the throughput of the attacked network, respectively. The third column shows the set of places of the TPNF model which support the optimal solution vector $\mathbf{y}^*$ of the min-max problem P_1. The subnets generated by such sets correspond to the slowest parts of the modeled system, i.e., they model the acquisition/release of the attacked facilities. The last three columns indicate the firing frequency ratios of the free-choice transitions representing the alternative paths from the sources to the three main terminals: Janbu, Al-Ju'aymah and Ras Tanura, respectively. Such ratios are computed considering the optimal solution vector $\mathbf{v}^*$. In all cases, the network throughput has been drastically reduced and, in the worst case, is equal to 0.523560 (Table 3, fourth and last rows),

meaning that the crude-oil output drops to 4.188480 mmbbl/day[7]. Then, the most damaging attack plans are either the simultaneous attack on the two pipes p_1 and p_3, or the attack on all the three facilities. Observe that for both plans the same slowest subnet is identified, corresponding to p_1. From the attackers point of view, the best plan is to damage just the two pipes.

The optimal solution vector $\mathbf{v}^*$ also gives useful information on the net behavior. Specifically, in the modeled system it provides suggestions on how to distribute the oil-flow between the alternative paths in order to reduce the economic loss due to an attack. In particular, we can observe that when just one of the two pipes p_1, p_3 is attacked, the oil-flow is directed as much as possible to the alternative path, according to the firing frequency interval restrictions associated to $path_1$ and $path_2$. On the other hand, in the case of a simultaneous attack on the two pipes, the best survivability solution is to split the oil-flow fifty-fifty between the two alternative pipelines to Janbu.

6 Conclusions

We have proposed a min-max problem to compute the lower bound cycle time of a transition t in a TPNF model. The min-max is an alternative to the LP-max problem previously stated for TPNF for the computation of the throughput upper bound of a transition.

We have formally proved that the optimal value of the min-max is the inverse of the optimal value of the LP-max. The main advantage of solving the former problem compared to the latter is that, besides the optimal value, the optimal solution also provides useful feedback to the analyst on the modeled system. In particular, the place variables indicate the slowest part of the system and the transition variables provide the *best* frequency assignment to the alternative behaviors that minimizes the cycle time of the slowest part.

We have implemented two solution algorithms for the min-max problem using *CPLEX* APIs, and have evaluated their performance considering a TPNF benchmark, built for the purpose.

Finally, we have used the min-max technique in a vulnerability analysis of the Saudi Arabian crude-oil distribution network, which has been a terrorist target in the past. The case study shows that the TPNF modeling formalism and its associated min-max problem technique are promising for the survivability analysis of critical systems and deserve further research.

As future work, we aim to study the applicability of the technique to other survivability-critical systems. Moreover, in order to provide support for the TPNF analysis under pessimistic assumptions, we plan to formulate similar optimization problems for computing the upper bound cycle time of a transition in a TPNF model.

---

[7] $= 8$ mmbbl/day $\cdot$ 0.523560.

A   Deriving the min-max from the LP-max problem

The min-max problem, introduced in Section 3, can be derived through a set of steps from the LP-max problem P_0 (1), recalled in Section 2. The former steps basically transform the initial LPP into equivalent LPPs, provided that the net is persistent, by considering the inverse relationship between the transition throughput and the transition cycle time. Afterwards, we apply a Lagrangian relaxation of the LP-max resulting from the previous steps. Reasoning on the meaning of the problem variables, we restrict the domain of the Lagrangian multipliers, thus leading to a min-max problem that provides an upper-bound of the minimum transition cycle time. Finally, we will prove that such a min-max problem is equivalent to the initial LP-max, i.e., an optimal value of the min-max is equal to the inverse of the optimal value of the initial LP-max.

