Logical Methods in Computer Science
Volume 19, Issue 4, 2023, pp. 2:1–2:44
https://lmcs.episciences.org/

Submitted Aug. 12, 2021
Published Oct. 12, 2023

FINDING CUT-OFFS IN LEADERLESS RENDEZ-VOUS PROTOCOLS IS EASY

A. R. BALASUBRAMANIAN [a], JAVIER ESPARZA [a], AND MIKHAIL RASKIN [b]

[a] Technische Universität München, Munich, Germany
e-mail address: bala.ayikudi@tum.de, esparza@in.tum.de

[b] LaBRI, University of Bordeaux, Talence, France
e-mail address: mikhail.raskin@u-bordeaux.fr

Abstract. In rendez-vous protocols an arbitrarily large number of indistinguishable finite-state agents interact in pairs. The cut-off problem asks if there exists a number B such that all initial configurations of the protocol with at least B agents in a given initial state can reach a final configuration with all agents in a given final state. In a recent paper [HS20], Horn and Sangnier proved that the cut-off problem is decidable (and at least as hard as the Petri net reachability problem) for protocols with a leader, and in EXPSPACE for leaderless protocols. Further, for the special class of symmetric protocols they reduce these bounds to PSPACE and NP, respectively. The problem of lowering these upper bounds or finding matching lower bounds was left open. We show that the cut-off problem is P-complete for leaderless protocols and in NC for leaderless symmetric protocols. Further, we also consider a variant of the cut-off problem suggested in [HS20], which we call the bounded-loss cut-off problem and prove that this problem is P-complete for leaderless protocols and NL-complete for leaderless symmetric protocols. Finally, by reusing some of the techniques applied for the analysis of leaderless protocols, we show that the cut-off problem for symmetric protocols with a leader is NP-complete, thereby improving upon all the elementary upper bounds of [HS20].

1. Introduction

Distributed systems are often designed for an unbounded number of participating agents. Therefore, they are not just one system, but an infinite family of systems, one for each number of agents. Parameterized verification addresses the problem of checking that all systems in the family satisfy a given specification.

In many application areas, agents are indistinguishable. This is the case in computational biology, where cells or molecules have no identities; in some security applications, where the agents' identities should stay private; or in applications where the identities can be abstracted away, like certain classes of multithreaded programs [GS92, AAD$^+$06, SCWB08,

Key words and phrases: rendez-vous protocols, cut-off problem, Petri nets.
This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 787367 (PaVeS).
The last author was affiliated with Technische Universität München when this work was done.

LOGICAL METHODS IN COMPUTER SCIENCE
DOI:10.46298/LMCS-19(4:2)2023

BMWK09, KKW10, NB15]. Following [BMWK09, KKW10], we use the term *replicated systems* for distributed systems with indistinguishable agents. Replicated systems include population protocols, broadcast protocols, threshold automata, and many other models [GS92, AAD+06, EFM99, DSTZ12, GKS+14]. They also arise after applying a *counter abstraction* [PXZ02, BMWK09]. In finite-state replicated systems the global state of the system is determined by the function (usually called a *configuration*) that assigns to each state the number of agents that currently occupy it. This feature makes many verification problems decidable [BJK+15, Esp16].

Surprisingly, there is no a priori relation between the complexity of a parameterized verification question (i.e., whether a given property holds for all initial configurations, or, equivalently, whether its negation holds for some configuration), and the complexity of its corresponding single-instance question (whether the property holds for a fixed initial configuration). Consider replicated systems where agents interact in pairs [GS92, HS20, AAD+06]. The complexity of single-instance questions is very robust. Indeed, checking most properties, including all properties expressible in LTL and CTL, is PSPACE-complete [Esp96]. On the contrary, the complexity of parameterized questions is very fragile, as exemplified by the following example. While the existence of a reachable configuration that populates a given state with *at least* one agent is in P, and so well below PSPACE, the existence of a reachable configuration that populates a given state with *exactly* one agent is as hard as the reachability problem for Petri nets, and so non-primitive recursive [Ler21, Las22, CO21]. This fragility makes the analysis of parameterized questions very interesting, but also much harder.

Work on parameterized verification has concentrated on whether every initial configuration satisfies a given property (see e.g. [GS92, EFM99, BMWK09, KKW10, DSTZ12]). However, applications often lead to questions of the form "do all initial configurations *in a given set* satisfy the property?", "do infinitely many initial configurations satisfy the property?", or "do all but finitely many initial configurations satisfy the property?". An example of the first kind is proving correctness of population protocols, where the specification requires that for a given partition $\mathcal{I}_0$, $\mathcal{I}_1$ of the set of initial configurations, and a partition Q_0, Q_1 of the set of states, runs starting from $\mathcal{I}_0$ eventually trap all agents within Q_0, and similarly for $\mathcal{I}_1$ and Q_1 [EGLM17]. An example of the third kind is the existence of *cut-offs*; cut-off properties state the existence of an initial configuration such that for all larger initial configurations some given property holds [EK02, BJK+15]. A systematic study of the complexity of these questions is still out of reach, but first results are appearing. In particular, Horn and Sangnier have recently studied the complexity of the *cut-off problem* for parameterized rendez-vous networks [HS20]. The problem takes as input a network with one single initial state *init* and one single final state *fin*, and asks whether there exists a cut-off B such that for every number of agents $n \geq B$, the final configuration in which all agents are in state *fin* is reachable from the initial configuration in which all agents are in state *init*.

Horn and Sangnier study two versions of the cut-off problem, for leaderless networks and networks with a leader. Intuitively, a leader is a distinguished agent with its own set of states. They show that in the presence of a leader the cut-off problem is decidable and at least as hard as reachability for Petri nets, which shows that the cut-off problem is Ackermann-hard and therefore not primitive recursive [Ler21, Las22, CO21]. For the leaderless case, they show that the problem is in EXPSPACE. Further, they also consider the special case of

Cut-off - [HS20]	Asymmetric rendez-vous	Symmetric rendez-vous
Presence of a leader	Decidable and Ackermann-hard	PSPACE
Absence of a leader	EXPSPACE	NP

Cut-off - This paper	Asymmetric rendez-vous	Symmetric rendez-vous
Presence of a leader		NP-complete
Absence of a leader	P-complete	NC

Table 1: Summary of the results for the cut-off problem by [HS20] and this paper.

Bounded-loss - This paper	Asymmetric rendez-vous	Symmetric rendez-vous
Absence of a leader	P-complete	NL-complete

Table 2: Summary of the results for the bounded-loss cut-off problem

symmetric networks, for which they obtain better upper bounds: PSPACE for the case of a leader, and NP in the leaderless case. These results are summarized at the top of Table 1.

In [HS20] the question of improving the upper bounds or finding matching lower bounds is left open. In this paper we close it with a surprising answer: All elementary upper bounds of [HS20] can be dramatically improved. In particular, our main result shows that the EXPSPACE bound for the leaderless case can be brought down to P. Further, the PSPACE and NP bounds of the symmetric case can be lowered to NP and NC, respectively, as shown at the bottom of Table 1. We also obtain matching lower bounds. Finally, we provide almost tight upper bounds for the size of the cut-off B; more precisely, we show that if B exists, then $B \in 2^{n^{O(1)}}$ for a protocol of size n.

Our results follow from two lemmas, called the Scaling and Insertion Lemmas, that connect the *continuous semantics* for Petri nets to their standard semantics. In the continuous semantics of Petri nets transition firings can be scaled by a positive rational factor; for example, a transition can fire with factor 1/3, taking "1/3 of a token" from its input places. The continuous semantics is a relaxation of the standard one, and its associated reachability problem is much simpler (polynomial [FH15, Blo20] instead of non-primitive recursive). The Scaling Lemma states that given two markings M, M' of a Petri net, if M' is reachable from M in the continuous semantics, then nM' is reachable from nM in the standard semantics for some $n \in 2^{m^{O(1)}}$, where m is the total size of the net and the markings. This lemma is implicitly proved in [FH15], but the bound on the size of n is hidden in the details of the proof, and we make it explicit here. The Insertion Lemma states that, given four markings M, M', L, L', if M' is reachable from M in the continuous semantics and the *marking equation* $L' = L + \mathcal{A}\mathbf{x}$ has a solution $\mathbf{x} \in \mathbb{Z}^T$ (observe that $\mathbf{x}$ can have negative components), then $nM' + L'$ is reachable from $nM + L$ in the standard semantics for some $n \in 2^{m^{O(1)}}$. We think that these lemmas can be of independent interest.

Further we also consider the following question which was proposed in [HS20] as a variant of the cut-off problem: Given a network with initial state *init* and final state *fin*, decide if there is a bound B such that for any number of agents, the initial configuration in which all agents are in the state *init* can reach a configuration in which at most B agents *are not* in the final state *fin*. We call this the *bounded-loss cut-off problem*. Intuitively, in the cut-off problem, we ask if for any sufficiently large population size, all agents can be transferred from the state *init* to the state *fin*. In the bounded-loss cut-off problem, we ask if it is always possible to "leave out" a bounded number of agents, and transfer everybody else from the state *init* to the state *fin*. By adapting the techniques developed for the cut-off problem, we prove that the bounded-loss cut-off problem is P-complete for leaderless networks and NL-complete for symmetric leaderless networks.

This paper is an extended version of the conference paper [BER21] published at FoSSaCS 2021. Compared to the conference version, this paper contains full proofs of our results. Moreover all the results pertaining to the bounded-loss cut-off problem are new.

The paper is organized as follows. Section 2 contains preliminaries; in particular, it defines the cut-off problem for rendez-vous networks and reduces it to the cut-off problem for Petri nets. Section 3 gives a polynomial time algorithm for the leaderless cut-off problem for *acyclic* Petri nets. Section 4 introduces the Scaling and Insertion Lemmas, and Section 5 presents the novel polynomial time algorithm for the cut-off problem for *general* Petri nets. Section 6 presents the polynomial time algorithm for the bounded-loss cut-off problem for rendez-vous protocols. Sections 7 and 8 present the results for symmetric networks, for the cases without and with a leader, respectively.

2. Preliminaries

Multisets. Let E be a finite set. For a semi-ring $(S, +, \cdot)$, a vector from E to S is a function $v : E \to S$. The set of all vectors from E to S will be denoted by S^E. Given a vector $v \in S^E$ and an element $\alpha \in S$, we let $\alpha \cdot v$ be the vector given by $(\alpha \cdot v)(e) = \alpha \cdot v(e)$ for all $e \in E$. For the sake of brevity, whenever there is no confusion, we sometimes abbreviate $\alpha \cdot v$ as αv. In this paper, the semi-rings we will be concerned with are the natural numbers $(\mathbb{N}, +, \cdot)$, the integers $(\mathbb{Z}, +, \cdot)$ and the non-negative rationals $(\mathbb{Q}_{\geq 0}, +, \cdot)$. The *support* of a vector v is the set $\llbracket v \rrbracket := \{e : M(e) \neq 0\}$ and its *size* is the number $\|v\| = \sum_{e \in \llbracket M \rrbracket} abs(M(e))$ where $abs(x)$ denotes the absolute value of x. Vectors from E to $\mathbb{N}$ are also called discrete multisets (or just multisets) and vectors from E to $\mathbb{Q}_{\geq 0}$ are called continuous multisets.

Given two vectors v, v' (over either $\mathbb{N}$, $\mathbb{Z}$ or $\mathbb{Q}_{\geq 0}$) we say that $v \leq v'$ if $v(e) \leq v'(e)$ for all $e \in E$ and we let $v + v'$ be the vector given by $(v + v')(e) = v(e) + v'(e)$ for all $e \in E$. Further, if v and v' are vectors over S where $S \in \{\mathbb{Z}, \mathbb{Q}_{\geq 0}\}$, then we define $v - v'$ as the vector given by $(v - v')(e) = v(e) - v'(e)$ for all e. On the other hand, if v and v' are multisets (i.e., vectors over $\mathbb{N}$) *such that* $v' \leq v$, then we define $(v - v')(e) = v(e) - v'(e)$ for all e.

The vector which maps every element of E to 0 (resp. 1) is denoted by $\mathbf{0}$ (resp. $\mathbf{1}$). We sometimes denote multisets and continuous multisets using a set-like notation, e.g. $\wr a, 2 \cdot b, c \int$ denotes the multiset given by $M(a) = 1, M(b) = 2, M(c) = 1$ and $M(e) = 0$ for all $e \notin \{a, b, c\}$.

Given an $I \times J$ matrix A with I and J sets of indices, $I' \subseteq I$ and $J' \subseteq J$, we let $A_{I' \times J'}$ denote the restriction of A to rows indexed by I' and columns indexed by J'.

2.1. Rendez-vous Protocols and the Cut-off Problem.

Let Σ be a fixed finite set which we will call the communication alphabet and we let $RV(\Sigma) = \{!a, ?a : a \in \Sigma\}$. The symbol $!a$ denotes that the message a is sent and $?a$ denotes that the message a is received.

Definition 2.1. A *rendez-vous protocol* $\mathcal{P}$ is a tuple $(Q, \Sigma, init, fin, R)$ where Q is a finite set of *states*, Σ is the *communication alphabet* consisting of a finite set of *messages*, $init, fin \in Q$ are the *initial* and *final* states respectively and $R \subseteq Q \times RV(\Sigma) \times Q$ is the set of *rules*.

The size $|\mathcal{P}|$ of a protocol is defined as the number of bits needed to encode $\mathcal{P}$ using some standard encoding. A configuration C of $\mathcal{P}$ is a multiset of states, where $C(q)$ should be interpreted as the number of agents in state q. We use $\mathcal{C}(\mathcal{P})$ to denote the set of all configurations of $\mathcal{P}$. An initial (resp. final) configuration C is a configuration such that $C(q) = 0$ if $q \neq \mathit{init}$ (resp. $C(q) = 0$ if $q \neq \mathit{fin}$). We use C^n_{init} (resp. C^n_{fin}) to denote the initial (resp. final) configuration such that $C^n_{\mathit{init}}(\mathit{init}) = n$ (resp. $C^n_{\mathit{fin}}(\mathit{fin}) = n$).

