Avoiding Exponential Explosion in Petri Net Models of Control Flows

Victor Khomenko[1(✉)] , Maciej Koutny[1] , and Alex Yakovlev[2]

[1] School of Computing, Newcastle University,
1 Science Square, Newcastle upon Tyne NE4 5TG, UK
{victor.khomenko,maciej.koutny}@ncl.ac.uk
[2] School of Engineering, Newcastle University,
Merz Court, Newcastle upon Tyne NE1 7RU, UK
alex.yakovlev@ncl.ac.uk

Abstract. We look at modelling of a choice between several 'bursts' of concurrent actions in a Petri net. If 'silent' transitions are disallowed, a construction based on Cartesian product is traditionally used, resulting in an exponential explosion in the model size.

We demonstrate that this exponential explosion can be avoided. We show the equivalence between this modelling problem and the problem of finding an edge clique cover of a complete multipartite graph, which gives major insights into the former problem as well as linking it to the existing results from graph theory.

It turns out that the exponential number of places created by the Cartesian product construction can be improved down to polynomial (quadratic) in the worst case, and down to logarithmic in the best (non-degraded) case. For example, to express a choice between 10 pairs of concurrent transitions, the Cartesian product construction creates 1024 places, even though 6 places are sufficient. We also derive several lower and upper bounds on the numbers of places and arcs.

As these results affect the 'core' modelling techniques based on Petri nets, eliminating a source of an exponential explosion, we hope they will have applications in Petri net modelling and translations of various formalisms to Petri nets. As an example, applying them to translate Burst Automata to Petri nets reduces the size of the resulting Petri net from exponential down to polynomial.

Keywords: Petri net · Complete multipartite graph · Edge clique cover · Control flow · Burst Automata

1   Introduction

Petri nets (PN) are often used as a modelling formalism, and their advantages include the simplicity of semantics, the intuitive graphical notation, the abundance of software tools, and the possibility of capturing behaviours concisely without making subsequent processing (e.g., formal verification or synthesis) undecidable. In particular, the possibility to create concise models is often the
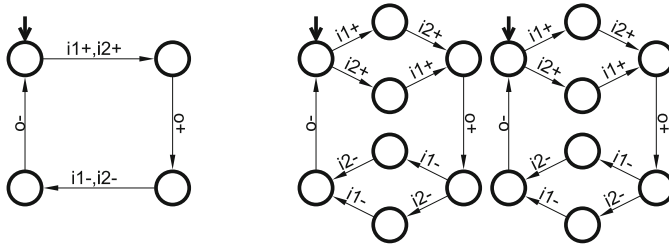
Fig. 1. A BA specification of the C-element and an FSM expressing its interleaving semantics. A C-element waits for both inputs to switch to 1 (actions i_1^+ and i_2^+) before switching its output to 1 (action o^+), and then waits for both inputs to switch to 0 (actions i_1^- and i_2^-) before switching its output to 0 (action o^-). It is assumed that the environment fulfills its part of the contract, i.e. each input switches only once before the output switches.
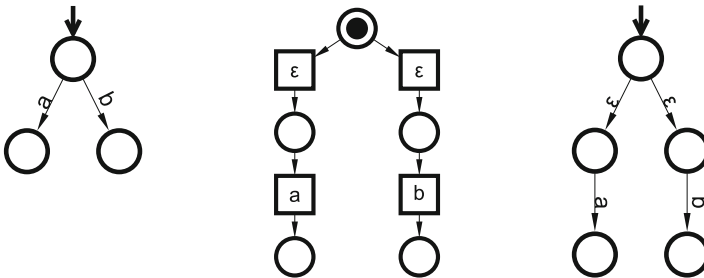

Fig. 2. A BA with singleton bursts, so coinciding with the FSM expressing its interleaving semantics (left); its PN translation prefixing each burst with a silent transition (middle); and the reachability graph (FSM) of this PN (right). Note that the two FSMs are language-equivalent but not weakly bisimilar.

key advantage of PNs over simpler formalisms like Finite State Machine (FSMs). Indeed, it is generally accepted that one is likely to encounter the exponential *state space explosion* [9] during, e.g., formal verification—this problem is believed to be fundamental (unless P=PSPACE), and mitigating this explosion using heuristics has been a hot research topic for many years. However, encountering an exponential explosion already during the modelling stage is both regrettable and indicative of problems in modelling techniques or even the formalism itself.

Unfortunately, some common PNs modelling techniques can indeed result in exponentially large models, even in common cases of simple control flows. As a motivating example we consider *Burst Automata* (BA) [3]—a more general version of Burst Mode [8] formalism from the area of asynchronous circuits design. Intuitively, BAs are similar to FSMs, except that their arcs are labelled not by single actions but by sets of actions ('bursts') which fire concurrently. One can define the interleaving semantics of BAs by allowing the actions in a burst to fire in any order, which results in the usual FSM, see the example in Fig. 1. For the purposes of formal verification and circuit synthesis, it would be advan-

tageous to develop a translation from BAs to PNs, in order to be able to use existing PN software. This means some kind of behavioural equivalence between the FSM expressing the interleaving semantics and the reachability graph of the resulting PN is required, e.g., language equivalence or bisimulation. As BAs are a very simple FSM-like formalism, it would be reasonable to expect that such a translation would be quite simple and efficient.