A.1   Rewriting the initial LPP

The objective function and the linear constraints of the problem P_0 (1) can be rewritten in matrix form as follows:

$$
\begin{aligned}
P_0 \;=\; &\text{maximize } \mathbf{\chi_1^T x} \\
&s.t.\ \mathbf{M = M_0 + C^T}\sigma \quad \text{(reachability)} \\
&\quad\ \mathbf{C^T x = 0}_P \quad \text{(token flow)} \\
&\quad\ \mathbf{M} \geq (\mathbf{B^T} \odot \mathbf{a})\,\mathbf{x} \quad \text{(enabling operational law)} \\
&\quad\ \mathbf{Rx} \leq \mathbf{0}_K \quad \text{(routing)} \\
&\quad\ \mathbf{x}, \sigma \geq \mathbf{0}_T,\ \mathbf{M} \geq \mathbf{0}_P \quad \text{(non negativity)}
\end{aligned}
$$

where $\chi_1[i] = \begin{cases} 1 & i = t_1, \\ 0 & i \neq t_1. \end{cases}$ is the indicator function and $\mathbf{x}$ is the transition throughput vector.

The token flow constraints are expressed using the incidence matrix $\mathbf{C^T} = (\mathbf{F - B})^T$. Observe that the enabling operational law constraints can be written in such a matrix form, where $\odot$ is the component-wise product operator, under the assumption of net persistency. Indeed, for each place it is possible to write a unique enabling constraint. In particular, when a place is a pre-condition for several transitions, then the latter are immediate and the constraint boils down to a non negativity constraint for the marking variable associated to the place. The routing constraints are specified by a set of linear inequalities $\mathbf{Rx} \leq \mathbf{0}_K$, where $\mathbf{R}$ is the routing matrix, $K \times |T|$ sized positive real valued —$K = \sum_{k=1}^{n} 2(|ECS_k| - 1)$ and n is the number of equal conflict sets. The routing matrix is defined as follows:

$$
\mathbf{R} = \begin{pmatrix} \mathbf{R^1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{R^k} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R^n} \end{pmatrix} \quad \begin{matrix} 2(|ECS_1| - 1) \\ \vdots \\ 2(|ECS_k| - 1) \\ \vdots \\ 2(|ECS_n| - 1) \end{matrix}
$$

where each sub-matrix $\mathbf{R^k}$ defines the firing frequency intervals of the transitions in the conflict set ECS_k, with reference transition t_{k_0}. It has the following structure:

$$\mathbf{R^k} = \begin{array}{c} \begin{matrix} t_{k_0} & t_{k_1} & \ldots & t_{k_i} & \ldots & t_{k_{J_k}} \end{matrix} \\ \begin{pmatrix} -\overline{r}^{k_1} & 1 & \ldots & 0 & \ldots & 0 \\ \underline{r}^{k_1} & -1 & \ldots & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ -\overline{r}^{k_i} & 0 & \ldots & 1 & \ldots & 0 \\ \underline{r}^{k_i} & 0 & \ldots & -1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ -\overline{r}^{k_{J_k}} & 0 & \ldots & 0 & \ldots & 1 \\ \underline{r}^{k_{J_k}} & 0 & \ldots & 0 & \ldots & -1 \end{pmatrix} \end{array}$$

We can then eliminate the marking variables $\mathbf{M}$ by merging the reachability and the enabling operational law constraints:

$$
\begin{aligned}
P_0 \;=\; & \text{maximize } \chi_1^{\mathbf{T}} \mathbf{x} && (8) \\
& s.t. \; (\mathbf{B^T} \odot \mathbf{a})\,\mathbf{x} - \mathbf{C^T}\sigma \leq \mathbf{M_0} \quad \text{(merged constraints)} \\
& \qquad \mathbf{R}\mathbf{x} \leq \mathbf{0}_K \quad \text{(routing)} \\
& \qquad \mathbf{C^T}\mathbf{x} = \mathbf{0}_P \quad \text{(token flow)} \\
& \qquad \mathbf{x}, \sigma \geq \mathbf{0}_T \quad \text{(non negativity)}
\end{aligned}
$$

A.2 Stating an equivalent LPP

Let us now consider the dual of the LPP (8):

$$
\begin{aligned}
D(P_0) \;=\; & \text{minimize } \mathbf{M_0^T}\alpha_1 \\
& s.t. \; (\mathbf{a^T} \odot \mathbf{B})\alpha_1 + \mathbf{R^T}\alpha_2 + \mathbf{C}\alpha_3 \geq \chi_1 \\
& \qquad -\mathbf{C}\alpha_1 \geq \mathbf{0}_T \\
& \qquad \alpha_1 \geq \mathbf{0}_P, \; \alpha_2 \geq \mathbf{0}_K, \; \alpha_3 \gtrless \mathbf{0}_P
\end{aligned}
$$

