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

Simona Bernardi (Centro Universitario de la Defensa, Zaragoza)
Javier Campos (Universidad de Zaragoza)

* Accepted in IEEE Transactions on Systems, Man, and Cybernetics: Systems

JCSD 2013
XXI Jornadas de Concurrencia y Sistemas Distribuidos

19-21 de junio de 2013
Donostia-San Sebastián
2006: Al-Qaeda attempted suicide bomb attack at the Abqaiq petroleum processing facility
Vulnerability analysis of the network

- Analyze the impact of a coordinated attack on the network throughput
  - Maximize throughput after attack
  - Identify critical paths
  - How to distribute the oil-flow between the alternative paths to maximize throughput
  - Reduce the economic loss due to an attack
Interval time Petri net, i.e. when enabled, $t$ fires within interval $[a, b]$ of time.

Routing intervals, i.e. per each firing of $t_1$, $t_2$ must fire in average a number of times in $[r, s]$.
Sources: oil fields (jointly produce 8-9 mmbbl/day of crude oil)
Seaport terminals: Ras Tanura (6 mmbbl), Janbu (4.5 mmbbl), Al Ju’aymah (3 mmbbl)
Example of two critic pipes (to Janbu) and a critic junction (Qadif)
(A) Petri net model of the attacked network.

\[
\text{traversal time} = \frac{1}{\text{throughput of transition 'end'}}
\]

(B) Subnet of a coordinated attack on three targets (Qadif junction and Janbu pipes) and repairs.
The Problem in *PN model terms*:

- Test different coordinated attack scenarios.
- Identify worst coordinated attack
  
  = *find attacked network with minimum throughput, and for it*:

  1. Compute the throughput
     
     = *throughput of transition “end”*

  2. Identify critical points
     
     = *bottleneck subnets*

  3. Compute optimum oil paths
     
     = *optimum routing ratios at choice places*
Before this work

• Problem 1. Compute the throughput
  = throughput of transition “end”, efficiently solved:

**Proposition 1**\(^(*)\) Let \(\mathcal{T} \mathcal{F} = (\mathcal{T}, \mathcal{R})\) 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:

\[
P_0 = \text{maximize } x[t_1] \\
\text{s.t. } M = M_0 + C^T \sigma \\
\sum_{t \in \cdot P} x[t] F[t, p] = \sum_{t \in \cdot P} x[t] B[t, p], \forall p \in P \\
M[p] \geq x[t] a[t] B[t, p], \forall t \in T \text{ and } \forall p \in \cdot t \\
r^j x[t_k] \leq \bar{r}^k x[t_j], \; \bar{r}^j x[t_j] \leq \bar{r}^j x[t_k], \forall t_j, t_k \in ECS \\
x, \sigma \geq 0_T, \; M \geq 0_P
\]

Before this work

• Problem 2. Identify critical points
  = bottleneck subnets

  Not (efficiently) solved

• Problem 3. Compute optimum oil paths
  = optimum routing ratios at choice places

  Not (efficiently) solved
After this work

- Problem 1. Compute the throughput
  \(= \text{throughput of transition "end"}, \) efficiently solved:

**Proposition 2** Let \(\mathcal{T}, \mathcal{F} = (\mathcal{T}, \mathcal{R})\) 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:

\[
P_1 = \min_{v \in D_v} \max_{y \in D_y} y^T (B^T \odot a) v
\]

\[
s.t. D_y : \{Cy = 0_T, M_0^T y = 1, y \geq 0_P\}
\]

\[
D_v : \{Rv \leq 0_K, C^T v = 0_P, v[t_1] = 1, v \geq 0_T\}
\]
After this work

• Problem 2. Identify critical points
  = *bottleneck subnets*, efficiently solved:

  Support of vector \( y^* \)
  of the optimal solution of the previous problem

• Problem 3. Compute optimum oil paths
  = *optimum routing ratios at choice places*,
  efficiently solved:

  Vector \( v^* \)
  of the optimal solution of the previous problem
Results (in this case study)

Worst coordinated attack with minimum risk for terrorists

| Facilities | Throughput | $||y^*||$ | Path ratios |
|------------|------------|----------|-------------|
|            |            |          | To Janbu $v_{path_i}^{*}$, $i = 1, 2$ | To Al-Ju’aymah $v_{path_i}^{*}$, $i = 3, 4, 5$ | To Ras Tanura $v_{path_i}^{*}$, $i = 6, 7, 8$ |
| $j_5$      | 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 | $\{p_1, s9, s146\}$ | $1; 1$ | $1.5; 1.5; 1$ | $1; 1.25; 1.5$ |

- **targets attacked**
- **bottleneck subnets**
- **inverse of traversal time**
- **optimum routing ratios at choice places**
Technically speaking

The main result of the paper is in the 5-pages Appendix:

\[ P_1 = \frac{1}{P_0} \]

where

\[ P_0 = \maximize \ x[t_1] \]
\[ \text{s.t.} \ M = M_0 + C^T \sigma \]
\[ \sum_{t \in \bullet_p} x[t] F[t, p] = \sum_{t \in \bullet_p^*} x[t] B[t, p], \forall p \in P \]
\[ M[p] \geq x[t] a[t] B[t, p], \forall t \in T \text{ and } \forall p \in \bullet t \]
\[ r^j x[t_k] \leq \bar{r}^j x[t_j], \ r^k x[t_j] \leq \bar{r}^j x[t_k], \forall t_j, t_k \in ECS \]
\[ x, \sigma \geq 0_T, \ M \geq 0_P \]

\[ P_1 = \min_{v \in D_v} \ \max_{y \in D_y} \ y^T (B^T \odot a) \ v \]
\[ \text{s.t.} \ D_y : \{ Cy = 0_T, M_0^T y = 1, y \geq 0_P \} \]
\[ D_v : \{ Rv \leq 0_K, C^T v = 0_P, v[t_1] = 1, v \geq 0_T \} \]
Additional results

• Proposal of two algorithms to solve the new programming problem ($P_1$):
  
  – An approximate sub-gradient method
  – Another exact method that previously requires the solution of $P_0$

• A benchmark of PN models to compare both solution algorithms
  
  – 40 Time PN models, several of them being case studies taken from the literature
Questions?