However, developing a compact translation from BAs to PNs is more complicated than one might expect. In particular, efficiently expressing a choice between several bursts of concurrent transitions is not trivial in PNs. In [3], a language-preserving linear size translation is proposed, that prefixes each burst with a silent 'fork' transition and then uses another silent 'join' transition after the burst to detect completion. Unfortunately, there are situations when this translation is unacceptable. First of all, silent transitions turn a deterministic model into a non-deterministic one which is often undesirable (e.g., non-determinism cannot be directly implemented physically, say in an asynchronous logic circuit [4]). Second, language equivalence may be too weak (e.g., it does not preserve branching time temporal properties or even deadlocks), and prefixing bursts with silent transitions breaks not only strong but also weak bisimulation, see Fig. 2.

To preserve strong bisimulation, the following *Cartesian Product Construction* ($\times$-construction) is traditionally used, see e.g., [2]. To express a choice between several bursts (i.e., sets of concurrent transitions) $B_1, B_2, \ldots, B_n$, this construction would create a set of places corresponding to tuples in the Cartesian product $B_1 \times B_2 \times \cdots \times B_n$, so that a place corresponding to a tuple $(b_1, \ldots, b_n)$ is connected to each transition b_i occurring in the tuple. This means that the number of created places is $|B_1| \cdot |B_2| \cdot \ldots \cdot |B_n|$, i.e., the PN size is exponential in the number of bursts.

In this paper we focus on efficiently expressing a choice between several bursts without using silent transitions. We demonstrate that the $\times$-construction is often sub-optimal, in particular one can always avoid an exponential explosion—a polynomial (quadratic) number of places is sufficient even in the worst case. Moreover, in the case of each burst containing two transitions, the $\times$-construction requires 2^n places while the construction proposed in this paper needs only $\log_2 n$ places (asymptotically)—a dramatic double-exponential reduction.

We believe that the proposed construction, which is the main contribution of the paper, will have many applications, as it affects the 'core' modelling techniques for PNs. In particular, translations from various formalisms to PNs relying on the $\times$-construction can be significantly improved by using the proposed construction instead, thus eliminating a source of exponential explosion. In Sect. 8, we show how to improve the bisimulation-preserving translation from BAs to PNs described in [3], from exponential down to polynomial. Note that the developed translation is just a simple example of applying the proposed construction rather than the focus of the paper, and there are more applications, e.g., in our forthcoming paper [7] we develop a polynomial PN translation for arbitrary control flows built from atomic actions using sequencing, parallel composition, and choice.

The proposed construction is based on the observation that the problem of implementing a choice between concurrent bursts in a PN using k places is

equivalent to finding an *edge clique cover* of a complete multipartite graph with k cliques. Hence, the minimal possible number of places is equivalent to the *edge clique cover number* (a.k.a. *intersection number* [5]) of a certain complete multipartite graph. The latter is a problem investigated for decades. Even though it is not completely solved, there are many useful published results, and we prove several new results helping to improve some upper bounds on the number of PN places and arcs. In practice, the optimality is usually not required, and one can use simple approximations with useful lower and upper bounds.

2   Basic Notions

In this section, we provide some basic notions related to Petri nets, complete multipartite graphs, and edge clique covers.

Petri Nets

We focus on *safe* (i.e., at most one token per place) PNs, which are often used for modelling control flows. For a safe PN, the total number of tokens in its initial marking cannot exceed the number of places, so we can define its size as the total number of places, transitions, and arcs, disregarding the initial marking. Note that the size of a PN is dominated by its arcs, except the uninteresting degraded case when there are many isolated nodes.

   In this paper, the set of transitions is usually given (e.g., when translating a model from some other formalism to PNs, the transitions often correspond to the occurrences of actions in that model), and the objective is to express the intended behaviour using small numbers of places and arcs. Note that having a small number of places is often desirable for formal verification as they correspond to state variables, and having a small number of arcs is desirable as they dominate the PN size.

Complete Multipartite Graphs

We consider undirected graphs with no parallel edges and no self-loops. A graph is called *multipartite* if its vertices are partitioned into several sets in such a way that there are no edges between vertices in the same partition. A multipartite graph is *complete* if, for every pair of vertices from different partitions, there is an edge connecting them. A complete multipartite graph with the partitions of sizes $t_1 \leq t_2 \leq \cdots \leq t_n$ will be denoted $K_{t_1,t_2,\ldots,t_n}$. Note that for the purposes of this paper one can assume that multipartite graphs have at least two partitions and that each partition contains at least two vertices—violating these assumptions leads to simple degraded cases. If the sizes of all partitions in $K_{t_1,t_2,\ldots,t_n}$ are equal, $t_1 = t_2 = \cdots = t_n = t$, the graph is called *balanced* and will be denoted $K_n(t)$.

Edge Clique Covers

A *clique* in a graph is a set of vertices which are pairwise connected by edges. A clique is called *maximal* if it is not a subset of any other clique. Note that for a complete multipartite graph, every maximal clique contains exactly one vertex

from each partition, and vice versa, by picking one vertex from each partition one always obtains a maximal clique.

A set of cliques in a graph form an *edge clique cover* (ECC) if for every edge in the graph there is at least one clique that contains both endpoints of this edge. The number of cliques in an ECC is called its *size*. Note that, given an ECC, one can expand each clique in it to some maximal one, without increasing the size of the ECC. The minimum possible size of an ECC of a graph G is called the *edge clique cover number* (a.k.a. *intersection number*) of G, and will be denoted $ecc(G)$.