Since the net is live, the objective function of $D(P_0)$ is strictly positive. Then, the dual problem is equivalent to maximizing, over the same set of constraints, the fractional problem with the objective function $\frac{1}{\mathbf{M_0^T}\alpha_1}$. Let us consider, then, the change of variables: $\mathbf{y_i} = \alpha_i t$, where $i = 1, 2, 3$ and $t > 0$. The fractional problem is transformed into the following LP-max:

$$
\begin{aligned}
Q \;=\; & \text{maximize } t && (9) \\
& s.t. \; \chi_1 t - (\mathbf{a^T} \odot \mathbf{B})\mathbf{y_1} - \mathbf{R^T}\mathbf{y_2} - \mathbf{C}\mathbf{y_3} \leq \mathbf{0}_T && (10) \\
& \qquad \mathbf{C}\mathbf{y_1} \leq \mathbf{0}_T && (11) \\
& \qquad \mathbf{M_0^T}\mathbf{y_1} = 1 && (12) \\
& \qquad t > 0, \mathbf{y_1} \geq \mathbf{0}_P, \mathbf{y_2} \geq \mathbf{0}_K, \mathbf{y_3} \gtrless \mathbf{0}_P && (13)
\end{aligned}
$$

A.3 Applying a Lagrangian relaxation

We relax the LP-max problem (9) by eliminating the T constraints (10) and by taking them into account in the objective function. Let $\mathbf{v} \in \mathbb{R}_0^+$ be the T-sized vector of Lagrangian multipliers, then the relaxed problem is:

$$L_{\mathbf{v}}(Q) \;=\; \text{maximize } t(1 - \chi_\mathbf{1}^\mathbf{T}\mathbf{v}) + \mathbf{y_1}^\mathbf{T}(\mathbf{B^T} \odot \mathbf{a})\, \mathbf{v} +$$
$$+\mathbf{y_2^T R v} + \mathbf{y_3^T C^T v}$$
$$s.t.\; \mathbf{y_1}, \mathbf{y_2}, \mathbf{y_3}, t \in D$$

where D is the admissible region defined by the set of remaining constraints (11,12,13) and the (positive) term added to the objective function of problem Q is a linear combination of the constraints (10). The objective function of the relaxed problem is greater than (or equal to) the objective function of the LP-max Q, over the admissible region of Q, so the optimal solution of $L_{\mathbf{v}}(Q)$ provides an upper bound for the optimal solution of Q, for all $\mathbf{v} \in \mathbb{R}_0^+$.

Now, the interesting problem is to identify the Lagrangian multipliers that guarantee the *best* upper bound, that is:

$$L_{\mathbf{v}^*}(Q) = \min_{\mathbf{v} \geq \mathbf{0}} L_{\mathbf{v}}(Q).$$

A.4 Restricting the domain of the Lagrangian multipliers

We can first observe that the domain D in $L_{\mathbf{v}}(Q)$ identifies the annullers $\mathbf{y_1}$ of $\mathbf{C}$. Indeed, by assumption, the net is live and inequality constraints (11) can be replaced by equality ones, i.e., $\mathbf{Cy_1} = \mathbf{0}_T$. Then, the weighted sum of tokens circulating in the set of places $\|\mathbf{y_1}\|$ is constant, and —by constraint (12)— in all reachable markings, $y_1[p]$ represents the ratio of tokens in p.

Secondly, the term $\mathbf{y_1^T}(\mathbf{B^T} \odot \mathbf{a})\, \mathbf{v} \geq 0$ in $L_{\mathbf{v}}(Q)$ evaluates the cycle time of the transitions belonging to the P-subnet generated by the set $\|\mathbf{y_1}\|$ if and only if $\mathbf{v}$ is the vector of visit ratios associated to the transitions in the P-subnet.

