Simulations of Conflict-or-Increase Basis Reachability Graphs

Chao Gu, Ziyue Ma, Zhiwu Li and Alessandro Giua

March 3, 2021

In this note, we use a parameterized system modelled by Petri net to test the efficiency of our approach based on the conflict-or-increase basis reachability graph (CI-BRG). All tests are carried out on a PC with an Intel Core i7-7700 CPU 3.60 GHz processor and 8.00 GB RAM.

The Simulation Results

As shown in Fig. 1, this benchmark is a parameterized plant selected from [1] while we make some minor adjustments, i.e., two self-looped arcs $t_{33} \leftrightarrow p_6$ and $t_{36} \leftrightarrow p_{21}$ and one directed arcs $t_{39} \to p_{46}$ are removed. This system consists of 46 places and 39 transitions, where the initial marking M_0 is parameterized as: $M_0 = \alpha p_1 + \beta p_{16} + p_{31} + p_{32} + p_{33} + p_{34} + p_{35} + p_{37} + p_{38} + p_{39} + 8p_{40} + p_{41}$. Let $\mathscr{F} = \mathscr{L}_{(\mathbf{w},k)} = \{M \in \mathbb{N}^m | \mathbf{w}^T \cdot M \leq k\}$, where

$$\mathbf{w} = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$$

and $k = 4$ (for runs $1-4$) or $k = 7$ (for runs $5-8$) to test non-blockingness of this plant for all cases.

Firstly, it can be inferred that $T_{conf} = \{t_6, t_7, t_8, t_9, t_{13}, t_{14}, t_{15}, t_{21}, t_{22}, t_{23}, t_{24}, t_{28}, t_{29}, t_{30}, t_{31}, t_{32}, t_{33}, t_{34}, t_{35}, t_{36}, t_{37}, t_{38}, t_{39}\}$ (marked in shadow) and $T_{inc} = \{t_1, t_7, t_{22}, t_{24}, t_{35}\}$ (boxed in red). Since the subnet induced by all transitions $t \in T \setminus (T_{conf} \cup T_{inc})$ is acyclic, a non-conflicting and non-increasing T_I can be obtained, i.e., $T_I = T \setminus (T_{conf} \cup T_{inc})$ and therefore $T_E = T \setminus T_I$. Thus, the corresponding CI-BRG $\mathscr{B}$ can be constructed based on $\pi = (T_E, T_I)$.

Moreover, for comparison, we compute the minimax-BRG [2] of the plant (denoted as $\mathscr{B}_{\mathscr{M}}$) in Fig. 1 by considering another basis partition $\pi' = (T'_E, T'_I)$ where $T'_E = \{t_1, t_6, t_7, t_9, t_{14}, t_{21}, t_{23}, t_{24}, t_{29}, t_{32}, t_{34}, t_{35}\}$ and $T'_I = T \setminus T'_E$. The set of all minimax basis markings in $\mathscr{B}_{\mathscr{M}}$ is denoted as $\mathscr{M}_{\mathscr{B}_{\mathscr{M}}}$.

In Table 1, for different values of α and β, the number of basis markings in $\mathscr{M}_{\mathscr{B}}$ in the CI-BRG $\mathscr{B}$, minimax basis markings in $\mathscr{M}_{\mathscr{B}_{\mathscr{M}}}$ in the minimax-BRG $\mathscr{B}_{\mathscr{M}}$, and all reachable markings $M \in R(N, M_0)$, as well as their computing times are listed in columns $1-9$. Meanwhile, the non-blockingness of each case and the ratios of node number and time assumptions in terms of reachability graph and $\mathscr{B}$ are respectively reported in columns $10-12$.

Through the results, we conclude that the CI-BRG-based approach outperforms that of the reachability-graph-based method in this plant for all cases; whereas, with the expansion of the system scale, CI-BRG shows the potential to be more efficient than the minimax-BRG in this study.

Table 1: Analysis of the reachability graph, minimax-BRG, and CI-BRG for the plant in Fig. 1.