The operational semantics of a rendez-vous protocol $\mathcal{P}$ is given by means of a transition system between the configurations of $\mathcal{P}$. Suppose a is a message in Σ and $r = (p, !a, p')$ and $r' = (q, ?a, q')$ are two rules of R. For any two configurations C and C', we say that $C \xRightarrow{r,r'} C'$ if $C \geq \wr p, q \int$ and $C' = C - \wr p, q \int + \wr p', q' \int$. Intuitively, the configuration C has agents at states p and q, and the agent at state p sends the message a and moves to p' and the agent at state q receives this message and moves to q'. We let $C \Rightarrow C'$ denote that there exist rules r, r' for which $C \xRightarrow{r,r'} C'$ and if this is the case, then we say that there is a transition from C to C'. As usual, $\xRightarrow{*}$ denotes the reflexive and transitive closure of $\Rightarrow$. The *cut-off problem for rendez-vous protocols*, as stated in [HS20], is then defined as the following decision problem.

Given: A rendez-vous protocol $\mathcal{P}$
Decide: Does there exist $B \in \mathbb{N}$ such that $C^n_{\mathit{init}} \xRightarrow{*} C^n_{\mathit{fin}}$ for every $n \geq B$?

If such a B exists then we say that $\mathcal{P}$ admits a cut-off and that B is a cut-off for $\mathcal{P}$.

Example 2.2. Let us consider the following protocol, which is taken from a slightly modified version of the family of protocols described in Figure 5 of [HS20].

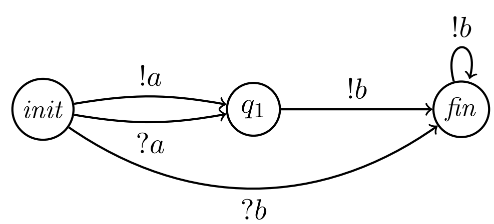

Figure 1: An example of a rendez-vous protocol

We can show that 4 is a cut-off for this protocol. Indeed, if $n \geq 4$, then we have the run $C^n_{\mathit{init}} \Rightarrow \wr (n-2) \cdot \mathit{init} + 2 \cdot q_1 \int \Rightarrow \wr (n-3) \cdot \mathit{init} + q_1 + 2 \cdot \mathit{fin} \int \Rightarrow \wr (n-4) \cdot \mathit{init} + 4 \cdot \mathit{fin} \int$. The first transition involves sending and receiving the message a from the state *init* and the other two involve sending the message b from q_1 and receiving it from *init*. Once we reach $\wr (n-4) \cdot \mathit{init} + 4 \cdot \mathit{fin} \int$, we can reach C^n_{fin} by repeatedly using the rules $(\mathit{fin}, !b, \mathit{fin})$ and $(\mathit{init}, ?b, \mathit{fin})$.

Further, we can show that no number strictly less than 4 can be a cut-off for this protocol. Indeed, suppose $C^n_{\mathit{init}} \xRightarrow{*} C^n_{\mathit{fin}}$. Since we need at least two agents for a transition to occur, it follows that $n \geq 2$. By construction of the protocol, the first transition along this run must be $C^n_{\mathit{init}} \Rightarrow \wr (n-2) \cdot \mathit{init} + 2 \cdot q_1 \int$. If $n = 2$, then the run gets stuck at this configuration, because no agent is at a state capable of receiving a message. If $n = 3$, then the only transition that is possible is $\wr \mathit{init} + 2 \cdot q_1 \int \Rightarrow \wr q_1 + 2 \cdot \mathit{fin} \int$, at which point we reach a configuration where no agent is capable of receiving a message. This implies that $n \geq 4$ and so no number strictly less than 4 can be a cut-off for this protocol.

2.2. **Petri Nets.** We now formally define Petri nets and see how we can relate rendez-vous protocols to Petri nets.

Definition 2.3. A *Petri net* is a tuple $\mathcal{N} = (P, T, Pre, Post)$ where P is a finite set of *places*, T is a finite set of *transitions*, Pre and $Post$ are matrices whose rows and columns are indexed by P and T respectively and whose entries belong to $\mathbb{N}$. The *incidence matrix* $\mathcal{A}$ of $\mathcal{N}$ is defined to be the $P \times T$ matrix given by $\mathcal{A} = Post - Pre$. Further by the *weight* of $\mathcal{N}$, we mean the largest absolute value appearing in the matrices Pre and $Post$.

The size $|\mathcal{N}|$ of a Petri net $\mathcal{N}$ is defined as the number of bits needed to encode $\mathcal{N}$ using some suitable encoding. For a transition $t \in T$ we let ${}^\bullet t = \{p : Pre[p,t] > 0\}$ and $t^\bullet = \{p : Post[p,t] > 0\}$. We extend this notation to sets of transitions in the obvious way. Given a Petri net $\mathcal{N}$, we can associate with it a graph where the vertices are $P \cup T$ and the edges are $\{(p,t) : p \in {}^\bullet t\} \cup \{(t,p) : p \in t^\bullet\}$. A Petri net $\mathcal{N}$ is called acyclic if its associated graph is acyclic.

A *marking* of a Petri net is a multiset $M \in \mathbb{N}^P$, which intuitively denotes the number of *tokens* that are present in every place of the net. For $t \in T$ and markings M and M', we say that M' is reached from M by firing t, denoted by $M \xrightarrow{t} M'$, if for every place p, $M(p) \geq Pre[p,t]$ and $M'(p) = M(p) + \mathcal{A}[p,t]$.

A *firing sequence* is any sequence of transitions $\sigma = t_1, t_2, \dots, t_k \in T^*$. The support of σ, denoted by $[\![\sigma]\!]$, is the set of all transitions which appear in σ. We let $\sigma\sigma'$ denote the concatenation of two sequences σ and σ' and we let σ^k denote the concatenation of σ with itself k times.

Given a firing sequence $\sigma = t_1, t_2, \dots, t_k$, we let $M \xrightarrow{\sigma} M'$ denote that there are markings $M_1, \dots, M_{k-1}$ such that $M \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \dots M_{k-1} \xrightarrow{t_k} M'$. Further, $M \rightarrow M'$ denotes that there exists $t \in T$ such that $M \xrightarrow{t} M'$, and $M \xrightarrow{*} M'$ denotes that there exists $\sigma \in T^*$ such that $M \xrightarrow{\sigma} M'$.

In the following, we will use the notation $M \xrightarrow{\sigma}$ (resp. $\xrightarrow{\sigma} M$) to denote that there exists a marking M' such that $M \xrightarrow{\sigma} M'$ (resp. $M' \xrightarrow{\sigma} M$).

The monotonicity property. Throughout the paper, we will repeatedly use the following property of Petri nets, called the *monotonicity property.* It roughly states that adding more tokens to a marking does not stop us from firing a firing sequence which was fireable before.

Proposition 2.4. *Suppose $M \xrightarrow{\sigma} M'$. Then $M + L \xrightarrow{\sigma} M' + L$ for any marking L.*

Proof. By induction on the length of σ. □

Marking equation of a Petri net system. A *Petri net system* is a triple $(\mathcal{N}, M, M')$ where $\mathcal{N}$ is a Petri net and M and M' are markings. The *marking equation* for $(\mathcal{N}, M, M')$ is the equation

$$M' = M + \mathcal{A}\mathbf{v}$$

over the variables $\mathbf{v}$. It is well known that $M \xrightarrow{\sigma} M'$ implies $M' = M + \mathcal{A}\overrightarrow{\sigma}$, where $\overrightarrow{\sigma} \in \mathbb{N}^T$ is the *Parikh image* of σ, defined as the vector whose component $\overrightarrow{\sigma}[t]$ for a transition t is equal to the number of times t appears in σ. Therefore, if $M \xrightarrow{\sigma} M'$ then $\overrightarrow{\sigma}$ is a nonnegative integer solution of the marking equation. However, the converse does not hold.

2.3. From rendez-vous protocols to Petri nets.

We now show that rendez-vous protocols can be seen as a special class of Petri nets. Indeed, rendez-vous protocols can be thought of as Petri nets in which no tokens are created or destroyed during a run.

Let $\mathcal{P} = (Q, \Sigma, init, fin, R)$ be a rendez-vous protocol. Create a Petri net $\mathcal{N}_\mathcal{P} = (P, T, Pre, Post)$ as follows. The set of places is Q. For each message $a \in \Sigma$ and for each pair of rules $r = (q, !a, s),\ r' = (q', ?a, s') \in R$, add a transition $t_{r,r'}$ to $\mathcal{N}_\mathcal{P}$ and set

- $Pre[p, t_{r,r'}] = 0$ for every $p \notin \{q, q'\}$, $Post[p, t_{r,r'}] = 0$ for every $p \notin \{s, s'\}$
- If $q = q'$ then $Pre[q, t_{r,r'}] = 2$, otherwise $Pre[q, t_{r,r'}] = Pre[q', t_{r,r'}] = 1$
- If $s = s'$ then $Post[s, t_{r,r'}] = 2$, otherwise $Post[s, t_{r,r'}] = Post[s', t_{r,r'}] = 1$

Intuitively, the transition $t_{r,r'}$ affects only the places in the set $\{q, q', s, s'\}$. Firing $t_{r,r'}$ involves removing two tokens from q if $q = q'$, and otherwise one token each from q and q'. Then, after removing these tokens, we put back two tokens in s if $s = s'$, and otherwise we put one token each in s and s'.

Due to the way $\mathcal{N}_\mathcal{P}$ is defined, we have that any configuration of the protocol $\mathcal{P}$ is also a marking of $\mathcal{N}_\mathcal{P}$, and vice versa. Further, we have the following proposition, whose proof immediately follows from the definition of $\mathcal{N}_\mathcal{P}$.

Proposition 2.5. *For any pair of configurations C, C' and any pair of rules r, r' we have that $C \xRightarrow{r,r'} C'$ over the protocol $\mathcal{P}$ if and only if $C \xrightarrow{t_{r,r'}} C'$ over the Petri net $\mathcal{N}_\mathcal{P}$. Consequently, it follows that $C \xRightarrow{*} C'$ over the protocol $\mathcal{P}$ if and only if $C \xrightarrow{*} C'$ over the Petri net $\mathcal{N}_\mathcal{P}$.*

We can now define *the cut-off problem for Petri nets* in the following manner.

Given : A Petri net system $(\mathcal{N}, M, M')$
Decide: Does there exist $B \in \mathbb{N}$ such that $n \cdot M \xrightarrow{*} n \cdot M'$ for every $n \geq B$?

If such a B exists, then we say that $(\mathcal{N}, M, M')$ admits a cut-off and that B is a cut-off for $\mathcal{P}$. By Proposition 2.5, note that B is a cut-off for a protocol $\mathcal{P}$ if and only if B is a cut-off for the Petri net system $(\mathcal{N}, \wr init \wr, \wr fin \wr)$. Hence, the cut-off problem for Petri nets generalizes the cut-off problem for rendez-vous protocols.

Example 2.6. Let us consider the rendez-vous protocol $\mathcal{P}$ from Example 2.2. Its associated Petri net $\mathcal{N}_\mathcal{P}$ is given in Figure 2.

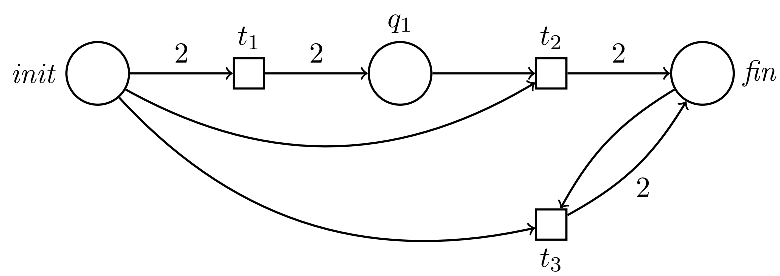

Figure 2: Petri net corresponding to the protocol from Figure 1

The three places of the Petri net correspond to the three states of the protocol $\mathcal{P}$. We also have three transitions: t_1 corresponds to the pair $(init, !a, q_1), (init, ?a, q_1)$, t_2 corresponds to the pair $(q_1, !b, fin), (init, ?b, fin)$ and t_3 corresponds to the pair $(fin, !b, fin), (init, ?b, fin)$. By Proposition 2.5 and by the argument given in Example 2.2, we can show that $C^n_{init} \xrightarrow{*} C^n_{fin}$ is a run in the Petri net $\mathcal{N}_\mathcal{P}$ if and only if $n \geq 4$. Hence, 4 is a cut-off for $(\mathcal{N}_\mathcal{P}, \wr init \int, \wr fin \int)$ and no number less than 4 can be a cut-off.

Our main result in this paper is the following:

Theorem 2.7. *The cut-off problem for Petri nets is decidable in polynomial time.*

Since the construction of $\mathcal{N}_\mathcal{P}$ can be done in polynomial time in the size of $\mathcal{P}$, by Proposition 2.5, we then get that:

Corollary 2.8. *The cut-off problem for rendez-vous protocols is decidable in polynomial time.*

3. The cut-off problem for acyclic Petri nets

As a warm-up to the cut-off problem, we first show that the cut-off problem for *acyclic Petri nets* can be solved in polynomial time. The reason for considering this special case first is that it illustrates one of the main ideas of the general case in a very pure form.

Let us fix a Petri net system $(\mathcal{N}, M, M')$ for the rest of this section, where $\mathcal{N} = (P, T, Pre, Post)$ is acyclic and $\mathcal{A}$ is its incidence matrix. It is well-known that in acyclic Petri nets the reachability relation is characterized by the marking equation.

Proposition 3.1 [Mur89, Theorem 16]**.** *Let $(\mathcal{N}, M, M')$ be an acyclic Petri net system. For every vector $\mathbf{x} \in \mathbb{N}^T$, $\mathbf{x}$ is a solution of the marking equation if and only if there is a firing sequence σ such that $\overrightarrow{\sigma} = \mathbf{x}$ and $M \xrightarrow{\sigma} M'$. Consequently, $M \xrightarrow{*} M'$ if and only if the marking equation has a nonnegative integer solution.*

This proposition shows that the reachability problem for acyclic Petri nets reduces to the feasibility problem (i.e., deciding the existence of a solution) for systems of linear Diophantine equations over the nonnegative integers. So the reachability problem for acyclic Petri nets is in NP, and in fact both the reachability and the feasibility problems are NP-complete [EN94].