Then, we choose the transition visit ratios as Lagrangian multipliers. This choice leads to introducing into the problem $L_{\mathbf{v}}(Q)$ the following restrictions: $D_v = \left\{ \mathbf{Rv} \leq \mathbf{0}_K, \mathbf{C^T v} = \mathbf{0}_P, v[t_1] = 1, \mathbf{v} \geq \mathbf{0}_T \right\}$, where $v[t_1]$ is the visit ratio associated to the transition of reference in the net.

Consequently, the first and the last term in $L_{\mathbf{v}}(Q)$ i.e., $t(1-\chi_\mathbf{1}^\mathbf{T}\mathbf{v}) = t(1-v[t_1])$ and $\mathbf{y_3^T C^T v}$, are null in the admissible region. The term $\mathbf{y_2^T R v}$ is not positive in the admissible region. Then, for a fixed visit ratio $\mathbf{v} \in D_v$, a (maximizing) optimal solution for the problem $L_{\mathbf{v}}(Q)$ is characterized by $\mathbf{y_2} = \mathbf{0}$. Considering the restrictions on the Lagrangian multipliers, we can rewrite the problem as:

$$P_1 \;=\; \min_{\mathbf{v} \in D_v} \; \max_{\mathbf{y} \in D_y} \; \mathbf{y^T}(\mathbf{B^T} \odot \mathbf{a})\, \mathbf{v} \tag{14}$$
$$s.t.\; D_y : \left\{ \mathbf{Cy} = \mathbf{0}_T, \mathbf{M_0^T y} = 1, \mathbf{y} \geq \mathbf{0}_P \right\}$$
$$D_v : \left\{ \mathbf{Rv} \leq \mathbf{0}_K, \mathbf{C^T v} = \mathbf{0}_P, v[t_1] = 1, \mathbf{v} \geq \mathbf{0}_T \right\}$$

Finally, the following relations hold among the stated problems:

$$P_1 \geq L_{\mathbf{v}^*}(Q) = D(Q) \geq Q = \frac{1}{D(P_0)}$$

where $D(Q)$ is the dual of Q. It is well known from the literature that the problem of finding the optimal Lagrangian multipliers $\mathbf{v}^*$ is equivalent to solving $D(Q)$. On the other hand, observe that we can assert that P_1 is an upper bound problem of $L_{\mathbf{v}^*}(Q)$ since the optimal Lagrangian multipliers $\mathbf{v}^*$ are necessarily transition visit ratios. Indeed, if one of the constraints in D_v were not satisfied, then $L_{\mathbf{v}^*}(Q)$ would be unbounded.

A.5 Equivalence with the initial LP-max

The next step is to prove that the min-max problem P_1 (14) is actually equivalent to the initial LP-max problem P_0, that is the optimal value of P_1 is equal to the inverse of the optimal value of P_0 (and viceversa). For this purpose, let us consider the dual problem of Q:

$$D(Q) \;=\; min \; \gamma \tag{15}$$
$$s.t. \; x[t_1] \geq 1 \tag{16}$$
$$-(\mathbf{B^T} \odot \mathbf{a})\, \mathbf{x} + \mathbf{C^T}\sigma + \mathbf{M_0}\gamma \geq \mathbf{0}_P \tag{17}$$
$$-\mathbf{R}\mathbf{x} \geq \mathbf{0}_K \tag{18}$$
$$\mathbf{C^T}\mathbf{x} = \mathbf{0}_P \tag{19}$$
$$\mathbf{x}, \sigma \geq \mathbf{0}_T, \gamma \gtreqless 0 \tag{20}$$

A Lagrangian relaxation of $D(Q)$ is obtained by subtracting from the objective function a linear combination of the P contraints (17) and by eliminating the latter from the set of constraints:

$$L_{\mathbf{y}}(D(Q)) \;=\; \text{minimize } (1 - \mathbf{y^T M_0})\gamma + \mathbf{y^T}(\mathbf{B^T} \odot \mathbf{a})\, \mathbf{x} - \mathbf{y^T C^T}\sigma$$
$$s.t. \; \gamma, \mathbf{x}, \sigma \in D$$

where D is the admissible region, defined by constraints (16,18,19,20), and $\mathbf{y} \geq \mathbf{0}$ are the non negative Lagrangian multipliers. The optimal solution of $L_{\mathbf{y}}(D(Q))$ provides a lower bound for $D(Q)$ and the Lagrangian multipliers that guarantee the *best* lower bound are solutions of the problem:

$$L_{\mathbf{y}^*}(D(Q)) = \max_{\mathbf{y} \geq \mathbf{0}} L_{\mathbf{y}}(D(Q)).$$