Run	α	β	$\lvert R(N,M_0)\rvert$	Time (s)	$\lvert\mathscr{M}_{\mathscr{B}_\mathscr{M}}\rvert$	Time (s)	$\lvert\mathscr{M}_\mathscr{B}\rvert$	Time (s)	Non-blocking?	$\lvert\mathscr{M}_\mathscr{B}\rvert/\lvert R(N,M_0)\rvert$	Time ratio
1	1	1	1966	10	284	2	604	1.7	Yes	30.7%	17%
2	1	2	12577	277	1341	15	2145	11	Yes	17%	4%
3	2	2	76808	12378	5961	179	7718	105	No	10%	0.8%
4	2	3	-	o.t.	14990	1028	16438	470	No	-	-
5	2	4	-	o.t.	26716	3126	26648	1248	Yes	-	-
6	3	3	-	o.t.	38551	6697	37118	2492	Yes	-	-
7	3	4	-	o.t.	67728	22018	59315	6449	No	-	-
8	4	4	-	o.t.	-	o.t.	101420	19491	No	-	-

* The computing time is denoted by *overtime* (o.t.) if the program does not terminate within 36,000 seconds (10 hours).

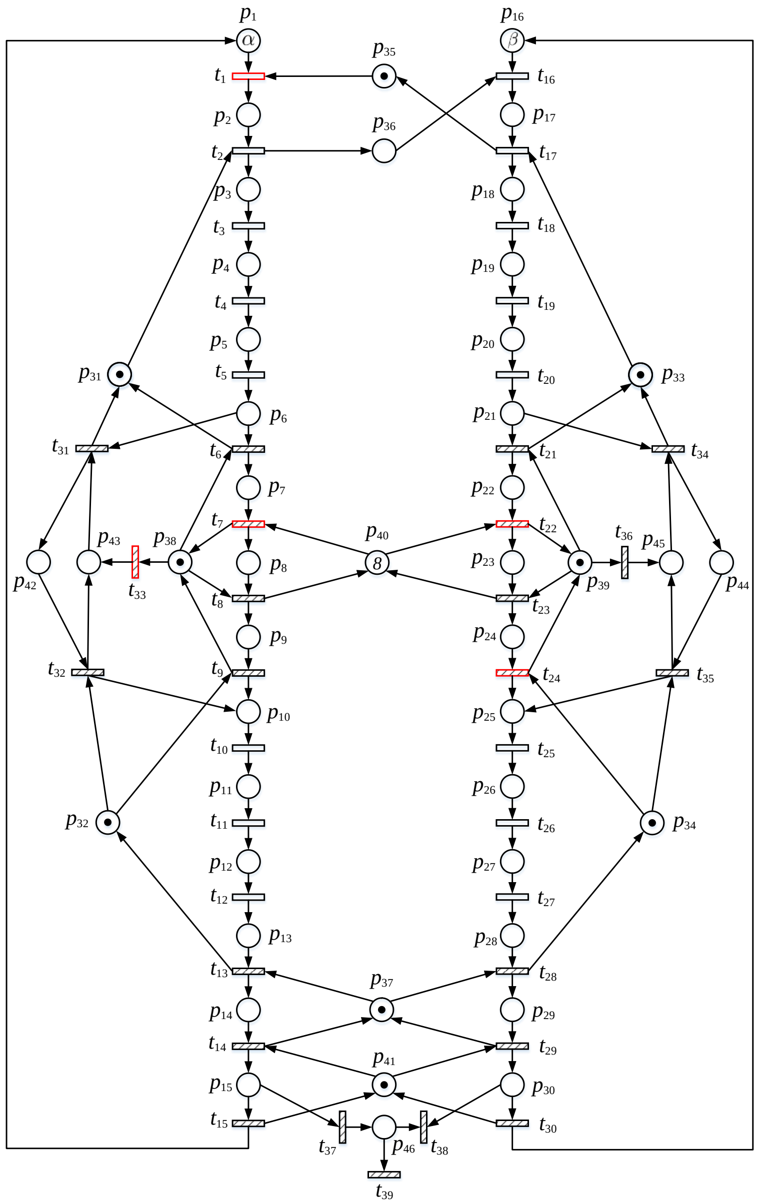

Figure 1: A parameterized plant $G = (N, M_0, \mathscr{F})$.

References

[1] M. P. Cabasino, A. Giua, M. Pocci, and C. Seatzu. Discrete event diagnosis using labeled Petri nets. an application to manufacturing systems. *Control Engineering Practice*, 19(9):989–1001, 2011.

[2] C. Gu, Z. Y. Ma, Z. W. Li, and A. Giua. Verification of nonblockingness in bounded Petri nets: a novel semi-structural approach. *arXiv preprint*, arXiv:2003.14204, 2020.