There are two ways to relax the conditions on the solution so as to make the feasibility problem solvable in polynomial time. Feasibility over the non-negative *rationals* and feasibility over all integers for systems of linear equations are both in P. Indeed, the feasibility problem over the non-negative rationals is simply an instance of the linear programming problem, which is in polynomial time. Further, feasibility in $\mathbb{Z}$ can be decided in polynomial time after computing the Smith or Hermite normal forms of the matrix in the marking equation (see e.g. [PZ89]), which can themselves be computed in polynomial time [KB79]. We now show that the cut-off problem for acyclic Petri net systems can be reduced to solving a polynomial number of instances of the linear programming problem and the feasibility problem for systems of linear equations over integers.

3.1. **Characterizing acyclic systems with cut-offs.** Horn and Sangnier proved in [HS20] a very useful characterization of cut-off admitting rendez-vous protocols: A rendez-vous protocol $\mathcal{P}$ admits a cut-off if and only if there exists $n \in \mathbb{N}$ such that $C^n_{init} \xRightarrow{*} C^n_{fin}$ and $C^{n+1}_{init} \xRightarrow{*} C^{n+1}_{fin}$. Their proof immediately generalizes to the case of Petri nets. Here, we refine their characterization and proof in a small, but important way, which will be helpful later on.

Lemma 3.2 [HS20, Lemma 23]. *A Petri net system $(\mathcal{N}, M, M')$ (acyclic or not) admits a cut-off if and only if there exists $n \in \mathbb{N}$ and firing sequences σ, σ' such that $n \cdot M \xrightarrow{\sigma} n \cdot M'$, $(n+1) \cdot M \xrightarrow{\sigma'} (n+1) \cdot M'$ and $[\![\sigma']\!] \subseteq [\![\sigma]\!]$. Moreover if such n, σ and σ' exist, then n^2 is a cut-off for the system.*

Proof. Suppose $(\mathcal{N}, M, M')$ admits a cut-off. Hence there exists $B \in \mathbb{N}$ such that for all $n \geq B$, there is a firing sequence σ_n satisfying $nM \xrightarrow{\sigma_n} nM'$. Let $T' = \{t_1, \dots, t_k\}$ be the set $\bigcup_{n \geq B} [\![\sigma_n]\!]$. For every transition t_i in T', let n_i be such that σ_{n_i} contains an occurrence of t_i. Further, let $n = \sum_{t_i \in T'} n_i$ and $\sigma = \sigma_{n_1}\sigma_{n_2}\dots\sigma_{n_k}$. Notice that, since $n_i M \xrightarrow{\sigma_{n_i}} n_i M'$ for each i, by the monotonicity property we have $nM \xrightarrow{\sigma} nM'$. Since σ contain at least one occurrence of each transition in T', if we set $\sigma' = \sigma_{n+1}$, the claim is satisfied.

Suppose there exists $n \in \mathbb{N}$ and firing sequences σ, σ' such that $n \cdot M \xrightarrow{\sigma} n \cdot M'$, $(n+1) \cdot M \xrightarrow{\sigma'} (n+1) \cdot M'$ and $[\![\sigma']\!] \subseteq [\![\sigma]\!]$. Let $s \geq n^2$. We can write s as $s = qn + r$ for some $q \in \mathbb{N}$ and some r such that $0 \leq r \leq n-1$. Since $s \geq n^2$, it follows that $q \geq n > r$. Hence, we can rewrite s as $s = r(n+1) + (q-r)n$. By the monotonicity property, it then follows that $sM \xrightarrow{(\sigma')^r(\sigma)^{(q-r)}} sM'$. Hence, n^2 is a cut-off for the system. □

Using this lemma, we give a characterization of those acyclic Petri net systems that admit a cut-off.

Theorem 3.3. *An acyclic Petri net system $(\mathcal{N}, M, M')$ admits a cut-off if and only if the marking equation has solutions $\mathbf{x} \in \mathbb{Q}^T_{\geq 0}$ and $\mathbf{y} \in \mathbb{Z}^T$ such that $[\![\mathbf{y}]\!] \subseteq [\![\mathbf{x}]\!]$.*

Proof. ($\Rightarrow$): Suppose $(\mathcal{N}, M, M')$ admits a cut-off. By Lemma 3.2, there exists $n \in \mathbb{N}$ and firing sequences σ, σ' such that $nM \xrightarrow{\sigma} nM'$, $(n+1)M \xrightarrow{\sigma'} (n+1)M'$ and $[\![\sigma']\!] \subseteq [\![\sigma]\!]$. By Proposition 3.1, this means that there exist $\mathbf{x}', \mathbf{y}' \in \mathbb{N}^T$ such that $[\![\mathbf{y}']\!] \subseteq [\![\mathbf{x}']\!]$, $nM' = nM + \mathcal{A}\mathbf{x}'$ and $(n+1)M' = (n+1)M + \mathcal{A}\mathbf{y}'$. Letting $\mathbf{x} = \mathbf{x}'/n$ and $\mathbf{y} = \mathbf{y}' - \mathbf{x}'$, we get our required vectors.

($\Leftarrow$): Suppose $\mathbf{x} \in \mathbb{Q}^T_{\geq 0}$ and $\mathbf{y} \in \mathbb{Z}^T$ are solutions of the marking equation such that $[\![\mathbf{y}]\!] \subseteq [\![\mathbf{x}]\!]$. Let μ be the least common multiple of the denominators of the components of $\mathbf{x}$, and let α be the largest absolute value of the numbers in the vector $\mathbf{y}$. By definition of μ we have $\alpha(\mu\mathbf{x}) \in \mathbb{N}^T$. Also, since $[\![\mathbf{y}]\!] \subseteq [\![\mathbf{x}]\!]$, it follows by definition of α that $[\![\alpha\mu\mathbf{x} + \mathbf{y}]\!] \subseteq [\![\alpha\mu\mathbf{x}]\!]$ and $\alpha(\mu\mathbf{x}) + \mathbf{y} \geq \mathbf{0}$. Since $M' = M + \mathcal{A}\mathbf{x}$ and $M' = M + \mathcal{A}\mathbf{y}$ we get

$$\alpha\mu M' = \alpha\mu M + \mathcal{A}(\alpha\mu\mathbf{x}) \qquad \text{and} \qquad (\alpha\mu+1)M' = (\alpha\mu+1)M + \mathcal{A}(\alpha\mu\mathbf{x} + \mathbf{y})$$

Taking $\alpha\mu = n$, by Proposition 3.1 we get that there are firing sequences σ and σ' such that $\overrightarrow{\sigma} = \mu\alpha\mathbf{x}$, $\overrightarrow{\sigma'} = \mu\alpha\mathbf{x} + \mathbf{y}$, $nM \xrightarrow{\sigma} nM'$ and $(n+1)M \xrightarrow{\sigma'} (n+1)M'$. Since $[\![\alpha\mu\mathbf{x} + \mathbf{y}]\!] \subseteq [\![\alpha\mu\mathbf{x}]\!]$, by Lemma 3.2, $(\mathcal{N}, M, M')$ admits a cut-off. □

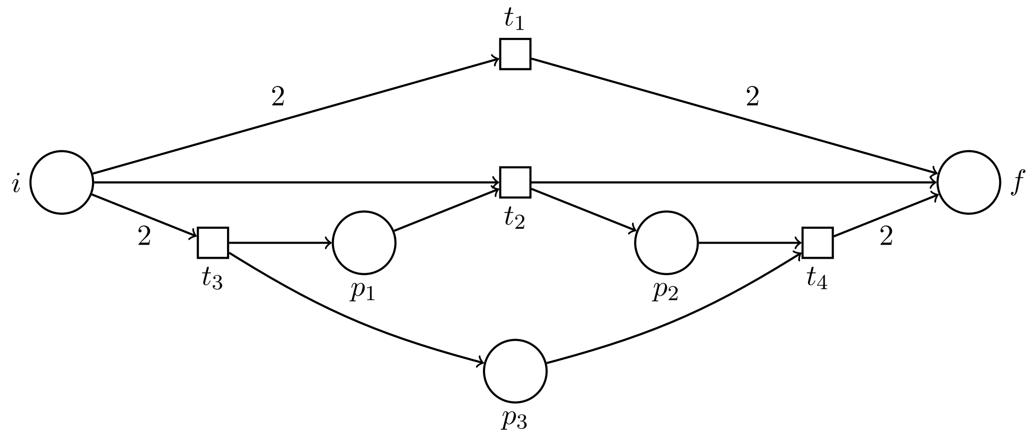

Figure 3: A net with cut-off 2.

Intuitively, the existence of the rational solution $\mathbf{x} \in \mathbb{Q}^T_{\geq 0}$ guarantees that $nM \xrightarrow{*} nM'$ for infinitely many n, and the existence of the integer solution $\mathbf{y} \in \mathbb{Z}^T$ guarantees that for one of those n we have $(n+1)M \xrightarrow{*} (n+1)M'$ as well.

Example 3.4. Consider the acyclic net system given by the net on Figure 3 along with the markings $M = \wr i \int$ and $M' = \wr f \int$. We claim that 2 is a cut-off for this system. Indeed, first notice that for every $k \geq 1$, we have $2kM \xrightarrow{t_1^k} 2kM'$. Further, for every $k \geq 1$, we have $(2k+1)M \xrightarrow{t_3 t_2 t_4} (2k-2)M + 3M' \xrightarrow{t_1^{k-1}} (2k+1)M'$. Hence, 2 is a cut-off for this system.

We now show that the conditions of Theorem 3.3 are satisfied for this system. Notice that the marking equation for M and M' gives the following five equations:

$$\begin{aligned}
0 &= 1 - 2\mathbf{v}_{t_1} - \mathbf{v}_{t_2} - 2\mathbf{v}_{t_3} && \text{(Equation for the place } i\text{)} \\
0 &= 0 - \mathbf{v}_{t_2} + \mathbf{v}_{t_3} && \text{(Equation for the place } p_1\text{)} \\
0 &= 0 + \mathbf{v}_{t_2} - \mathbf{v}_{t_4} && \text{(Equation for the place } p_2\text{)} \\
0 &= 0 + \mathbf{v}_{t_3} - \mathbf{v}_{t_4} && \text{(Equation for the place } p_3\text{)} \\
1 &= 0 + 2\mathbf{v}_{t_1} + \mathbf{v}_{t_2} + 2\mathbf{v}_{t_4} && \text{(Equation for the place } f\text{)}
\end{aligned}$$

Notice that $\mathbf{x} = (\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$ and $\mathbf{y} = (-1, 1, 1, 1)$ are both solutions to these equations. Further, since $[\![\mathbf{y}]\!] \subseteq [\![\mathbf{x}]\!]$, the conditions of Theorem 3.3 are satisfied.

3.2. **Polynomial time algorithm.** We derive a polynomial time algorithm for the cut-off problem from the characterization of Theorem 3.3. The first step is the following lemma. A very similar lemma is proved in [FH15], but since the proof is short we give it for the sake of completeness.

Lemma 3.5. *If the marking equation is feasible over $\mathbb{Q}_{\geq 0}$, then one can compute in polynomial time a solution $\mathbf{u}$ such that for every solution $\mathbf{y}$, $[\![\mathbf{y}]\!] \subseteq [\![\mathbf{u}]\!]$.*

Proof. If $\mathbf{y}, \mathbf{z} \in \mathbb{Q}_{\geq 0}^T$ are solutions of the marking equation, then we have $M' = M + \mathcal{A}((\mathbf{y} + \mathbf{z})/2)$ and $[\![\mathbf{y}]\!] \cup [\![\mathbf{z}]\!] \subseteq [\![(\mathbf{y}+\mathbf{z})/2]\!]$. Hence if the marking equation is feasible over $\mathbb{Q}_{\geq 0}$, then there is a solution $\mathbf{u}$ such that for every solution $\mathbf{y}$, $[\![\mathbf{y}]\!] \subseteq [\![\mathbf{u}]\!]$.

To find such a solution in polynomial time we proceed as follows. For every transition t we solve the linear program $M' = M + \mathcal{A}\mathbf{v}, \mathbf{v} \geq \mathbf{0}, \mathbf{v}_t > 0$. (Recall that solving linear programs over the non-negative rationals can be done in polynomial time). Let $\{t_1, \ldots, t_n\}$ be the set of transitions whose associated linear programs are feasible over $\mathbb{Q}_{\geq 0}^T$, and let $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ be solutions to these programs. Then $\mathbf{u} = 1/n \cdot \sum_{i=1}^n \mathbf{u}_i$ is a solution of the marking equation that satisfies the desired property. □

We now have all the ingredients to give a polynomial time algorithm.

Theorem 3.6. *The cut-off problem for acyclic net systems can be solved in polynomial time.*

Proof. First, we check that the marking equation has a solution over the non-negative rationals. If such a solution does not exist, by Theorem 3.3 the given net system does not admit a cut-off.

Suppose such a solution exists. By Lemma 3.5 we can find a non-negative rational solution $\mathbf{x}$ with maximum support in polynomial time. Let U contain all the transitions t such that $\mathbf{x}_t = 0$. We now check in polynomial time if the marking equation has a solution $\mathbf{y}$ over $\mathbb{Z}^T$ such that $\mathbf{y}_t = 0$ for every $t \in U$. By Theorem 3.3 such a solution exists if and only if the net system admits a cut-off. □

4. The Scaling and Insertion lemmas

Similar to the case of acyclic net systems, we would like to provide a characterization of net systems admitting a cut-off and then use this characterization to derive a polynomial time algorithm. Unfortunately, for general net systems there is no characterization of reachability akin to Proposition 3.1 for acyclic systems. To this end, we prove two intermediate lemmas to help us come up with a characterization for (general) net systems which admit a cut-off. We believe that these two lemmas could be of independent interest in their own right. Further, the proofs of both the lemmas are provided so that it will enable us later on to derive a bound on the cut-off for net systems.