3   Equivalence Between the Problems of Modelling a Choice Between Bursts in a PN and Finding an ECC of a Complete Multipartite Graph

Suppose that we have pairwise disjoint[1] bursts $B_1, B_2, \ldots, B_n$ where each burst is a non-empty set of transitions, and the intention is to create a choice between these bursts. Hence, the problem is to add some places and connect them to these transitions (by place→transition arcs) so that the transitions within each burst must be concurrent,[2] but any pair of transitions from different bursts must be in conflict. More precisely, the following requirements must be satisfied:

ReqChoice For every pair of transitions from different bursts, there is a place connected to both of them (this creates choices between transitions from different bursts).

ReqConc A place cannot be connected to more than one transition from the same burst (otherwise these transitions would not be concurrent). An alternative and stricter formulation of this requirement is that a place must be connected to exactly one transition from each burst—this in addition ensures that no tokens are left behind after a burst fires, which is essential for control flows containing cycles, a common case in practice. We will denote this strict formulation by **ReqConc**(strict). Note that if **ReqConc**(strict) is satisfied then the number of arcs can be determined from the number of places by multiplying the latter by the number of bursts, i.e., by n.

ReqNoDups No two places are connected to the same set of transitions (this would create redundancy – one of such places can be removed without affecting the PN's behaviour).

Furthermore, we define the problem size as $|B_1| + \cdots + |B_n|$—the upper and lower bounds on the number of required places and arcs will be relative to it.

---

[1] In practice, transitions are often labelled by actions, and there is no requirement for the sets of labels corresponding to bursts to be disjoint, e.g., it is possible to have bursts $B_1 = \{t_1\}$ and $B_2 = \{t_2\}$ where t_1 and t_2 are distinct transitions labelled by the same action a. In particular, BAs allow non-determinism, and the PN translation in Sect. 8 expresses it by creating distinct transitions with the same label.

[2] Note that the $|B_i|$-ary concurrency relation intended here is not reducible to pairwise concurrency.

One can see that the ×-construction satisfies the above conditions, including **ReqConc**(strict). However, it generates $|B_1| \cdot |B_2| \cdot \ldots \cdot |B_n|$ places and $n \cdot |B_1| \cdot |B_2| \cdot \ldots \cdot |B_n|$ arcs, i.e., the PN size is exponential.

The natural questions now are whether the above requirements are possible to achieve with fewer places and arcs—in particular, whether the PN size can be polynomial in the problem size, what would be the minimal number of places, and whether it is possible to derive some useful lower and upper bounds on the size of the smallest PN. We show that these problems can be reformulated in terms of finding ECCs of a complete multipartite graph, which provides revealing insights and helps one to find positive answers to these questions.

Consider the conflict (i.e., choice) relation between the transitions. It is symmetric and irreflexive, and so can be represented by an undirected graph without self-loops such that there is an edge between two vertices iff the transitions corresponding to these vertices belong to different bursts. Thus, the graph is a complete multipartite graph with its partitions corresponding to the bursts, i.e., $K_{|B_1|,|B_2|,\ldots,|B_n|}$. Furthermore, any complete multipartite graph represents the conflict relation for some family of bursts.

Now consider a place connected to several transitions picked from different bursts. The vertices corresponding to these transitions form a clique in the graph representing the conflict relation. Thus, given any ECC of this graph, one can create a place for each clique in the ECC and connect it to transitions corresponding to the vertices occurring in the clique, and the resulting set of places will satisfy the above requirements. Furthermore, one can in addition satisfy **ReqConc**(strict) by extending every clique in the ECC to a maximal one. Hence, there is a 1-to-1 correspondence between cliques and places, or between maximal cliques and places in the **ReqConc**(strict) case. This simple observation allows one to answer some of the posed questions:

- The *trivial* ECC where each edge is covered by a clique with two vertices (which can then be extended to a maximal clique if necessary) has only a quadratic number of cliques at most, which yields a polynomial translation with a quadratic number of places and either quadratic (for **ReqConc**) or cubic (for **ReqConc**(strict)) number of arcs—already a huge improvement on the exponential ×-construction.
- In bipartite graphs there are no cliques with more than two vertices, and so the minimal ECC coincides with the trivial one. Consider the case of two bursts of equal size, $|B_1| = |B_2| = k$. One can see that the graph of the conflict relation is the bipartite graph $K_{k,k}$, i.e., the number of PN places cannot be smaller than k^2. This observation yields a quadratic worst case lower bound on the number of places, that matches the above upper bound. Furthermore, this gives a quadratic worst case lower bound on the number of arcs, that matches the above upper bound for the **ReqConc** case, though there is still a gap between this lower bound and the upper bound for **ReqConc**(strict).

These simple observations, though giving matching worst case upper and lower bounds on the number of places as well as the number of arcs in the **ReqConc** case, do not provide the full picture. It turns out that one can often

do much better than the above quadratic worst case lower bound suggests, e.g., in the next section, we give an example where a logarithmic number of places is sufficient.

Furthermore, the **ReqConc**(strict) case is more important in practice, and given that the size of a PN is normally dominated not by places but by arcs, the gap between quadratic lower and cubic upper bounds on the number of arcs needs narrowing and ideally closing. In what follows, we derive some improved upper bounds in the **ReqConc**(strict) case, including a quadratic upper bound for the cases where the sizes of bursts are the same or at least not too different, i.e., the graph of the conflict relation is balanced or almost balanced.

4   A Logarithmic Case

Above we derived a polynomial (quadratic) worst-case bound on the number of places, which is an exponential improvement over the $\times$-construction. In this section, we consider the best-case scenario, excluding the degraded cases of a single burst, and the situation when some of the bursts are singletons. The former is trivial, and the latter can be reduced to a smaller non-degraded case by removing all singleton bursts. It turns out that in the best case a logarithmic number of places is both necessary and sufficient, yielding a dramatic double-exponential improvement on the $\times$-construction.