Now, reasoning on the meaning of the problem variable as we did for the Lagrangian relaxation of Q, the Lagrangian multipliers $\mathbf{y}$ must satisfy the following constraints: $D_y = \{\mathbf{C}\mathbf{y} = \mathbf{0}_T, \mathbf{M_0^T}\mathbf{y} = 1, \mathbf{y} \geq \mathbf{0}_P\}$. We can then rewrite the problem as:

$$P_2 \;=\; \max_{\mathbf{y} \in D_y} \; \min_{\mathbf{x} \in D_x} \; \mathbf{y^T}(\mathbf{B^T} \odot \mathbf{a})\, \mathbf{x} \tag{21}$$
$$s.t. \; D_x : \{\mathbf{C^T}\mathbf{x} = \mathbf{0}_P, \mathbf{R}\mathbf{x} \leq \mathbf{0}_K, x[t_1] \geq 1, \mathbf{x} \geq \mathbf{0}_T\}$$
$$D_y : \{\mathbf{C}\mathbf{y} = \mathbf{0}_T, \mathbf{M_0^T}\mathbf{y} = 1, \mathbf{y} \geq \mathbf{0}_P\}.$$

Observe that the problems P_1 and P_2 can be converted one into the other considering the change of variables $\mathbf{v} = \mathbf{x}/x[t_1]$ and the following relations hold:

$$P_1 \geq D(Q) \geq Q \geq P_2. \tag{22}$$

Proposition 4 *The two problems P_1 and P_2 are equivalent.*

Proof. We prove the equivalence by showing that an optimal solution $(\tilde{\mathbf{y}}, \tilde{\mathbf{x}})$ of P_2 satisfies the property $\tilde{x}[t_1] = 1$. Let us assume, by contradiction, that there exists an optimal solution $(\tilde{\mathbf{y}}, \tilde{\mathbf{x}})$ such that $\tilde{x}[t_1] > 1$. Then $(\tilde{\mathbf{y}}, \tilde{\mathbf{v}})$ —where $\tilde{\mathbf{v}} = \frac{\tilde{\mathbf{x}}}{\tilde{x}[t_1]}$— is an admissible solution of P_1 and:

$$
\begin{aligned}
\Phi(\tilde{\mathbf{y}}, \tilde{\mathbf{v}}) &= \tilde{\mathbf{y}}^{\mathbf{T}}(\mathbf{B}^{\mathbf{T}} \odot \mathbf{a})\ \tilde{\mathbf{v}} \\
&= \tilde{\mathbf{y}}^{\mathbf{T}}(\mathbf{B}^{\mathbf{T}} \odot \mathbf{a})\ \frac{\tilde{\mathbf{x}}}{\tilde{x}[t_1]}\ <\ \tilde{\mathbf{y}}^{\mathbf{T}}(\mathbf{B}^{\mathbf{T}} \odot \mathbf{a})\ \tilde{\mathbf{x}}\ = \\
&= \Psi(\tilde{\mathbf{y}}, \tilde{\mathbf{x}})
\end{aligned}
$$

where Φ, Ψ are the objective functions of P_1 and P_2, respectively. This contradicts the relation (22), i.e., $P_1 \geq P_2$.

Summarizing:

- Every optimal solution $(\tilde{\mathbf{y}}, \tilde{\mathbf{x}})$ of P_2 is an optimal solution $(\tilde{\mathbf{y}}, \tilde{\mathbf{v}})$ of P_1 and viceversa, since $\tilde{\mathbf{x}} = \tilde{\mathbf{v}}$, and:

$$\Phi(\tilde{\mathbf{y}}, \tilde{\mathbf{v}}) = F(\tilde{t}, \tilde{\mathbf{y}}, \mathbf{0}, \tilde{\mathbf{y}}_3) = G(\tilde{\gamma}, \tilde{\sigma}, \tilde{\mathbf{x}}) = \Psi(\tilde{\mathbf{y}}, \tilde{\mathbf{x}})$$

  where Φ, F, G and Ψ are the objective functions of the problems P_1, Q, $D(Q)$ and P_2, respectively.
- Since $Q = 1/D(P_0)$, the optimal value of the LP-max problem P_0 is equal to $\frac{1}{\Phi(\tilde{\mathbf{y}}, \tilde{\mathbf{v}})}$, where $(\tilde{\mathbf{y}}, \tilde{\mathbf{v}})$ is an optimal solution of P_1.