4.1. **The Scaling Lemma.**

The Scaling Lemma shows that, given a Petri net system $(\mathcal{N}, M, M')$, deciding whether $nM \xrightarrow{*} nM'$ holds for some $n \geq 1$ can be done in polynomial time; moreover, if $nM \xrightarrow{*} nM'$ holds for some n, then it holds for some n which can be described by at most $(|\mathcal{N}|(\log \|M\| + \log \|M'\|))^{O(1)}$ bits. The name of the lemma is due to the fact that the firing sequence leading from nM to nM' is obtained by *scaling up* a *continuous firing sequence* from M to M'; the existence of such a continuous firing sequence can be decided in polynomial time due to a result by Fraca and Haddad [FH15].

In the rest of the section we first recall continuous Petri nets and the characterization of [FH15], and then present the Scaling Lemma. As mentioned in the introduction, this lemma is implicitly proved in [FH15], but the bound on n is hidden in the details of the proof, and we make it explicit here.

4.1.1. *Reachability in continuous Petri nets.* Petri nets can be given a *continuous semantics* (see e.g. [AD98, RHS10, FH15]), in which markings are continuous multisets; we call them *continuous markings.* A continuous marking M enables a transition t *with factor* $\lambda \in (0,1]$ if $M(p) \geq \lambda \cdot Pre[p,t]$ for every place p; we also say that M enables λt. If M enables λt, then λt can fire or occur, leading to a new marking M' given by $M'(p) = M(p) + \lambda \cdot \mathcal{A}[p,t]$ for every $p \in P$. We denote this by $M \xrightarrow[\mathbb{Q}]{\lambda t} M'$, and say that M' is reached from M by firing λt. A *continuous firing sequence* is any sequence of the form $\sigma = \lambda_1 t_1, \lambda_2 t_2, \ldots, \lambda_k t_k \in ((0,1] \times T)^*$. We let $M \xrightarrow[\mathbb{Q}]{\sigma} M'$ denote that there exist continuous markings $M_1, \ldots, M_{k-1}$ such that $M \xrightarrow[\mathbb{Q}]{\lambda_1 t_1} M_1 \xrightarrow[\mathbb{Q}]{\lambda_2 t_2} M_2 \cdots M_{k-1} \xrightarrow[\mathbb{Q}]{\lambda_k t_k} M'$. We can then define $M \xrightarrow[\mathbb{Q}]{*} M'$, $M \xrightarrow[\mathbb{Q}]{\sigma}$ and $\xrightarrow[\mathbb{Q}]{\sigma} M$ in a similar manner as before.

The *Parikh image* of $\sigma = \lambda_1 t_1, \lambda_2 t_2, \ldots, \lambda_k t_k$ is the vector $\overrightarrow{\sigma} \in \mathbb{Q}^T_{\geq 0}$ where $\overrightarrow{\sigma}[t] = \sum_{i=1}^k \delta_{i,t}\lambda_i$, where $\delta_{i,t} = 1$ if $t_i = t$ and 0 otherwise. The support of σ is the support of its Parikh image $\overrightarrow{\sigma}$. If $M \xrightarrow[\mathbb{Q}]{\sigma} M'$ then $\overrightarrow{\sigma}$ is a solution of the marking equation over $\mathbb{Q}^T_{\geq 0}$, but the converse does not hold. In [FH15], Fraca and Haddad strengthen this necessary condition to make it also sufficient, and use the resulting characterization to derive a polynomial time algorithm.

Theorem 4.1 [FH15, Theorem 20 and Proposition 27]**.** *Let $(\mathcal{N}, M, M')$ be a Petri net system. Then the following statements are true:*

- *For a continuous firing sequence σ, $M \xrightarrow[\mathbb{Q}]{\sigma} M'$ is true if and only if $\overrightarrow{\sigma}$ is a solution of the marking equation over $\mathbb{Q}^T_{\geq 0}$ and there exist continuous firing sequences τ, τ' such that $[\![\tau]\!] = [\![\sigma]\!] = [\![\tau']\!]$, $M \xrightarrow[\mathbb{Q}]{\tau}$ and $\xrightarrow[\mathbb{Q}]{\tau'} M'$.*
- *It can be decided in polynomial time if $M \xrightarrow[\mathbb{Q}]{*} M'$ holds.*

4.1.2. *Scaling.* It follows easily from the definitions that $nM \xrightarrow{*} nM'$ holds for some $n \geq 1$ if and only if $M \xrightarrow[\mathbb{Q}]{*} M'$. Indeed, if $M \xrightarrow[\mathbb{Q}]{\sigma} M'$ for some continuous firing sequence $\sigma = \lambda_1 t_1, \lambda_2 t_2, \ldots, \lambda_k t_k$, then we can scale this continuous firing sequence to a discrete sequence $nM \xrightarrow[\mathbb{Q}]{n\sigma} nM'$ where n is the smallest number such that $n\lambda_1, \ldots, n\lambda_k \in \mathbb{N}$, and $n\sigma = t_1^{n\lambda_1} t_2^{n\lambda_2} \ldots t_k^{n\lambda_k}$. For the other direction if $nM \xrightarrow{\sigma} nM'$ holds for some $n \geq 1$ and some $\sigma = t_1, \ldots, t_k$, then $M \xrightarrow[\mathbb{Q}]{\sigma/n} M'$ is true where σ/n is the continuous firing sequence given by $\sigma/n = t_1/n, t_2/n, \ldots, t_k/n$. So Theorem 4.1 immediately implies that we can decide in polynomial time if there is a number $n \geq 1$ satisfying $nM \xrightarrow{*} nM'$. The following lemma also gives a bound on n.

Lemma 4.2. *Let $(\mathcal{N}, M, M')$ be a Petri net system with weight w such that $M \xrightarrow[\mathbb{Q}]{\sigma} M'$ for some continuous firing sequence σ. Let m be the number of transitions in $[\![\sigma]\!]$ and let ℓ be $\|\overrightarrow{\sigma}\|$. Let k be the smallest natural number such that $k\overrightarrow{\sigma} \in \mathbb{N}^T$. Then, there exists a firing sequence $\tau \in T^*$ such that $[\![\tau]\!] = [\![\sigma]\!]$, $\|\overrightarrow{\tau}\| \leq \mu\ell$ and*

$$\mu \cdot M \xrightarrow{\tau} \mu \cdot M'$$

where $\mu = 16w(w+1)^{2m}k^2\ell$.

We shall first give an intuitive idea behind the proof of Lemma 4.2. Assume that $M \xrightarrow[\mathbb{Q}]{\sigma} M'$. For the purpose of studying this firing sequence, we can remove from the net, all transitions that do not occur in σ, and all places that are neither input nor output places of some transition of σ. Let $\mathcal{N}'$ be the resulting net. We show that $\beta M \xrightarrow{*} \beta M'$ for a sufficiently large β. In a first step, we show in Lemma 4.3, that for a sufficiently large n we can find markings M_1 and M_2 that mark every place of $\mathcal{N}'$ and satisfy the following properties: $nM \xrightarrow{*} M_1$, $M_2 \xrightarrow{*} nM'$. Hence, if we show that $n'M_1 \xrightarrow{*} n'M_2$ is true for a sufficiently large n', then by the monotonicity property, we would have $nn'M \xrightarrow{*} n'M_1 \xrightarrow{*} n'M_2 \xrightarrow{*} nn'M'$ and we would be done. To show that such an n' exists, we apply a folklore lemma showing that if $L_2 = L_1 + \mathcal{A}(m\mathbf{x})$ for some number m and some $\mathbf{x} \in \mathbb{N}^T$, and $L_1 \xrightarrow{\tau}$ and $\xrightarrow{\tau} L_2$ for some firing sequence τ such that $\overrightarrow{\tau} = \mathbf{x}$, then $L_1 \xrightarrow{\tau^m} L_2$ where τ^m denotes the concatentation of τ with itself m times (Lemma 4.4). Finally, we show in Lemma 4.3 itself that M_1 and M_2 can be chosen so that for sufficiently large n', the markings $L_1 := n'M_1$ and $L_2 := n'M_2$ satisfy the preconditions of Lemma 4.4.

We now proceed to prove Lemmas 4.3 and 4.4 and then we prove Lemma 4.2 using these two lemmas.

Lemma 4.3. *Let M be a marking such that $M \xrightarrow[\mathbb{Q}]{\sigma}$ for some continuous firing sequence σ. Let m be the size of $[\![\sigma]\!]$ and let w be the weight of $\mathcal{N}$. Then there exists a firing sequence*

$$((w+1)^m \cdot M) \xrightarrow{\tau} L$$

such that $[\![\tau]\!] = [\![\sigma]\!]$, $\|\overrightarrow{\tau}\| \leq 2(w+1)^m$ and $L(p) > 0$ for every $p \in [\![M]\!] \cup {}^\bullet[\![\sigma]\!] \cup [\![\sigma]\!]^\bullet$.

Proof. Let $t_1, t_2, \ldots, t_m$ be the transitions of $[\![\sigma]\!]$, sorted according to the order of their first occurrence in σ. Let $\beta_0 = 1$, $\beta_i = (w+1)^i$ for every $i \in \{1, \ldots, m\}$ and define the sequence

$$\tau'_0 = \epsilon$$

$$\tau'_i = t_1^{\beta_{i-1}} t_2^{\beta_{i-2}} \cdots t_i^{\beta_0}$$

For each $i \in \{0, \ldots, m\}$, we now show that there exists a firing sequence τ_i and a marking M_i such that $(\beta_i \cdot M) \xrightarrow{\tau_i} M_i$, the Parikh images of τ_i and τ'_i are the same and $M_i(p) > 0$ for every $p \in [\![M]\!] \cup {}^\bullet[\![\tau_i]\!] \cup [\![\tau_i]\!]^\bullet$. If this claim is true, then by definition of each β_i, we have $\|\overrightarrow{\tau_i}\| = \|\overrightarrow{\tau'_i}\| \leq 2(w+1)^i$ and also that $[\![\tau_m]\!] = [\![\sigma]\!]$. Hence, the lemma would then follow by taking $L := M_m$. Therefore, all that remains is to prove the claim which we do by induction on i.

Basis: $i = 0$. Then $\beta_i = 1$ and the result follows from $M \xrightarrow{\epsilon} M$.

Induction Step: $i \geq 1$. By induction hypothesis,

$$(\beta_{i-1} \cdot M) \xrightarrow{\tau_{i-1}} M_{i-1}$$

where $M_{i-1}(p) > 0$ for every $p \in [\![M]\!] \cup {}^\bullet[\![\tau_{i-1}]\!] \cup [\![\tau_{i-1}]\!]^\bullet$. Since $\beta_i = (w+1)\beta_{i-1}$ we have

$$(\beta_i \cdot M) \xrightarrow{\tau_{i-1}^{(w+1)}} ((w+1) \cdot M_{i-1})$$

Since $M \xrightarrow[\mathbb{Q}]{\sigma}$ and t_i appears in σ after $t_1, \ldots, t_{i-1}$, it follows that every place $p \in {}^\bullet t_i$ has at least $(w+1)$ tokens at the marking $(w+1) \cdot M_{i-1}$. So $(w+1) \cdot M_{i-1} \xrightarrow{t_i} M_i$ for some marking M_i such that $M_i(p) > 0$ for every $p \in [\![M]\!] \cup {}^\bullet\{t_1, \ldots, t_i\} \cup \{t_1, \ldots, t_i\}^\bullet$. Hence, if we take τ_i to be the sequence $\tau_{i-1}^{(w+1)}\ t_i$, then the proof of the induction step becomes complete. ◻

Lemma 4.4. *Let $(\mathcal{N}, M, M')$ be a Petri net system, and let σ be a firing sequence such that $M \xrightarrow{\sigma}$, $\xrightarrow{\sigma} M'$ and $n\overrightarrow{\sigma}$ is a solution of the marking equation for some $n \geq 1$. Then $M \xrightarrow{\sigma^n} M'$.*

Proof. Since $n\overrightarrow{\sigma}$ is a solution of the marking equation, $M \xrightarrow{\sigma^n}$ implies $M \xrightarrow{\sigma^n} M'$. So it suffices to prove $M \xrightarrow{\sigma^n}$. We proceed by induction on n.

Basis: $n = 1$**.** Then $M \xrightarrow{\sigma}$ follows immediately from the assumptions.

Induction hypothesis: Assume that for some number n and for all Petri net systems $(\mathcal{N}, M, M')$ and firing sequences σ for which $M \xrightarrow{\sigma}$, $\xrightarrow{\sigma} M'$ and $n\overrightarrow{\sigma}$ is a solution to the marking equation, we have shown that $M \xrightarrow{\sigma^n} M'$.