Consider the case of each burst containing two transitions. The $\times$-construction would create 2^n places, but it turns out that $\log_2 n$ places are sufficient (asymptotically). The conflict relation graph in this case is $K_n(2)$, and the problem boils down to finding a small ECC of this graph. This problem was solved in [6], which proved that $ecc(K_n(2)) \sim log_2 n$.

The idea of the construction is as follows. For simplicity, we assume that $n = \binom{k-1}{k/2}$ for some even k. One can check (e.g., using wolframalpha.com) that

$$\lim_{k \to +\infty} \frac{\log_2 n}{k} = \lim_{k \to +\infty} \frac{\log_2 \binom{k-1}{k/2}}{k} = 1 \,,$$

and so $k \sim log_2 n$. Consider the family of subsets of size $k/2$ of $\{1, \ldots, k\}$. One can build a multipartite graph with these subsets as vertices, and with two vertices being connected iff their intersection is non-empty (this construction is called the *intersection graph* [5] of a family of sets). In our case each subset has a non-empty intersection with all but one other subset in the family, viz. its complement, and so $K_n(2)$ is the intersection graph of this family of subsets. One can then observe that the vertices corresponding to subsets sharing some element i are pairwise connected and so form a clique. Moreover, the cliques corresponding to the elements of $\{1, \ldots, k\}$ form an ECC of size k.

For example, let $n = 10 = \binom{6-1}{6/2}$, i.e., $k = 6$ places are sufficient to express all the conflicts between 10 binary bursts (compared to $2^{10} = 1024$ places created by the $\times$-construction). We consider all 3-element subsets of the set $\{1, \ldots, 6\}$ and pair subsets with their complements:
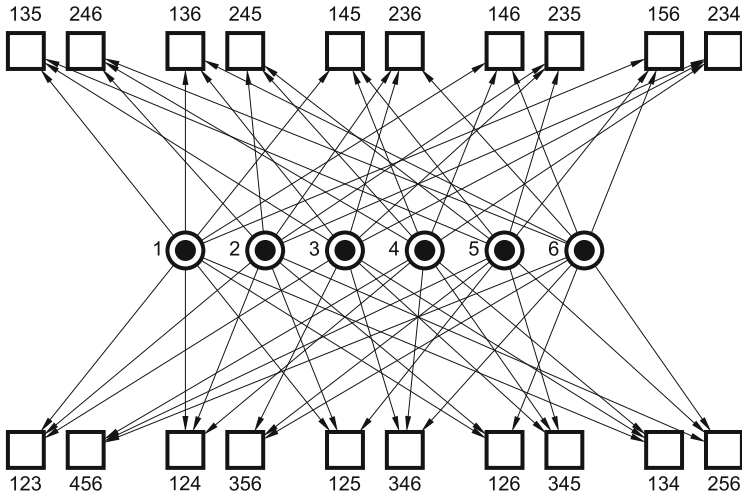
Fig. 3. A PN model with 6 places expressing a choice between 10 binary bursts. The transitions are labelled by 3-element subsets of $\{1, \ldots, 6\}$, e.g., '123' corresponds to $\{1, 2, 3\}$.

$$\{1, 2, 3\} \ \{4, 5, 6\}$$
$$\{1, 2, 4\} \ \{3, 5, 6\}$$
$$\{1, 2, 5\} \ \{3, 4, 6\}$$
$$\{1, 2, 6\} \ \{3, 4, 5\}$$
$$\{1, 3, 4\} \ \{2, 5, 6\}$$
$$\{1, 3, 5\} \ \{2, 4, 6\}$$
$$\{1, 3, 6\} \ \{2, 4, 5\}$$
$$\{1, 4, 5\} \ \{2, 3, 6\}$$
$$\{1, 4, 6\} \ \{2, 3, 5\}$$
$$\{1, 5, 6\} \ \{2, 3, 4\}$$

One can see that the intersection graph of this family of subsets is $K_{10}(2)$ where the vertices in each of the 10 partitions correspond to these 10 pairs of complimentary sets. Furthermore, one can cover all the edges of this graph by 6 cliques, where the i-th clique comprises vertices corresponding to the subsets containing i, for each $i = 1, \ldots, 6$.

In the corresponding PN, the pairs of transitions in these 10 bursts can be labelled by the above pairs of subsets, the places be labelled by the numbers $1, \ldots, 6$, and the connection be such that there is an arc from place i to a transition labelled by subset S iff $i \in S$, see Fig. 3.

5   Upper Bounds for the Balanced Case

In this section, we provide two different upper bounds on the number of places and arcs for the balanced case, i.e., when all the bursts have the same size. It

turns out one can improve the trivial upper bound on the number of arcs derived in Sect. 3.

The idea of the first bound is based on the observation that, by generating a sufficient number k of random cliques, one can cover every edge with high probability (in the sense that the expected number of uncovered edges is <1), which implies that there is an ECC of size k. Furthermore, the probability of an edge being uncovered falls exponentially with k, so k does not have to be large. The following result was inspired by [1, Lemma 3.2]. Using the specifics of balanced complete multipartite graphs, we reformulated that result to avoid references to graph's complement, streamlined the proof, and obtained a better multiplicative constant.

Proposition 1. $ecc(K_n(t)) \leq \left\lceil 2t^2 \ln \frac{tn}{\sqrt{2}} \right\rceil$ *if* $n, t \geq 2$.