References

1. S. Bernardi and J. Campos, "Computation of performance bounds for real-time systems using Time Petri Nets," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 2, pp. 168–180, May 2009.
2. S. Bernardi, J. Campos, and J. Merseguer, "Timing-failure risk assessment of UML design using Time Petri Net bound techniques," *IEEE Trans. on Industrial Informatics*, vol. 7, no. 1, pp. 90–104, Feb. 2011.
3. S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.
4. J. Campos and M. Silva, "Structural techniques and performance bounds of stochastic Petri net models," *LNCS*, vol. 609, pp. 352–391, 1992.
5. IBM, "ILOG CPLEX Optimizer," 2011.
6. S. Bernardi and J. Campos, "On performance bounds for interval Time Petri Nets," in *Proceedings of QEST, 27-30 September 2004, Enschede, The Netherlands*. IEEE Computer Society, 2004, pp. 50–59.

7. J. Campos, "Performance bounds for synchronized queueing networks," Ph.D. dissertation, Universidad de Zaragoza, Spain, Research Report GISI-RR-90-20, October 1990.

8. E. P. Naumovich, S. Bernardi, and M. Gribaudo, "ITPN-PerfBound: A performance bound tool for interval Time Petri Nets," in *Proceedings of TACAS. York, UK, March 22-29*, ser. LNCS, S. Kowalewski and A. Philippou, Eds., vol. 5505. Springer, 2009, pp. 50–53.

9. L. Berardinelli, S. Bernardi, V. Cortellessa, and J. Merseguer, "The Fault-error-failure Chain: a Challenge for Modeling and Analyzing Performability in UML-based Software Architectures," 2010, technical Report, Università dell' Aquila.

10. J. Merseguer, J. Campos, and E. Mena, "Analysing internet software retrieval systems: Modeling and performance comparison," *Wireless Networks: The Journal of Mobile Communication Computation and Information*, vol. 9, pp. 223–238, 5 2003.

11. A. Zimmermann, *Stochastic Discrete Event Systems.* Springer, 2007.

12. E. Gómez-Martínez and J. Merseguer, "Performance modeling and analysis of the Universal Control Hub," in *Computer Performance Engineering - EPEW, Bertinoro, Italy, September 23-24.*, ser. LNCS, A. Aldini, M. Bernardo, L. Bononi, and V. Cortellessa, Eds., vol. 6342. Springer, 2010, pp. 160–174.

13. ——, "Impact of SOAP implementations in the performance of a web service-based application," in *Proceedings of ISPA, Sorrento, Italy, December 4-7*, ser. LNCS, G. Min, B. D. Martino, L. T. Yang, M. Guo, and G. Rünger, Eds., vol. 4331. Springer, 2006, pp. 884–896.

14. E. Gómez-Martínez, S. Ilarri, and J. Merseguer, "Performance analysis of mobile agents tracking," in *Proceedings of WOSP, Buenos Aires, Argentina, February 5-8*, V. Cortellessa, S. Uchitel, and D. Yankelevich, Eds. ACM, 2007, pp. 181–188.

15. G. G. Brown, W. M. Carlyle, J. Salmerón, and K. Wood, "Analyzing the vulnerability of critical infrastructure to attack and planning defenses," in *Tutorials in Operations Research*. INFORMS, 2005, pp. 102–123.

16. B. Berthomieu and M. Diaz, "Modeling and Verification of Time Dependent Systems Using Time Petri Nets," *IEEE Trans. on Soft. Eng.*, vol. 12, no. 3, pp. 259–273, March 1991.

17. E. Vicario, "Static Analysis and Dynamic Steering of Time-Dependent Systems," *IEEE Trans. on Soft. Eng.*, vol. 27, no. 8, pp. 728–748, August 2001.