Induction step: We now prove the claim for $n+1$. Let M_1, M_1' be the markings such that $M \xrightarrow{\sigma} M_1$ and $M_1' \xrightarrow{\sigma} M'$. We first claim that $M_1 \xrightarrow{\sigma}$ holds. Let $\sigma = t_1 t_2 \ldots t_m$, and let $L_1, \ldots, L_{m-1}, L_1', \ldots, L_{m-1}'$ be the markings given by

$$M \xrightarrow{t_1} L_1 \cdots L_{m-1} \xrightarrow{t_m} M_1 \qquad \text{and} \qquad M_1' \xrightarrow{t_1} L_1' \cdots L_{m-1}' \xrightarrow{t_m} M' \ .$$

Further, let $\sigma_0 := \epsilon$ and $\sigma_i := t_1 \cdots t_i$ for every $1 \leq i \leq m$, and define $K_i := M_1 + \mathcal{A} \cdot \overrightarrow{\sigma_i}$. We prove that $K_i \xrightarrow{t_{i+1}}$ holds for every $0 \leq i < m$, which implies the claim that $M_1 \xrightarrow{\sigma}$. To prove that $K_i \xrightarrow{t_{i+1}}$ holds, it suffices to show that $K_i(p) \geq \mathit{Pre}[p, t_{i+1}]$ holds for every place p. Since $L_i \xrightarrow{t_{i+1}}$ and $L_i' \xrightarrow{t_{i+1}}$, we have $L_i(p) \geq \mathit{Pre}[p, t_{i+1}]$ and $L_i'(p) \geq \mathit{Pre}[p, t_{i+1}]$. Notice that since $M' = M + \mathcal{A} \cdot ((n+1)\overrightarrow{\sigma})$, we have $L_i' = L_i + \mathcal{A} \cdot (n\overrightarrow{\sigma})$. Let $\sigma[i]$ be the cyclic permutation of σ starting at t_{i+1} (or equivalently, ending at t_i). By the definition of the marking K_i we have $K_i = L_i + \mathcal{A} \cdot \overrightarrow{\sigma[i]}$. Since $\mathcal{A} \cdot \overrightarrow{\sigma} = \mathcal{A} \cdot \overrightarrow{\sigma[i]}$, letting $\Delta := \mathcal{A} \cdot \overrightarrow{\sigma}$, we obtain altogether

$$\begin{array}{rcccl} L_i(p) & & & \geq & \mathit{Pre}[p, t_{i+1}] \\ K_i(p) & = & L_i(p) + \Delta(p) & & \\ L_i'(p) & = & L_i(p) + (n-1)\Delta(p) & \geq & \mathit{Pre}[p, t_{i+1}] \end{array}$$

This implies $K_i(p) \geq \mathit{Pre}[p, t_{i+1}]$, and the claim is proved. By the claim we have $M_1 \xrightarrow{\sigma}$; moreover $\xrightarrow{\sigma} M'$, and $n\overrightarrow{\sigma}$ is a solution of the marking equation for $(\mathcal{N}, M_1, M')$. By induction hypothesis $M_1 \xrightarrow{\sigma^{n-1}} M'$, and so $M \xrightarrow{\sigma^n} M'$. □

We also need the following minor proposition.

Proposition 4.5. *Let w be the weight of the net $\mathcal{N}$. Suppose M is a marking and σ is a firing sequence such that $M(p) \geq w \cdot \overrightarrow{\sigma}(t)$ for every $t \in [\![\sigma]\!]$ and for every $p \in {}^\bullet t$. Then $M \xrightarrow{\sigma}$.*

Proof. By induction on $\|\overrightarrow{\sigma}\|$. □

Using the above three auxiliary results, we are now ready to prove Lemma 4.2.

Proof of Lemma 4.2. Let $\beta := (w+1)^m$ and $\gamma := 4w\beta k\ell$. We split the proof into three parts, by providing the desired firing sequence as a concatenation of three firing sequences of the form

$$(4k\beta\gamma \cdot M) \xrightarrow{*} \gamma L_1 \xrightarrow{*} \gamma L_2 \xrightarrow{*} \big(4k\beta\gamma \cdot M'\big) \tag{4.1}$$

for some markings L_1 and L_2.

First firing sequence: Since $M \xrightarrow[\mathbb{Q}]{\sigma} M'$, by Lemma 4.3 there exists a firing sequence τ_1 and a marking M_1 such that $\|\overrightarrow{\tau_1}\| \leq 2\beta$, $[\![\tau_1]\!] = [\![\sigma]\!]$, $M_1(p) > 0$ for all $p \in {}^\bullet[\![\sigma]\!] \cup [\![\sigma]\!]^\bullet$ and $(\beta \cdot M) \xrightarrow{\tau_1} M_1$. Let $L_1 = (4k-1)\beta M + M_1$. By the monotonicity property, we have that $4k\beta M \xrightarrow{\tau_1} (4k-1)\beta M + M_1 = L_1$. Once again by the monotonicity property, we have $4k\beta\gamma M \xrightarrow{\tau_1^\gamma} \gamma L_1$, thereby completing the first part of the run. We note that by construction of L_1, we have $(\gamma L_1)(p) \geq \gamma$ for every $p \in {}^\bullet[\![\sigma]\!] \cup [\![\sigma]\!]^\bullet$.

Third firing sequence: The third part is similar to the first part, except that we apply Lemma 4.3 to the reverse net. Notice that since $\xrightarrow[\mathbb{Q}]{\sigma} M'$ in the Petri net $\mathcal{N}$, it follows that $M' \xrightarrow[\mathbb{Q}]{\sigma^{-1}}$ in the *reverse net* $\mathcal{N}^{-1}$ where $\mathcal{N}^{-1} := (P, T, Post, Pre)$ and σ^{-1} is the reverse of σ. Using Lemma 4.3, we can get a firing sequence τ_2' and a marking M_2 such that $\|\overrightarrow{\tau_2'}\| \leq 2\beta$, $[\![\tau_2']\!] = [\![\sigma]\!]$, $M_2(p) > 0$ for all $p \in {}^\bullet[\![\sigma]\!] \cup [\![\sigma]\!]^\bullet$ and $(\beta \cdot M') \xrightarrow{\tau_2'} M_2$ in the reverse net $\mathcal{N}^{-1}$. Let $L_2 = (4k-1)\beta M' + M_2$. By the same argument as the first part, we can conclude that $4k\beta\gamma M' \xrightarrow{\tau_2'^\gamma} \gamma L_2$ in the reverse net $\mathcal{N}^{-1}$. Letting $\tau_2 := \tau_2'^{-1}$ we get that $\gamma L_2 \xrightarrow{\tau_2^\gamma} 4k\beta\gamma M'$ in the net $\mathcal{N}$. We note that by construction of L_2, we have $(\gamma L_2)(p) \geq \gamma$ for every $p \in {}^\bullet[\![\sigma]\!] \cup [\![\sigma]\!]^\bullet$.

Second firing sequence: By construction of τ_1 and τ_2 we have that $\|\overrightarrow{\tau_1}\|, \|\overrightarrow{\tau_2}\| \leq 2\beta$. It follows then that $\|\overrightarrow{\tau_1}\| + \|\overrightarrow{\tau_2}\| \leq 4\beta$. Once again by construction of τ_1 and τ_2 we have that $[\![\tau_1]\!] = [\![\tau_2]\!] = [\![\sigma]\!]$. Further by construction of the number k, it follows that all the non-zero components of $4k\beta\overrightarrow{\sigma}$ are at least 4β. Hence, if we define

$$v := 4\beta k\overrightarrow{\sigma} - (\overrightarrow{\tau_1} + \overrightarrow{\tau_2}) \tag{4.2}$$

then v is a non-negative integer vector.

Let τ_3 be any firing sequence such that $\overrightarrow{\tau_3} = v$. We claim that $\gamma L_1 \xrightarrow{\tau_3}$, $\xrightarrow{\tau_3} \gamma L_2$ and $\gamma L_2 = \gamma L_1 + \mathcal{A} \cdot (\gamma \cdot v)$. Notice that if this claim is true, then by Lemma 4.4 we have that $\gamma L_1 \xrightarrow{\tau_3^\gamma} \gamma L_2$ and then the second part of equation (4.1) will also be done.

All that is left to prove are the three claims. As remarked at the end of the first part of the construction, notice that $(\gamma L_1)(p) \geq \gamma$ for every $p \in {}^\bullet[\![\sigma]\!] \cup [\![\sigma]\!]^\bullet$. Since $\overrightarrow{\tau_3} = v$, by equation (4.2), notice that $[\![\tau_3]\!] \subseteq [\![\sigma]\!]$ and $w \cdot \|\overrightarrow{\tau_3}\| \leq w \cdot 4\beta k \cdot \|\overrightarrow{\sigma}\| = w \cdot 4\beta k\ell = \gamma$. Hence it follows that $(\gamma L_1)(p) \geq w\|\overrightarrow{\tau_3}\|$ for every $p \in {}^\bullet[\![\tau_3]\!] \cup [\![\tau_3]\!]^\bullet$. Applying Proposition 4.5, we get that $\gamma L_1 \xrightarrow{\tau_1}$. By the same argument applied to the reverse net $\mathcal{N}^{-1}$, we get that $\gamma L_2 \xrightarrow{\tau_2^{-1}}$ in the reverse net $\mathcal{N}^{-1}$, hence leading to $\xrightarrow{\tau_2} \gamma L_2$.

Finally, notice that

$$\begin{aligned}
L_2 &= 4k\beta \cdot M' - \mathcal{A} \cdot \overrightarrow{\tau_2} && (L_2 \xrightarrow{\tau_2} 4k\beta \cdot M') \\
&= 4\beta k \cdot M + \mathcal{A} \cdot 4\beta k \cdot \overrightarrow{\sigma} - \mathcal{A} \cdot \overrightarrow{\tau_2} && (M \xrightarrow[\mathbb{Q}]{\sigma} M') \\
&= L_1 - \mathcal{A} \cdot \overrightarrow{\tau_1} + \mathcal{A} \cdot 4\beta k \cdot \overrightarrow{\sigma} - \mathcal{A} \cdot \overrightarrow{\tau_2} && (4k\beta \cdot M \xrightarrow{\tau_1} L_1) \\
&= L_1 + \mathcal{A} \cdot (4\beta k \cdot \overrightarrow{\sigma} - (\overrightarrow{\tau_1} + \overrightarrow{\tau_2})) \\
&= L_1 + \mathcal{A} \cdot v && (\text{Equation (4.2)})
\end{aligned}$$

and so $\gamma \cdot L_2 = \gamma \cdot L_1 + \mathcal{A} \cdot (\gamma \cdot v)$. Since all the three claims have been proven, it follows that equation (4.1) is true.

Now, let us analyse the support and the norm of the Parikh image of the final firing sequence that we obtain. Notice that the sequence that we construct is $4k\beta\gamma M \xrightarrow{\tau_1^\gamma} \gamma L_1 \xrightarrow{\tau_3^\gamma} \gamma L_2 \xrightarrow{\tau_2^\gamma} 4k\beta\gamma M'$. Let $\tau = \tau_1^\gamma \tau_3^\gamma \tau_2^\gamma$. By construction, we know that $[\![\tau_1]\!] = [\![\tau_2]\!] = [\![\sigma]\!]$ and $[\![\tau_3]\!] \subseteq [\![\sigma]\!]$. Hence, $[\![\tau]\!] = [\![\sigma]\!]$. Further, $\|\overrightarrow{\tau}\| = \gamma(\|\overrightarrow{\tau_1}\| + \|\overrightarrow{\tau_2}\| + \|\overrightarrow{\tau_3}\|)$ and applying equation (4.2), we get $\|\overrightarrow{\tau}\| = 4k\beta\gamma\|\overrightarrow{\sigma}\| = 4k\beta\gamma\ell$. This then completes the proof of Lemma 4.2. □

We can now prove our main result of this subsection, namely the Scaling Lemma.

Lemma 4.6 (**Scaling Lemma**). *Let $(\mathcal{N}, M, M')$ be a net system such that $M \xrightarrow[\mathbb{Q}]{\sigma} M'$ for some σ. Then there exist n and m which can be described by a polynomial number of bits in $|\mathcal{N}|(\log\|M\| + \log\|M'\|)$ such that $nM \xrightarrow{\sigma} nM'$ for some τ with $[\![\tau]\!] = [\![\sigma]\!]$ and $\|\overrightarrow{\tau}\| \leq m$.*

Proof. Suppose $M \xrightarrow[\mathbb{Q}]{\sigma} M'$. Let U be the support of σ. By [BFHH17, Proposition 3.2], there is a formula $\phi(M, M', \mathbf{v})$ in the existential theory of linear rational arithmetic $\mathrm{Th}(\mathbb{Q}, +, <)$, whose size is linear in the size of the net $\mathcal{N}$ such that $\phi(M, M', \mathbf{x})$ is true if and only if there exists σ' such that $\overrightarrow{\sigma'} = \mathbf{x}$ and $M \xrightarrow[\mathbb{Q}]{\sigma'} M'$. To this formula, let us add the constraints $\mathbf{v}_t > 0 \iff t \in U$ and let the resulting formula be ξ. Note that $\xi(M, M', \mathbf{x})$ is true if and only if there exists σ' such that $\overrightarrow{\sigma'} = \mathbf{x}, [\![\sigma']\!] = U$ and $M \xrightarrow[\mathbb{Q}]{\sigma'} M'$.

By [Son85, Lemma 3.2] if a formula in the existential theory of linear rational arithmetic is satisfiable, then it is satisfiable by a solution which can be described using a polynomial number of bits in the size of the formula. Hence, applying this result to the formula $\xi(M, M', \mathbf{v})$, we get that there exists τ' such that $M \xrightarrow[\mathbb{Q}]{\tau'} M'$, $[\![\tau']\!] = [\![\sigma]\!]$ and the numerator and denominator of every entry of $\overrightarrow{\tau'}$ can be described using a polynomial number of bits in $|\mathcal{N}|(\log\|M\| + \log\|M'\|)$. Notice that the smallest natural number k such that $k \cdot \overrightarrow{\tau'} \in \mathbb{N}^T$ is at most the least common multiple of the denominators of all the numbers in the vector $\overrightarrow{\tau'}$. Since the size of $[\![\tau']\!]$ is at most the number of transitions of $\mathcal{N}$, if we let w be the weight of the net $\mathcal{N}$, then it is easy to verify that the quantities $(w+1)^{[\![\tau']\!]}$, $\|\overrightarrow{\tau'}\|$ and k can all be described using a polynomial number of bits in $|\mathcal{N}|(\log\|M\| + \log\|M'\|)$. Applying Lemma 4.2 now finishes the proof. □

4.2. **The Insertion Lemma.** In the acyclic case, the existence of a cut-off is roughly characterized by the existence of solutions to the marking equation over $\mathbb{Q}_{\geq 0}^T$ and $\mathbb{Z}^T$. Intuitively, in the general case we replace the existence of solutions over $\mathbb{Q}_{\geq 0}^T$ by the conditions of the Scaling Lemma, and the existence of solutions over $\mathbb{Z}^T$ by the Insertion Lemma:

Lemma 4.7 (**Insertion Lemma**). *Let $k \in \mathbb{N}$ and let M, M', L, L' be markings of $\mathcal{N}$ satisfying $M \xrightarrow{\sigma} M'$ for some σ and $L' = L + \mathcal{A}\mathbf{y}$ for some $\mathbf{y} \in \mathbb{Z}^T$ such that $[\![\mathbf{y}]\!] \subseteq [\![\sigma]\!]$. Then $\mu M + kL \xrightarrow{*} \mu M' + kL'$ for $\mu = \alpha k + \alpha\|\overrightarrow{\sigma}\|nw + \|\mathbf{y}\|nw$, where $\alpha \in \mathbb{N}$ is the smallest number such that $\alpha\mathbf{1} + \mathbf{y} \geq \mathbf{0}$, w is the weight of $\mathcal{N}$ and n is the number of places in ${}^\bullet[\![\sigma]\!]$.*

Proof. We first provide an intuition behind the proof. In a first stage, we asynchronously execute multiple "copies" of the firing sequence σ from multiple "copies" of the marking M, until we reach a marking in which all places of ${}^{\bullet}[\![\sigma]\!]$ contain a sufficiently large number of tokens. At this point we temporarily interrupt the executions of the copies of σ to *insert* k firing sequences each with Parikh mapping $\alpha\overrightarrow{\sigma} + \mathbf{y}$. The net effect of this sequence is to transfer αk copies of M to M', leaving the other copies untouched, and exactly k copies of L to L'. In the third stage, we resume the interrupted executions of the copies of σ, which completes the transfer of the remaining copies of M to M'. We now proceed to the formal proof.