Proof. One can pick a random maximal clique in $K_n(t)$ by randomly and uniformly picking a vertex from each of the n parts. Suppose that k (to be chosen appropriately below) such cliques are picked independently.

Given a random clique as above, an edge (u, v) is covered by it iff both u and v were picked from their partitions, i.e., the probability that a particular edge is covered by a random maximal clique is $1/t^2$. Hence the probability that an edge is not covered by any of the k chosen random cliques is

$$\left(1 - \frac{1}{t^2}\right)^k \leq e^{-k/t^2},$$

where the inequality follows from $1 - x \leq e^{-x}$ for all real x.

There are $\frac{t^2 n(n-1)}{2} < \frac{t^2 n^2}{2}$ edges in $K_n(t)$, so the expected number of edges not covered by any of the k cliques does not exceed

$$\frac{t^2 n^2}{2} e^{-k/t^2}.$$

We now choose $k = \left\lceil t^2 \ln \frac{t^2 n^2}{2} \right\rceil > t^2 \ln \frac{t^2 n^2}{2}$ (note that the inequality here is indeed strict because $n, t \geq 2$ are integers and the natural logarithm of an integer distinct from 1 is never an integer). Substituting this value for k into the above formula for the expected number of edges not covered by any of the k cliques, we have

$$\frac{t^2 n^2}{2} e^{-\frac{\left\lceil t^2 \ln \frac{t^2 n^2}{2} \right\rceil}{t^2}} < \frac{t^2 n^2}{2} e^{-\frac{t^2 \ln \frac{t^2 n^2}{2}}{t^2}} = 1,$$

i.e., it is strictly below 1. Hence, it is possible to choose

$$k = \left\lceil t^2 \ln \frac{t^2 n^2}{2} \right\rceil = \left\lceil 2t^2 \ln \frac{tn}{\sqrt{2}} \right\rceil$$

cliques forming an ECC of $K_n(t)$. □

One can see that the proof of the above proposition yields a randomised algorithm for generating a small ECC, and the derived upper bound is usually good in practice. However, it is expressed in terms of the number n of partitions in $K_n(t)$ and their (common) size t rather than the number $v = t \cdot n$ of vertices $K_n(t)$ (note that v is the problem size that was defined as the total size of all bursts). Reformulating this bound in terms of v yields

$$\left\lceil 2\frac{v^2}{n^2} \ln \frac{v}{\sqrt{2}} \right\rceil = O\left(\frac{v^2}{n^2} \log v\right),$$

which means that the number of places in the PN is $O\left(v^2 \log v\right)$ if n is fixed. This means that in some cases this upper bound can be worse than the trivial quadratic one derived in Sect. 3. However, the above expression allows one to improve on the trivial cubic upper bound on the number of arcs in the **Req-Conc**(strict) case derived in Sect. 3. Recall that in this case the number of arcs equals to the number of places multiplied by n, i.e.,

$$n\left\lceil 2\frac{v^2}{n^2} \ln \frac{v}{\sqrt{2}} \right\rceil = O\left(\frac{v^2}{n} \log v\right).$$

This bound is quite good, and becomes $O\left(v \log v\right)$ if n is linear in v. The worst case is when n is fixed—the bound then becomes $O\left(v^2 \log v\right)$, which almost (but not quite) matches the quadratic worst case lower bound.

The natural question now is whether one can achieve quadratic (in v) upper bounds on the numbers of both places and arcs, thus completely closing the gap between the worst case lower and upper bounds at least in the balanced case. The following simple result provides a new (to our knowledge) upper bound on the edge clique cover number of $K_n(t)$ that allows one to answer this question affirmatively.

Proposition 2. $ecc(K_n(t)) \leq nt^2/2$, where $t, n \geq 2$.

Proof. We construct an ECC comprised of the cliques:

- C_{pij}, where $p = 1, \ldots, n$, $i = 2, \ldots, t$ and $j = 1, \ldots, i-1$, such that C_{pij} comprises the i-th vertex from p-th partition and the j-th vertex from every other partition.
- C_i, where $i = 1, \ldots, t$, containing the i-th vertex from every partition.

Note that the edge from the i-th vertex from p-th partition to j-th vertex in some other partition p' is covered by either C_{pij} (if $i > j$) or $C_{p'ji}$ (if $j > i$) or C_i (if $i = j$), so it is indeed an ECC. The size of this ECC is

$$nt(t-1)/2 + t = nt^2/2 - nt/2 + t \leq nt^2/2,$$

and so $ecc(K_n(t)) \leq nt^2/2$. □

The above result can be used to derive quadratic upper bounds on the numbers of both places and arcs. Indeed, when expressed in terms of v, this bound becomes $\frac{v^2}{2n}$ and so cannot be worse than $O(v^2)$ even if n is fixed. Furthermore, the number of arcs in **ReqConc**(strict) case can be obtained by multiplying by n, which yields $\frac{v^2}{2}$.

Hence, in the balanced case we have obtained quadratic worst case upper bounds on both places and arcs, which match the quadratic worst case lower bound derived in Sect. 3.