18. J. Wang, Y. Deng, and G. Xu, "Reachability Analysis of Real-Time Systems Using Time Petri Nets," *IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 30, no. 5, pp. 725–736, Oct. 2000.

19. D. Xu, X. He, and Y. Deng, "Compositional Schedulability Analysis of Real-Time Systems Using Time Petri Nets," *IEEE Trans. on Software Engineering*, vol. 28, no. 10, pp. 984–996, October 2002.

20. D. Lime and O. Roux, "A Translation Based Method for the Timed Analysis of Scheduling Extended Time Petri Nets," in *Proc. RTSS*. IEEE Computer Society, 2004, pp. 187–196.

21. G. Bucci, S. Sassoli, and E. Vicario, "Correctness verification and performance analysis of real-time systems using Stochastic Preemptive Time Petri nets," *IEEE Trans. on Soft. Eng.*, vol. 31, no. 11, pp. 913–921, Nov. 2005.

22. S. Han and H. Y. Youn, "Modeling and Analysis of Time-Critical Context-Aware Service Using Extended Interval Timed Colored Petri Nets," *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 3, pp. 630–640, May 2012.

23. H. Wang and Q. Zeng, "Modeling and Analysis for Workflow Constrained by Resources and Nondetermined Time: An Approach Based on Petri Nets," *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 38, no. 4, pp. 802 –817, July 2008.

24. J. Campos, G. Chiola, J. Colom, and M. Silva, "Properties and performance bounds for timed marked graphs," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 39, no. 5, pp. 386–401, May 1992.

25. G. Chiola, C. Anglano, J. Campos, J. Colom, and M. Silva, "Operational analysis of timed Petri nets and application to the computation of performance bounds," in *Proceedings of PNPM*. Toulouse, France: IEEE CS, October 1993, pp. 128–137.

26. Z. Liu, "Performance Analysis of Stochastic Timed Petri Nets Using Linear Programming Approach," *IEEE Trans. on Soft. Eng.*, vol. 24, no. 11, November 1998.

27. R. J. Rodríguez and J. Júlvez, "Accurate Performance Estimation for Stochastic Marked Graphs by Bottleneck Regrowing," in *Proceedings of the 7th European Performance Engineering Workshop (EPEW)*, ser. LNCS, vol. 6342. Springer, September 2010, pp. 175–190.

28. A. Ramírez-Treviño, E. Ruiz-Beltrán, J. Arámburo-Lizárraga, and E. López-Mellado, "Structural Diagnosability of DES and Design of Reduced Petri Net Diagnosers," *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 3, pp. 416–429, March 2012.

29. H. Ohl, *Fonctionnements répétitifs de systèmes flexibles de production manufacturière: analyse et optimisation des performances à l'aide de réseaux de Petri.* Univ. des Sciences et Technologies, 1995.

30. M. Di Loreto, S. Gaubert, R. Katz, and J. Loiseau, "Duality between invariant spaces for max-plus linear discrete event systems," *SIAM J. Control. Optim.*, vol. 48, no. 8, pp. 5606–5628, 2010.

31. B. Trouillet, O. Korbaa, and J.-C. Gentina, "Formal approach of fms cyclic scheduling," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 1, pp. 126 –137, jan 2007.

32. A. El Amraoui, M. Manier, A. El Moudni, and M. Benrejeb, "A mixed linear program for a multi-part cyclic hoist scheduling problem," *International Journal on Sciences and Techniques of Automatic Control & Computer Engineering*, vol. 2, pp. 612–623, December 2008.

33. L. Recalde, E. Teruel, and M. Silva, "On Linear Algebraic Techniques for Liveness Analysis of PT Systems," *Journal of Circuits Systems and Computers*, vol. 8, no. 1, pp. 223–265, 1998.

34. P. Denning and J. Buzen, "The Operational Analysis of Queueing Network Models," *ACM Comput. Surv.*, vol. 10, no. 3, pp. 225–261, 1978.

35. USEIA, "Country Analysis Brief - Saudi Arabia," U.S. Energy Information Administration, Independent Statistics & Analysis, Tech. Rep., January 2011. [Online]. Available: http://www.eia.gov/cabs/Saudi_Arabia/Full.html