Let $\mathbf{x}$ be the Parikh image of σ, i.e., $\mathbf{x} = \overrightarrow{\sigma}$. Since $M \xrightarrow{\sigma} M'$, by the marking equation we have $M' = M + \mathcal{A}\mathbf{x}$.

First stage: Let $\lambda_x = \|x\|$, $\lambda_y = \|y\|$ and $\mu = \alpha k + \alpha\lambda_x nw + \lambda_y nw$. Let $\sigma := r_1, r_2, \ldots, r_j$ and let $M =: M_0 \xrightarrow{r_1} M_1 \xrightarrow{r_2} M_2 \ldots M_{j-1} \xrightarrow{r_j} M_j := M'$. Notice that for each place $p \in {}^{\bullet}[\![\sigma]\!]$, there exists a marking $M_{i_p} \in \{M_0, \ldots, M_{j-1}\}$ such that $M_{i_p}(p) > 0$.

Since each of the markings in $\{M_{i_p}\}_{p \in {}^{\bullet}[\![\sigma]\!]}$ can be obtained from M by firing a (suitable) prefix of σ, by the monotonicity property, it follows that starting from the marking $\mu M + kL = \alpha k M + kL + (\alpha\lambda_x nw + \lambda_y nw)M$, we can reach the marking L_0 given by $L_0 := \alpha k M + kL + \sum_{p \in {}^{\bullet}[\![\sigma]\!]} (\alpha\lambda_x w + \lambda_y w)M_{i_p}$. This completes our first stage.

Second stage - Insert: Since $[\![\mathbf{y}]\!] \subseteq [\![\sigma]\!]$, if $\mathbf{y}(t) \neq 0$ then $\mathbf{x}(t) \neq 0$. Since $\mathbf{x}(t) \geq 0$ for every transition, by the definition of α, it now follows that $(\alpha\mathbf{x} + \mathbf{y})(t) \geq 0$ for every transition t and $(\alpha\mathbf{x} + \mathbf{y})(t) > 0$ precisely for those transitions in $[\![\sigma]\!]$.

Let ξ be any firing sequence such that $\overrightarrow{\xi} = \alpha\mathbf{x} + \mathbf{y}$ and let $\texttt{Inter} := \sum_{p \in {}^{\bullet}[\![\sigma]\!]} (\alpha\lambda_x w + \lambda_y w)M_{i_p}$. Notice that for each place $p \in {}^{\bullet}[\![\sigma]\!]$, $\texttt{Inter}(p) \geq (\alpha\lambda_x + \lambda_y)w \geq \|(\alpha\mathbf{x} + \mathbf{y})\| \cdot w$. For each $1 \leq i \leq k$, set L_i to be the marking $\alpha(k-i)M + \alpha i M' + (k-i)L + iL' + \texttt{Inter}$. By Proposition 4.5 and the marking equation, we have that for each $i \geq 0$, $L_i \xrightarrow{\xi} L_{i+1}$. Hence, starting from L_0, we reach the marking $L_k = \alpha k M' + kL' + \sum_{p \in {}^{\bullet}[\![\sigma]\!]} (\alpha\lambda_x w + \lambda_y w)M_{i_p}$. This completes our second stage.

Third stage: Notice that for each place $p \in {}^{\bullet}[\![\sigma]\!]$, by construction of M_{i_p}, there is a firing sequence which takes the marking M_{i_p} to the marking M'. By the monotonicity property, it then follows that there is a firing sequence which takes the marking L_k to the marking $\alpha k M' + kL' + \sum_{p \in {}^{\bullet}[\![\sigma]\!]}(\alpha\lambda_x w + \lambda_y w)M' = \mu M' + kL'$. This completes our third stage and also completes the desired firing sequence from $\mu M + kL$ to $\mu M' + kL'$. ☐

5. The cut-off problem for general Petri nets

Let $(\mathcal{N}, M, M')$ be a net system with $\mathcal{N} = (P, T, Pre, Post)$, such that $\mathcal{A}$ is its incidence matrix. As in Section 3, we first characterize the Petri net systems that admit a cut-off, and then provide a polynomial time algorithm to decide the existence of a cut-off for Petri nets (and hence also for rendez-vous protocols).

5.1. **Characterizing systems with cut-offs.** We generalize the characterization of Theorem 3.3 for acyclic Petri net systems to general ones.

Theorem 5.1. *A Petri net system $(\mathcal{N}, M, M')$ admits a cut-off if and only if there is a continuous firing sequence σ such that $M \xrightarrow[\mathbb{Q}]{\sigma} M'$ and the marking equation has a solution $\mathbf{y} \in \mathbb{Z}^T$ such that $[\![\mathbf{y}]\!] \subseteq [\![\sigma]\!]$.*

Proof. ($\Rightarrow$): Assume $(\mathcal{N}, M, M')$ admits a cut-off. By Lemma 3.2, there exist $n \in \mathbb{N}$ and firing sequences τ, τ' such that $nM \xrightarrow{\tau} nM'$, $(n+1)M \xrightarrow{\tau'} (n+1)M'$ and $[\![\tau']\!] \subseteq [\![\tau]\!]$.

Let $\tau = t_1 t_2 \cdots t_k$ and let $M_0 := nM \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \ldots \xrightarrow{t_k} M_k := nM'$. It is then easy to verify that that $M_0/n \xrightarrow{t_1/n} M_1/n \xrightarrow{t_2/n} M_2/n \ldots \xrightarrow{t_k/n} M_k/n$. This means that if we set $\sigma := t_1/n,\ t_2/n,\ \ldots\ ,\ t_k/n$ then $M \xrightarrow[\mathbb{Q}]{\sigma} M'$. Further, by the marking equation we have $nM' = nM + \mathcal{A}\overrightarrow{\tau}$ and $(n+1)M' = (n+1)M + \mathcal{A}\overrightarrow{\tau'}$. Let $\mathbf{y} = \overrightarrow{\tau'} - \overrightarrow{\tau}$. Then $\mathbf{y} \in \mathbb{Z}^T$ and $M' = M + \mathcal{A}\mathbf{y}$. Since $[\![\tau']\!] \subseteq [\![\tau]\!] = [\![\sigma]\!]$, we have $[\![\mathbf{y}]\!] \subseteq [\![\sigma]\!]$.

($\Leftarrow$): Assume that there exists a continuous firing sequence σ and a vector $\mathbf{y}' \in \mathbb{Z}^T$ such that $[\![\mathbf{y}']\!] \subseteq [\![\sigma]\!]$, $M \xrightarrow[\mathbb{Q}]{\sigma} M'$ and $M' = M + \mathcal{A}\mathbf{y}'$. Let $s = |\mathcal{N}|(\log \|M\| + \log \|M'\|)$. We first claim that we can find a vector $\mathbf{y}$ such that $[\![\mathbf{y}]\!] \subseteq [\![\sigma]\!]$, $M' = M + \mathcal{A}\mathbf{y}$ and $\mathbf{y}$ can be described using a polynomial number of bits in s. Indeed, let $t_1, \ldots, t_k$ be the set of transitions not in $[\![\sigma]\!]$ and consider the system of equations given by $M' = M + \mathcal{A}\mathbf{v},\ \mathbf{v}_{t_1} = 0,\ \mathbf{v}_{t_2} = 0,\ \ldots,\ \mathbf{v}_{t_k} = 0$. We know that there is at least one integer solution to this system, namely $\mathbf{y}'$. It is well known that if a system of linear equations over the integers is feasible, then there is a solution which can be described using a number of bits which is polynomial in the size of the input (see e.g. [KM78]). Applying this result to our system of equations proves our claim.

Now, since $M \xrightarrow[\mathbb{Q}]{\sigma} M'$, by Lemma 4.6 there exists n, m (both of which can be described using a polynomial number of bits in s) and a firing sequence τ such that $[\![\tau]\!] = [\![\sigma]\!]$, $\|\overrightarrow{\tau}\| \leq m$ and $nM \xrightarrow{\tau} nM'$. Since $\mathbf{y}$ can be described by a polynomial number of bits in s, by Lemma 4.7, there exists μ (which can once again be described using a polynomial number of bits in s) such that $\mu n M + M \xrightarrow{*} \mu n M' + M'$. By Lemma 3.2 the system $(\mathcal{N}, M, M')$ admits a cut-off which can be described by a polynomial number of bits in s. □

Notice that we have actually proved that if a net system admits a cut-off then it admits a cut-off which is expressible by a polynomial number of bits in its size. Since the cut-off problem for a rendez-vous protocol $\mathcal{P}$ can be reduced to the cut-off problem for the Petri net system $(\mathcal{N}_{\mathcal{P}}, \wr init \int, \wr fin \int)$, it follows that:

Corollary 5.2. *If the system $(\mathcal{N}, M, M')$ admits a cut-off then it admits a cut-off which is expressible by a polynomial number of bits in $|\mathcal{N}|(\log \|M\| + \log \|M'\|)$. Hence, if a rendez-vous protocol $\mathcal{P}$ admits a cut-off then it admits a cut-off which is at most $2^{|\mathcal{P}|^{O(1)}}$.*

It is already known from [HS20, Section 6.3] that there are protocols whose smallest cut-off is at least exponential in the size of the protocol. Hence, the above corollary gives almost tight bounds on the smallest cut-off of a protocol.

Example 5.3. Let us consider the Petri net $\mathcal{N}_{\mathcal{P}}$ from Figure 2 given in Example 2.6. We have seen that 4 is the smallest cut-off for the system $(\mathcal{N}_{\mathcal{P}}, init, fin)$. We now show that the conditions of Theorem 5.1 are satisfied for this system. Note that if we set

$\sigma = t_1/8,\ t_2/4,\ t_3/2$ then $\wr init \wr \xrightarrow[\mathbb{Q}]{\sigma} \wr fin \wr$ is a valid run of this net. Further, the marking equation for this system is:

$$
\begin{aligned}
0 &= 1 - 2\mathbf{v}_{t_1} - \mathbf{v}_{t_2} - \mathbf{v}_{t_3} && \text{(Equation for the place } init\text{)} \\
0 &= 0 + 2\mathbf{v}_{t_1} - \mathbf{v}_{t_2} && \text{(Equation for the place } q_1\text{)} \\
1 &= 0 + 2\mathbf{v}_{t_2} + \mathbf{v}_{t_3} && \text{(Equation for the place } f\text{)}
\end{aligned}
$$

Notice that $\mathbf{y} = (1, 2, -3)$ is a solution to the marking equation such that $[\![\mathbf{y}]\!] \subseteq [\![\sigma]\!]$. Hence, the conditions of Theorem 5.1 are satisfied.

5.2. **Polynomial time algorithm.** We use the characterization given in the previous section to provide a polynomial time algorithm for the cut-off problem. The following lemma, which is very similar to Lemma 3.5, was proved in [FH15] and enables us to find a firing sequence between two markings with maximum support.

Lemma 5.4 [FH15, Lemma 12 and Proposition 26]**.** *Let FS be the set of all continuous firing sequences τ such that $M \xrightarrow[\mathbb{Q}]{\tau} M'$. Then, there exists a sequence $\sigma \in FS$ such that $[\![\tau]\!] \subseteq [\![\sigma]\!]$ for every $\tau \in FS$. Moreover, the support of such a sequence σ can be computed in polynomial time.*

We now have all the ingredients to prove the existence of a polynomial time algorithm.

Theorem 5.5. *The cut-off problem for Petri nets can be solved in polynomial time.*

Proof. The proof is exactly the same as the proof of Theorem 3.6, except that instead of checking if the marking equation over $\mathbb{Q}_{\geq 0}$ is feasible, we first check if M can reach M' over the continuous semantics and if so, obtain the maximum support of all the firing sequences τ such that $M \xrightarrow[\mathbb{Q}]{\tau} M'$ and then use this maximum support to construct additional constraints for the marking equation over the integers. □

5.3. **P-hardness.** We now have the following lemma, which enables us to derive a P-completeness result for the cut-off problem.

Lemma 5.6. *The cut-off problem for rendez-vous protocols is* P*-hard.*

Proof. We give a logspace reduction from the *Circuit Value Problem (CVP)* which is known to be P-hard [Lad75].

CVP is defined as follows: We are given a Boolean circuit C with n input variables $x_1, \ldots, x_n$ and m gates $g_1, \ldots, g_m$. We are also given an assignment v for the input variables and a gate *out*. We have to check if the output of the gate *out* is 1, when the input variables are assigned values according to the assignment v.

We represent each binary gate g as a tuple $(\circ, s_1, s_2)$ where $\circ \in \{\vee, \wedge\}$ denotes the operation of g and $s_1, s_2 \in \{x_1, \ldots, x_n, g_1, \ldots, g_m\}$ are the inputs to g. In a similar fashion each unary gate g is represented as a tuple $(\neg, s)$.