6   Upper Bounds for the Almost Balanced Case

In this section, we show that the upper bounds from the previous sections can be transferred (up to a multiplication by a constant) to the unbalanced case provided it is not 'too unbalanced'. The idea is based on the following observations:

- $K_{t_1, t_2, \ldots, t_n}$ is an induced sub-graph in $K_n(t_n)$, and so $ecc(K_{t_1, t_2, \ldots, t_n}) \leq ecc(K_n(t_n))$, as any ECC of the latter can be turned into an ECC of the former by deleting the vertices which are not in $K_{t_1, t_2, \ldots, t_n}$ from each clique (and then extending each clique to a maximal one if required).
- The number of vertices in $K_n(t_n)$ is not much greater than that in $K_{t_1, t_2, \ldots, t_n}$ provided that the latter is not 'too unbalanced', so the upper bounds derived in the previous section do not become too large w.r.t. the size of $K_{t_1, t_2, \ldots, t_n}$.

Formally, let $b \geq 1$ be some fixed real number. A complete multipartite graph $K_{t_1, t_2, \ldots, t_n}$ is called *b-balanced* if $t_n \leq \frac{b}{n} \sum_{i=1}^{n} t_i = \frac{b}{n} v$, i.e., the size of the biggest partition is within the factor b of the average partition size. One can observe that the number of vertices in $K_n(t_n)$ is then

$$nt_n \leq n \left(\frac{b}{n} v \right) = bv,$$

i.e., within the factor b of the number v of vertices in $K_{t_1, t_2, \ldots, t_n}$. Hence the bounds derived in Sect. 5 can be lifted to 'almost balanced' graphs as follows.

Proposition 3 (Lifting Propositions 1 and 2 to almost balanced case). *Suppose that $K_{t_1, t_2, \ldots, t_n}$ is b-balanced and $n, t_i \geq 2$. Then*

$$ecc(K_{t_1, t_2, \ldots, t_n}) \leq \left\lceil 2 \frac{(bv)^2}{n^2} \ln \frac{bv}{\sqrt{2}} \right\rceil$$

and

$$ecc(K_{t_1, t_2, \ldots, t_n}) \leq \frac{(bv)^2}{2n} \, ,$$

where $v = \sum_{i=1}^{n} t_i$ is the number of vertices in $K_{t_1, t_2, \ldots, t_n}$.

Proof. Follows directly from the $\left\lceil 2 \frac{v^2}{n^2} \ln \frac{v}{\sqrt{2}} \right\rceil$ and $\frac{v^2}{2n}$ bounds for the balanced case derived in Sect. 5. □

Hence, in the almost balanced case the numbers of places and arcs are quadratic in the worst case. Furthermore, Proposition 5 below improves the former bound down to $\left\lceil 2t_n t_{n-1} \ln \frac{v}{\sqrt{2}} \right\rceil$. Note that, for a b-balanced graph, $t_{n-1} \leq t_n \leq \frac{bv}{n}$ and, moreover, b is removed from under the logarithm.

7 Upper Bounds for the Unbalanced Case

In this section, we consider the case of complete multipartite graphs $K_{t_1,t_2,\ldots,t_n}$ which are 'very unbalanced'. We slightly improve on the trivial upper bound on the number of arcs. However, this may still be cubic in the worst case, e.g., if n, t_n and t_{n-1} are linear in v.

First, we recall the following result from [5]. It refers to *edge clique partitions*, which are a special case of ECCs with every edge covered exactly by one clique. We denote the *edge clique partition number* $ecp(G)$ of a graph G as the smallest possible number of cliques in an edge clique partition of G. Trivially, $ecc(G) \leq ecp(G)$.

Proposition 4 (adapted from [5]). *Let G be a graph with v vertices. Then G has an edge clique partition of size at most $\lfloor v^2/4 \rfloor$ consisting of edges and triangles. Moreover, $ecp(G) = \lfloor v^2/4 \rfloor$ if and only if G is $K_{\lfloor v/2 \rfloor, \lceil v/2 \rceil}$.*

This translates into a $\lfloor v^2/4 \rfloor$ upper bound on the number of places and $n \lfloor v^2/4 \rfloor$ upper bound on the number of arcs in **ReqConc**(strict) case, which may be $O(v^3)$ if n (the number of bursts) grows linearly in v.

It should be noted that this bound is rather pessimistic: Edge clique partition is a very special case of ECC, and moreover only edges and triangles are used in this partition. Hence one can hope that the bound on arcs could be improved:

– The worst case for Proposition 4 is a balanced bipartite graph, in which case $n = 2$ and so the number of arcs is quadratic (and it remains quadratic for any fixed n).
– When n grows linearly, much larger cliques than edges and triangles can be formed.

As a result, we hope that a sub-cubic or even a quadratic bound on the number of arcs can be obtained, and leave this question for future research.

We now generalise Proposition 1 to the unbalanced case, which gives a slightly better bound than that derived in Sect. 6.

Proposition 5. $ecc(K_{t_1,t_2,\ldots,t_n}) \leq \left\lceil 2t_n t_{n-1} \ln \frac{v}{\sqrt{2}} \right\rceil$ *where v is the number of vertices in $K_{t_1,t_2,\ldots,t_n}$ and $n, t_i \geq 2$.*

Proof. One can pick a random maximal clique in $K_{t_1,t_2,\ldots,t_n}$ by randomly and uniformly picking a vertex from each of the n partitions. Suppose that k (to be chosen appropriately below) such cliques are picked independently.

Given a random clique as above, an edge (u, v) is covered by it iff both u and v were picked from their partitions, i.e., the probability that an edge (u, v) is covered by a random maximal clique is $1/(t_u t_v)$, where t_u and t_v are the sizes of partitions u and v are coming from. Hence the probability that an edge (u, v) is not covered by any of the k chosen random cliques is

$$\left(1 - \frac{1}{t_u t_v}\right)^k \le e^{-k/(t_u t_v)} \le e^{-k/t_n t_{n-1}},$$

where the former inequality follows from $1 - x \le e^{-x}$ for all real x.