Let $v(x_i) \in \{0, 1\}$ denote the value assigned to the variable x_i by the assignment v. Similarly, let $v(g_i)$ denote the output of the gate g_i when the input variables are assigned values according to the assignment v. Hence, the problem is to determine if $v(out)$ is 1.

The reduction. We now define a rendez-vous protocol as follows: Our alphabet Σ will be $\{a, z\} \cup \{(g, b_1, b_2) : g$ is a binary gate and $b_1, b_2 \in \{0, 1\}\} \cup \{(g, b) : g$ is an unary gate and $b \in \{0, 1\}\}$. We will have an initial state *init*, $2n$ states $q^0_{x_1}, q^1_{x_1}, \dots, q^0_{x_n}, q^1_{x_n}$ and $2m$ states $q^0_{g_1}, q^1_{g_1}, \dots, q^0_{g_m}, q^1_{g_m}$. The rules are defined as follows:

- For each $1 \le i \le n$, we have the rules $(init, !a, q^{v(x_i)}_{x_i})$ and $(init, ?a, q^{v(x_i)}_{x_i})$. These two rules correspond to setting the value of x_i to $v(x_i)$.
- For each binary gate $g = (\circ, s_1, s_2)$ and for each $b_1, b_2 \in \{0, 1\}$, we have the rules $(q^{b_1}_{s_1}, !(g, b_1, b_2), q^{b_1 \circ b_2}_g)$ and $(q^{b_2}_{s_2}, ?(g, b_1, b_2), q^{b_1 \circ b_2}_g)$. These rules say that if the output of s_1 is b_1 and the output of s_2 is b_2, then the output of g is $b_1 \circ b_2$.
- For each unary gate $g = (\neg, s)$ and each $b \in \{0, 1\}$, we have the rules $(q^b_s, !(g, b), q^{\neg b}_g)$ and $(q^b_s, ?(g, b), q^{\neg b}_g)$. These rules guarantee that the output of g is the negation of the output of s.
- Finally we have the rule $(q^1_{out}, !z, q^1_{out})$ and also the rule $(q, ?z, q^1_{out})$ for every state q. These rules guarantee that once a process reaches the state q^1_{out}, it can make all the other processes reach q^1_{out} as well.

We then set our final state *fin* to be q^1_{out}. Notice that this construction can be performed using a logarithmic amount of space. Constructing the states and the letters of the alphabet can be accomplished by iterating over all the input variables and the gates of the circuit which can be performed using a logarithmic amount of space on the work-tape. Constructing each rule requires a constant number of pointers to the input, which can also be maintained by using a logarithmic amount of space. Hence, the entire reduction can be carried out by using only a logarithmic amount of space on the work-tape.

Some preliminary observations. Before we proceed to the correctness of the construction, we set up some notation and make some preliminary observations. For a state q and an initial configuration C^k_{init}, we say that C^k_{init} can *cover* q if there exists a run $C^k_{init} \xRightarrow{*} D$ such that $D(q) > 0$. We say that q is coverable if it can be covered from some initial configuration C^k_{init}.

Remark 5.7. In our construction the final state *fin* is taken to be q^1_{out}. Notice that we have the rule $(fin, !z, fin)$ in our protocol and also the rules $(q, ?z, fin)$ for every state q. By using this collection of rules, it is easy to see that if D is a configuration such that $D(fin) > 0$ and $|D| = k$, then D can reach the configuration C^k_{fin}. This implies that C^k_{init} can cover *fin* if and only if C^k_{init} can reach C^k_{fin}.

Remark 5.8. By definition of coverability, notice that if C^k_{init} can cover a state q then C^l_{init} can also cover q for any $l \ge k$. By the previous remark this means that there exists some B such that C^B_{init} can cover *fin* if and only if there exists some B such that for all $k \ge B$, C^k_{init} can reach C^k_{fin}.

By these two remarks, to prove that the reduction is correct it suffices to prove the following statement: $v(out) = 1$ if and only if *fin* is coverable.

Proof of correctness. We prove a stronger statement than what is required. We claim that

> For any $h \in \{x_1, \dots, x_n, g_1, \dots, g_m\}$, $v(h) = b$ if and only if the state q^b_h is coverable.

($\Rightarrow$): Let $h_1, \dots, h_{n+m}$ be a topological ordering of the underlying DAG of the circuit C. We will prove by induction on this ordering that if $v(h) = b$ for some gate h, then q_h^b is coverable. For the base case of $h = h_1$, notice that since h_1 has no predecessors, it must be some input gate x_i. Let $v(x_i) = b$. By definition of the rules $(\mathit{init}, !a, q_{x_i}^b)$ and $(\mathit{init}, ?a, q_{x_i}^b)$, it follows that from C_{init}^2 we can cover $q_{x_i}^b$. For the induction step, suppose for some $i > 1$, we have already proved the claim for all $h \in \{h_1, \dots, h_{i-1}\}$. Let $v(h_i) = b_i$. There are now multiple cases:

- Suppose h_i is an input gate. Then by the same argument which was given for the base case, we can show that $q_{h_i}^{b_i}$ is coverable.
- Suppose h_i is an unary gate of the form $(\neg, h_j)$ where $j < i$. Let $v(h_j) = b_j$. We then have $v(h_i) = b_i = \neg b_j$. By induction hypothesis, $q_{h_j}^{b_j}$ is coverable and so there is some k such that C_{init}^k can cover $q_{h_j}^{b_j}$. Hence, this means that from C_{init}^{2k} we can reach a configuration D such that $D(q_{h_j}^{b_j}) \geq 2$. From D, by using the rules $(q_{h_j}^{b_j}, !(h_i, b_j), q_{h_i}^{\neg b_j})$ and $(q_{h_j}^{b_j}, ?(h_i, b_j), q_{h_i}^{\neg b_j})$, we can now cover $q_{h_i}^{\neg b_j} = q_{h_i}^{b_i}$.
- Suppose h_i is a binary gate of the form $(\circ, h_j, h_k)$ where $j, k < i$. Let $v(h_j) = b_j$ and $v(h_k) = b_k$. We then have $v(h_i) = b_i = b_j \circ b_k$. By induction hypothesis, $q_{h_j}^{b_j}$ and $q_{h_k}^{b_k}$ are coverable and so there exist some ℓ and ℓ' such that C_{init}^{ℓ} can cover $q_{h_j}^{b_j}$ and $C_{\mathit{init}}^{\ell'}$ can cover $q_{h_k}^{b_k}$. Hence, this means that from $C_{\mathit{init}}^{\ell+\ell'}$ we can reach a configuration D such that $D(q_{h_j}^{b_j}) \geq 1$ and $D(q_{h_k}^{b_k}) \geq 1$. From D, by using the rules $(q_{h_j}^{b_j}, !(h_i, b_j, b_k), q_{h_i}^{b_j \circ b_k})$ and $(q_{h_k}^{b_k}, ?(h_i, b_j, b_k), q_{h_i}^{b_j \circ b_k})$, we can now cover $q_{h_i}^{b_j \circ b_k} = q_{h_i}^{b_i}$.

($\Leftarrow$): We now show that if for some h, there is a k such that $C_{\mathit{init}}^k \xRightarrow{*} D$ is a run satisfying $D(q_h^b) > 0$, then $v(h) = b$. We do this by induction on the length of the run from C_{init}^k to D. For the base case of a single step given by $C_{\mathit{init}}^k \xRightarrow{r,r'} D$, notice that since we start from a configuration where all the agents are in the state init, it must be the case that $r = (\mathit{init}, !a, q_{x_i}^{v(x_i)})$ and $r' = (\mathit{init}, ?a, q_{x_j}^{v(x_j)})$ for some input variables x_i and x_j. Hence, h can only be either x_i or x_j and in both of these cases, the claim is true.

For the induction step, assume that we have proven the claim for all runs of length at most i and suppose for some ℓ, we have a run from C_{init}^{ℓ} to D of length $i+1$ satisfying $D(q_h^b) > 0$. Let $C_{\mathit{init}}^{\ell} \xRightarrow{*} D' \xRightarrow{r,r'} D$. If $D'(q_h^b) > 0$, then by induction hypothesis we are already done. Otherwise, by construction of the protocol $\mathcal{P}$, one of the following cases must hold:

- $r = (\mathit{init}, !a, q_{x_i}^{v(x_i)})$ and $r' = (\mathit{init}, ?a, q_{x_j}^{v(x_j)})$ for some input variables x_i and x_j: This case is similar to the base case.
- $r = (q_s^b, !(h, b), q_h^{\neg b})$ and $r' = (q_s^b, ?(h, b), q_h^{\neg b})$ where $b \in \{0, 1\}$ and h is an unary gate of the form $h = (\neg, s)$. Hence $D'(q_s^b) > 0$ and so by induction hypothesis, we have that $v(s) = b$. Hence $v(h) = \neg v(s) = \neg b$.
- $r = (q_{s_1}^{b_1}, !(h, b_1, b_2), q_h^b)$ and $r' = (q_{s_2}^{b_2}, ?(h, b_1, b_2), q_h^b)$ where $b_1, b_2 \in \{0, 1\}$, h is a binary gate of the form $h = (\circ, s_1, s_2)$ and $b = b_1 \circ b_2$. Hence $D'(q_{s_1}^{b_1}) > 0$ and $D'(q_{s_2}^{b_2}) > 0$ and so by induction hypothesis, we have that $v(s_1) = b_1$ and $v(s_2) = b_2$. Hence $v(h) = v(s_1) \circ v(s_2) = b_1 \circ b_2 = b$.

- $r = (q^1_{out}, !z, q^1_{out})$ and $r' = (q, ?z, q^1_{out})$ for some state q. In this case, notice that h must be q_{out} and b must be 1. By construction of the run, it must be the case that $D'(q^1_{out}) > 0$ and so by induction hypothesis we are already done.

Hence, the induction step is complete and we have proved our claim, which also completes the proof of correctness of the reduction. □

Since rendez-vous protocols are a special case of Petri nets, this also proves that the cut-off problem for Petri nets is P-hard. Therefore we get:

Theorem 5.9. *The cut-off problems for Petri nets and rendez-vous protocols are* P*-complete.*

6. The bounded-loss cut-off problem for rendez-vous protocols

In this section, we consider the following *bounded-loss cut-off problem for rendez-vous protocols*, which was suggested as a variant of the cut-off problem in Section 7 of [HS20].

Given: A rendez-vous protocol $\mathcal{P} = (Q, \Sigma, init, fin, R)$
Decide: Is there $B \in \mathbb{N}$ such that for every $n \in \mathbb{N}$, there is a configuration D_n with $C^n_{init} \xRightarrow{*} D_n$ and $D_n(fin) \geq n - B$?

If such a B exists, then we say that the protocol $\mathcal{P}$ has the bounded-loss cut-off property and that B is a bounded-loss cut-off for $\mathcal{P}$.

Intuitively, the cut-off problem asks if for all large enough population sizes n, we can move n agents from the state *init* to the state *fin*. The bounded-loss cut-off problem asks if there is a bound B such that for all population sizes n, we can move $n - B$ agents from the state *init* to the state *fin* and leave the remaining B agents in any state of the protocol $\mathcal{P}$. Intuitively, we are allowed to "leave out" a bounded number of agents while moving everybody else to the final state.

Example 6.1. Let us consider the protocol $\mathcal{P}$ from Figure 1 given in Example 2.2. We have seen that 4 is a cut-off for $\mathcal{P}$. Let us modify $\mathcal{P}$ so that we get the protocol in Figure 4.

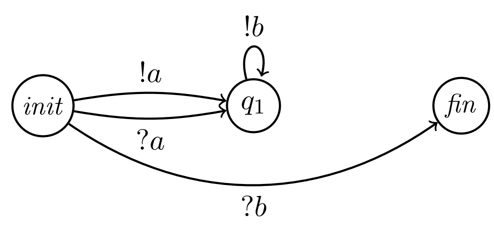

Figure 4: A modification of the protocol from Figure 1

We first observe that this protocol does not admit a cut-off. Indeed, the only possible step from any initial configuration consists of moving two agents from the state *init* to q_1 by using the message a. However, since there are no outgoing rules from q_1, it follows that these two agents can never leave q_1 from here on. Hence, this protocol does not admit a cut-off. But 2 is a bounded-loss cut-off for this protocol because once we move 2 agents from *init* to q_1, we can move all the remaining agents from *init* to *fin* by using the rules $(q_1, !b, q_1)$ and $(init, ?b, fin)$.

Our main result in this section is that

Theorem 6.2. *The bounded-loss cut-off problem is* P*-complete.*

We prove this by adapting the techniques developed for the cut-off problem. Similar to the cut-off problem, we first give a characterization of protocols having the bounded-loss cut-off property and then use this characterization to give a polynomial-time algorithm for deciding the bounded-loss cut-off problem.

6.1. **Characterization of protocols having a bounded-loss cut-off.** For the rest of this section, we fix a rendez-vous protocol $\mathcal{P} = (Q, \Sigma, init, fin, R)$. We need a couple of notations to state the required characterization. Given the protocol $\mathcal{P}$, we consider the Petri net $\mathcal{N}_\mathcal{P} = (P, T, Pre, Post)$ with incidence matrix $\mathcal{A} = Post - Pre$ that we constructed in Section 2, and then consider the associated Petri net system $(\mathcal{N}_\mathcal{P}, \lbag init \rbag, \lbag fin \rbag)$. In this Petri net, we say that $\lbag init \rbag$ can *cover* $\lbag fin \rbag$ by a continuous firing sequence σ if there exists a marking M such that $\lbag init \rbag \xrightarrow[\mathbb{Q}]{\sigma} M$ and $M(fin) > 0$. We now prove that

Theorem 6.3. *The protocol $\mathcal{P}$ has a bounded-loss cut-off if and only if in the Petri net system $(\mathcal{N}_\mathcal{P}, \lbag init \rbag, \lbag fin \rbag)$, $\lbag init \rbag$ can cover $\lbag fin \rbag$ using a continuous firing sequence σ and the marking equation has a solution $\mathbf{y} \in \mathbb{Q}^T_{\geq 0}$ such that $[\![\mathbf{y}]\!] \subseteq [\![\sigma]\!]$.*

Proof. ($\Rightarrow$): Assume $\mathcal{P}$ has the bounded-loss cut-off property. By Proposition 2.5, there must exist $B \in \mathbb{N}$ such that for all $n \in \mathbb{N}$, there exists D_n with $C^n_{init} \xrightarrow{*} D_n$ in the Petri net $\mathcal{N}_\mathcal{P}$ and $D_n(fin) \geq n - B$.