There are $\frac{1}{2}\left(v^2 - \sum_{i=1}^{s} t_i^2\right) < \frac{v^2}{2}$ edges in the graph, so the expected number of edges not covered by any of the k cliques does not exceed

$$\frac{v^2}{2} e^{-k/t_n t_{n-1}}.$$

We now choose $k = \left\lceil t_n t_{n-1} \ln \frac{v^2}{2} \right\rceil > t_n t_{n-1} \ln \frac{v^2}{2}$ (note that the inequality here is indeed strict because $v \ge 4$ and $t_i \ge 2$ are integers and the natural logarithm of an integer distinct from 1 is never an integer). Substituting this value for k into the above formula for the expected number of edges not covered by any of the k cliques, we have

$$\frac{v^2}{2} e^{-\frac{\left\lceil t_n t_{n-1} \ln \frac{v^2}{2} \right\rceil}{t_n t_{n-1}}} < \frac{v^2}{2} e^{-\frac{t_n t_{n-1} \ln \frac{v^2}{2}}{t_n t_{n-1}}} = 1 \,,$$

i.e., it is strictly below 1. Hence, it is possible to choose

$$k = \left\lceil t_n t_{n-1} \ln \frac{v^2}{2} \right\rceil = \left\lceil 2 t_n t_{n-1} \ln \frac{v}{\sqrt{2}} \right\rceil$$

cliques forming an ECC of $K_{t_1, t_2, \dots, t_n}$.   □

8   A Polynomial Bisimulation-Preserving Translation from BAs to PNs

In [3], three translations from BAs to PNs were developed. One of them is linear—but it uses silent 'fork' and 'join' transitions for each burst and so preserves only language equivalence but not bisimulation (not even weak bisimulation)—as illustrated in Fig. 2. The second translation uses 'join' (but not 'fork') transitions and the ×-construction. As a result, it preserves weak bisimulation but is exponential. The third translation uses neither 'fork' nor 'join' transitions, and preserves strong bisimulation, but it also depends on the ×-construction and thus is exponential (and often larger than the second translation).

We now show how to eliminate a source of exponential explosion in the latter construction by replacing the ×-construction by the ECC-based one, thus

obtaining a polynomial bisimulation-preserving translation from BAs to PNs. Note that the developed translation is just a simple example of applying the proposed ECC-based construction rather than the focus of the paper, and there are more applications, e.g., in our forthcoming paper [7] we develop a polynomial PN translation for arbitrary control flows built from atomic actions using sequencing, parallel composition, and choice.

The improved translation is illustrated in Fig. 4 and works as follows.

– Each burst B is represented by $|B|$ transitions corresponding to occurrences of actions in B and labelled by the corresponding actions. (Empty bursts are interpreted as ε-transitions in FSMs, and so for the purposes of this translation are replaced by singleton bursts $\{\varepsilon\}$.) No other transitions are created by the translation. Then each state s of the BA is considered in turn, together with its incoming and outgoing bursts.
– For the incoming bursts of s, a set P_s^{in} of new places is created, so that $|P_s^{in}|$ is the maximal input burst cardinality (and hence of linear size). The transitions in the input bursts are then connected by transition→place arcs to the places in P_s^{in}, so that the i-th transition in each burst is connected to the i-th place in P_s^{in}. Moreover, for bursts with fewer than $|P_s^{in}|$ transitions, extra arcs are added so that each place in P_s^{in} is connected to exactly one transition in the burst (e.g., one can connect the first transition in the burst to all the unmatched places in P_s^{in}).
– For the outgoing bursts of s, the **ReqConc**(strict) variant of the ECC-based construction presented in this paper is applied, yielding a set P_s^{out} of places. Note that $|P_s^{out}|$ is at most quadratic, and the number of created arcs is at most cubic (in the total size of all output bursts).
– To enforce the causality between the input and output bursts, a set of places $P_s^{(in,out)} = P_s^{in} \times P_s^{out}$ is created,[3] where each place $(p_{in}, p_{out}) \in P_s^{(in,out)}$ inherits its incoming arcs from p_{in} and its outgoing arcs from p_{out}. After that, the places in P_s^{in} and P_s^{out} are removed from the PN, together with their arcs.
– Finally, if s is the initial state of the BA, all the places in $P_s^{(in,out)}$ are initially marked.

Clearly, the resulting PN is strongly bisimilar with the original BA (in fact, its reachability graph is isomorphic to the FSM expressing the interleaving semantics of BA, which is an even stronger equivalence), and its size is polynomial in the size of BA, improving thus the exponential translation of [3].

It should be noted that self-loops with non-singleton bursts in BAs may cause the resulting PN to be unsafe (2-bounded). If this is undesirable, the problem can be easily avoided by replicating BA states with self-loops before the translation, as follows. If s is a state with a self-loop, a new state s' is created that inherits all

---

[3] This aspect of the translation can be improved as explained in our forthcoming paper [7]. In fact, the number of places in the Petri net in Fig. 4(bottom) can be reduced from 12 down to 6. However, the given simple construction is already sufficient for the translation to be polynomial.
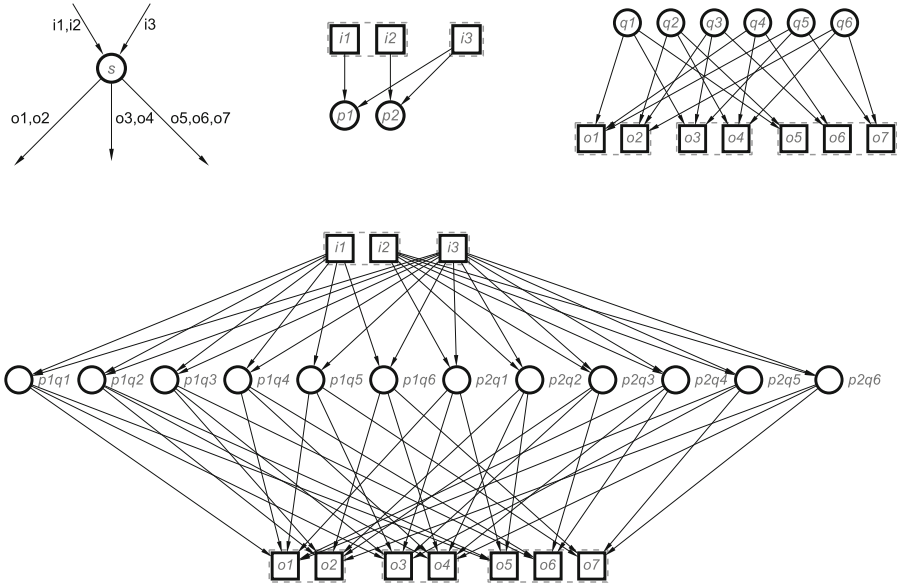
Fig. 4. An example of bisimulation-preserving BA to PN translation: (top-left) A BA state with its incoming and outgoing bursts; (top-middle) PN translations of the incoming bursts—the maximal incoming burst size is two, so two places are created; (top-right) PN translations of the outgoing bursts—$ecc(K_{2,2,3}) = 6$ places are created; (bottom) combined PN—the places corresponds to pairs in $\{p_1, p_2\} \times \{q_1, \ldots, q_6\}$.

the incoming and outgoing arcs of s, except the self-loops. Then each self-loop at s labelled by a burst B is replaced by two B-labelled arcs, $s \to s'$ and $s' \to s$. This transformation at most doubles the size of the BA.

9   Conclusions

In this paper, we observed that the $\times$-construction often used for the modelling of a choice between concurrent bursts is sub-optimal and causes an exponential explosion in the size of PNs that can be avoided by better modelling. We showed equivalence between this modelling problem, and the problem of finding an ECC of a complete multipartite graph. It provided helpful insights into the former problem as well as linking it to existing results from graph theory. This enabled us:

– To show that the exponential number of places created by the $\times$-construction can be improved down to polynomial (quadratic) even in the worst case, and down to logarithmic in the best (non-degraded) case.
– To derive quadratic worst case lower and upper bounds on the number of places.

– To derive quadratic worst case lower and upper bounds on the number of arcs in **ReqConc** case.
– To derive quadratic worst case lower and upper bounds on the number of arcs in **ReqConc**(strict) case for the balanced and 'almost balanced' cases.
– To derive a quadratic worst case lower bound and a cubic worst case upper bound on the number of arcs in **ReqConc**(strict) case, as well as several upper bounds which in some situation can be better than cubic. There is still a gap between the quadratic lower and cubic upper bounds, closing which we leave for future research.
– To obtain a polynomial bisimulation-preserving translation from BAs to PNs, as an example of applying the proposed construction to improve the exponential translation in [3].

These results eliminate a source of exponential explosion in PNs when modelling control flows and in translations from various formalisms to PNs. We believe that these results will have wide applications, as they affect the 'core' modelling techniques based on PNs.

In our future work, besides improving the upper bounds as explained above, we plan to lift the proposed ECC-based construction to the case of more general control flows, and in our forthcoming paper [7] we present a polynomial PN translation for arbitrary control flows built from atomic actions using sequencing, parallel composition, and choice.

References

1. Alon, N.: Covering graphs by the minimum number of equivalence relations. Combinatorica **6**(3), 201–206 (1986)
2. Best, E., Devillers, R.R., Koutny, M.: Petri Net Algebra. Monographs in Theoretical Computer Science. An EATCS Series. Springer, Heidelberg (2001). https://doi.org/10.1007/978-3-662-04457-5
3. Chan, A., Sokolov, D., Khomenko, V., Lloyd, D., Yakovlev, A.: Burst automaton: framework for speed-independent synthesis using burst-mode specifications (2021, submitted paper)
4. Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L., Yakovlev, A.: Logic Synthesis for Asynchronous Controllers and Interfaces. Springer, Heidelberg (2002). https://doi.org/10.1007/978-3-642-55989-1
5. Erdös, P., Goodman, A.W., Pósa, L.: The representation of a graph by set intersections. Can. J. Math. **18**, 106–112 (1966)
6. Gregory, D.A., Pullman, N.J.: On a clique covering problem of Orlin. Discrete Math. **41**(1), 97–99 (1982)
7. Khomenko, V., Koutny, M., Yakovlev, A.: Slimming down Petri boxes: Compact Petri net models of control flows (2022, in preparation)

8. Nowick, S.M., Dill, D.L.: Synthesis of asynchronous state machines using a local clock. In: Proceedings 1991 IEEE International Conference on Computer Design: VLSI in Computer & Processors, ICCD 1991, Cambridge, MA, USA, 14–16 October 1991, pp. 192–197. IEEE Computer Society (1991)
9. Valmari, A.: The state explosion problem. In: Reisig, W., Rozenberg, G. (eds.) ACPN 1996. LNCS, vol. 1491, pp. 429–528. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-65306-6_21