Consider the infinite sequence of markings $D_1, D_2, \dots$. Notice that in all of these markings, at most B tokens are not in the place fin. This means that there must exist a subsequence $D_{i_1}, D_{i_2}, \dots$ with $B < i_1 < i_2 < \dots$ and a marking D of size $B' \leq B$ such that each $D_{i_j} = D + \lbag (i_j - B') \cdot fin \rbag$.

Now, consider the sequence of runs $C^{i_1}_{init} \xrightarrow{\sigma_{i_1}} D_{i_1}, C^{i_2}_{init} \xrightarrow{\sigma_{i_2}} D_{i_2}, \dots$ and consider the corresponding Parikh vectors $\overrightarrow{\sigma_{i_1}}, \overrightarrow{\sigma_{i_2}}, \dots$. By Dickson's lemma, there must exist indices $k < l$ such that $\overrightarrow{\sigma_{i_k}} \leq \overrightarrow{\sigma_{i_l}}$. Let $\sigma_{i_l} = t_1 t_2 \cdots t_m$. Construct the continuous firing sequence $\sigma := t_1/i_l, t_2/i_l, \cdots, t_m/i_l$. From the fact that $C^{i_l}_{init} \xrightarrow{\sigma_{i_l}} D_{i_l}$, we can easily conclude by induction on m that $\lbag init \rbag \xrightarrow[\mathbb{Q}]{\sigma} D_{i_l}/i_l$. Since $D_{i_l}(fin) > 0$, this means that $\lbag init \rbag$ can cover $\lbag fin \rbag$ using σ.

Further, by the marking equation we have $D + \lbag (i_k - B') \cdot fin \rbag = C^{i_k}_{init} + \mathcal{A} \cdot \overrightarrow{\sigma_{i_k}}$ and $D + \lbag (i_l - B') \cdot fin \rbag = C^{i_l}_{init} + \mathcal{A} \cdot \overrightarrow{\sigma_{i_l}}$. Setting $\mathbf{y} = (\overrightarrow{\sigma_{i_l}} - \overrightarrow{\sigma_{i_k}})/(i_l - i_k)$ gives us that $\lbag fin \rbag = \lbag init \rbag + \mathcal{A}\mathbf{y}$. By assumption on k and l, it follows that $\mathbf{y} \in \mathbb{Q}^T_{\geq 0}$ and $[\![\mathbf{y}]\!] \subseteq [\![\sigma_{i_l}]\!] = [\![\sigma]\!]$.

($\Leftarrow$): Assume that there exists a continuous firing sequence σ and a vector $\mathbf{y} \in \mathbb{Q}^T_{\geq 0}$ such that $[\![\mathbf{y}]\!] \subseteq [\![\sigma]\!]$, $\lbag init \rbag$ can cover $\lbag fin \rbag$ using σ and $\lbag fin \rbag = \lbag init \rbag + \mathcal{A}\mathbf{y}$.

Let $\lbag init \rbag \xrightarrow[\mathbb{Q}]{\sigma} M''$. By the Scaling lemma (Lemma 4.6), there exists some $n \in \mathbb{N}$ and some firing sequence τ such that $\lbag n \cdot init \rbag \xrightarrow{\tau} n \cdot M''$ and $[\![\tau]\!] = [\![\sigma]\!]$. Further, since $\mathbf{y} \in \mathbb{Q}^T_{\geq 0}$, it follows that there exists a $k \in \mathbb{N}$ such that $k\mathbf{y} \in \mathbb{N}^T$ and so the smallest number $\alpha \in \mathbb{N}$ such that $\alpha \mathbf{1} + k\mathbf{y} \geq \mathbf{0}$ is in fact 0.

Let $M = \lbag n \cdot init \rbag$ and $M' = n \cdot M''$. To summarize, we have that $M \xrightarrow{\tau} M'$ and $\lbag k \cdot fin \rbag = \lbag k \cdot init \rbag + \mathcal{A}(k\mathbf{y})$ where $[\![k\mathbf{y}]\!] = [\![\mathbf{y}]\!] \subseteq [\![\sigma]\!] = [\![\tau]\!]$. Let $\mu = \|\mathbf{y}\| n' w$, where n' is the number of places in ${}^\bullet[\![\tau]\!]$ and w is the weight of $\mathcal{N}_\mathcal{P}$. By the Insertion lemma (Lemma 4.7), it follows that for any $s \in \mathbb{N}$, $\mu M + \Lbag sk \cdot init \Rbag \xrightarrow{*} \mu M' + \Lbag sk \cdot fin \Rbag$. Unpacking the definition of M, it follows that for any $s \in \mathbb{N}$,

$$\Lbag (\mu n + sk) \cdot init \Rbag \xrightarrow{*} \mu M' + \Lbag sk \cdot fin \Rbag \tag{6.1}$$

We now claim that $B := \mu n + k$ is a bounded-loss cut-off for $\mathcal{P}$. To prove this, we have to show that for any $a \in \mathbb{N}$, there exists D_a with $C^a_{init} \xrightarrow{*} D_a$ and $D_a(fin) \geq a - B$. Suppose $a \leq \mu n + k$. Then we simply set D_a to be C^a_{init}. On the other hand, suppose $a > \mu n + k$. Let $b = a - \mu n - k$. Write b as $b = qk + r$ for some $q \in \mathbb{N}$ and some r such that $0 \leq r \leq k - 1$. By equation (6.1) and the monotonicity property, it follows that $C^a_{init} = \Lbag (\mu n + (q+1)k + r) \cdot init \Rbag \xrightarrow{*} \mu M' + \Lbag (q+1)k \cdot fin \Rbag + \Lbag r \cdot init \Rbag$. Notice that $(q+1)k \geq b = a - B$. Hence, we can set D_a to be $\mu M' + \Lbag (q+1)k \cdot fin \Rbag + \Lbag r \cdot init \Rbag$, which completes the proof. □

Note that this characterization is very similar to the characterization for the cut-off property, where the reachability condition is replaced with a coverability condition and the condition of the marking equation having a solution over the integers is replaced with it having a solution over the non-negative rationals.

Example 6.4. Let us consider the protocol from Figure 4. We have seen that 2 is a bounded-loss cut-off for this protocol. The Petri net corresponding to this protocol is given in Figure 5.

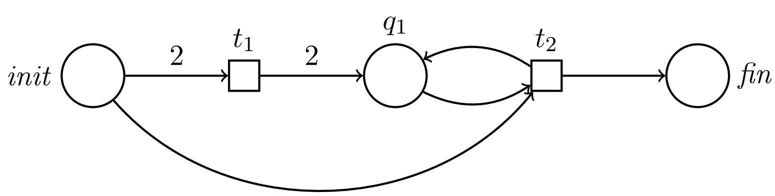

Figure 5: Petri net corresponding to the protocol from Figure 4

We now show that the conditions of Theorem 6.3 are satisfied here. Indeed, if we set $\sigma = t_1/4,\ t_2/2$ then $\Lbag init \Rbag$ can cover $\Lbag fin \Rbag$ using σ. Further, the marking equation for the markings $\Lbag init \Rbag$ and $\Lbag fin \Rbag$ is given by,

$$\begin{aligned} 0 &= 1 - 2\mathbf{v}_{t_1} - \mathbf{v}_{t_2} && \text{(Equation for the place } init\text{)} \\ 0 &= 0 + 2\mathbf{v}_{t_1} && \text{(Equation for the place } q_1\text{)} \\ 1 &= 0 + \mathbf{v}_{t_2} && \text{(Equation for the place } f\text{)} \end{aligned}$$

Notice that $\mathbf{y} = (0, 1)$ is a solution to the marking equation such that $[\![\mathbf{y}]\!] \subseteq [\![\sigma]\!]$. Hence, the conditions of Theorem 6.3 are satisfied.

6.2. **Polynomial time algorithm.** We use the characterization given in the previous subsection to provide a polynomial time algorithm for the bounded-loss cut-off problem. Similarly to Lemma 5.4, we have the following lemma which could be inferred from the results of [FH15].

Lemma 6.5 [FH15, Lemma 12 and Proposition 29]**.** *Let FS be the set of all continuous firing sequences τ such that $\wr init \wr$ can cover $\wr fin \wr$. Then, there exists a sequence $\sigma \in FS$ such that $[\![\tau]\!] \subseteq [\![\sigma]\!]$ for every $\tau \in FS$. Moreover, the support of such a sequence σ can be computed in polynomial time.*

We now have all the ingredients to prove the existence of a polynomial time algorithm.

Theorem 6.6. *The bounded-loss cut-off problem can be solved in polynomial time.*

Proof. First, we check that $\wr init \wr$ can cover $\wr fin \wr$ over the continuous semantics, which can be done in polynomial time [FH15, Proposition 29]. If this is not true, then by Theorem 6.3, we can immediately reject. Otherwise, using Lemma 6.5, in polynomial time we compute the maximum support U of all the firing sequences τ such that $\wr init \wr$ can cover $\wr fin \wr$ using τ. We now check in polynomial time, if the marking equation over $(\mathcal{N}_{\mathcal{P}}, \wr init \wr, \wr fin \wr)$ has a solution $\mathbf{x}$ over the non-negative rationals such that $\mathbf{x}_t = 0$ for any $t \notin U$. By Theorem 6.3 such a solution exists if and only if the protocol admits a bounded-loss cut-off. □

6.3. **P-Hardness.** We now have the following lemma, which enables us to derive a P-completeness result for the bounded-loss cut-off problem.

Lemma 6.7. *The bounded-loss cut-off problem is* P*-hard.*

Proof. Similar to the P-hardness proof of the cut-off problem, we reduce from CVP. Let C be a Boolean circuit and let v be an assignment to the input variables of C. We consider the same rendez-vous protocol $\mathcal{P}$ that we constructed in the reduction for the cut-off problem in Lemma 5.6. We claim that

$\mathcal{P}$ has a cut-off if and only if $\mathcal{P}$ has a bounded-loss cut-off.

Note that if this claim is true, then by the reduction in Lemma 5.6, this would immediately imply P-hardness for the bounded-loss cut-off problem. We now proceed to prove this claim.

Suppose B is a cut-off for $\mathcal{P}$. This means that for every $n \geq B$, $C^n_{init} \xrightarrow{*} C^n_{fin}$. For $n < B$, let D_n be C^n_{init} and for $n \geq B$, let D_n be C^n_{fin}. By definition of B, it follows that for every n, C^n_{init} can reach D_n and $D_n(fin) \geq n - B$. Hence, B is also a bounded-loss cut-off for $\mathcal{P}$.

Suppose B is a bounded-loss cut-off for $\mathcal{P}$. Hence, for every n, there exists a marking D_n such that C^n_{init} can reach D_n and $D_n(fin) \geq n - B$. In particular for any $n > B$, it follows that C^n_{init} can reach a marking D_n such that $D_n(fin) > 0$. By Remark 5.7, it follows that C^n_{init} can reach C^n_{fin}. Hence, for every $n > B$, C^n_{init} can reach C^n_{fin} and so $B + 1$ is a cut-off for $\mathcal{P}$. □

7. The cut-off and bounded-loss cut-off problems for symmetric rendez-vous protocols

In [HS20], Horn and Sangnier introduced symmetric rendez-vous protocols, where sending and receiving a message at each state has the same effect, and showed that the cut-off problem for this class of protocols is in NP. We improve on their result and show that we can decide this problem in NC. We now formally define symmetric protocols.

Definition 7.1. A rendez-vous protocol $\mathcal{P} = (Q, \Sigma, \mathit{init}, \mathit{fin}, R)$ is *symmetric* if its set of rules is symmetric under swapping $!a$ and $?a$ for every $a \in \Sigma$, i.e., for every $a \in \Sigma$, we have that $(q, !a, q') \in R$ if and only if $(q, ?a, q') \in R$.

Remark 7.2. Since $(q, !a, q') \in R$ if and only if $(q, ?a, q') \in R$ for a symmetric protocol $\mathcal{P}$, in the following, we will simply denote rules of a symmetric protocol as a tuple in $Q \times \Sigma \times Q$, with the understanding that (q, a, q') denotes that there are two rules of the form $(q, !a, q')$ and $(q, ?a, q')$ in the protocol.

Example 7.3. Let us consider the symmetric protocol in Figure 6, where the alphabet Σ is taken to be $\{a, b, c\}$.

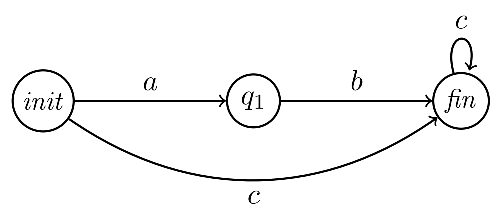

Figure 6: A symmetric protocol

Notice that 2 is a cut-off for this protocol. Indeed, starting from any initial configuration of size at least 2, we can first move 2 agents from *init* to *fin* by making them pass through q_1. Once we have put these two agents in the *fin* state, we can move all the remaining agents from the *init* state to the *fin* state by means of the rules $(\mathit{fin}, !c, \mathit{fin})$ and $(\mathit{init}, ?c, \mathit{fin})$.

7.1. **Characterization of symmetric protocols admitting a cut-off.** Let us fix a symmetric protocol $\mathcal{P} = (Q, \Sigma, \mathit{init}, \mathit{fin}, R)$ for the rest of this section. Horn and Sangnier proved the following nice characterization of symmetric protocols that admit a cut-off.

Proposition 7.4 [HS20, Lemma 18]**.** *The protocol $\mathcal{P}$ admits a cut-off if and only if there exists an even number e and an odd number o such that C^e_{init} can reach C^e_{fin} and C^o_{init} can reach C^o_{fin}.*

We will now translate this characterization into one that is more amenable to algorithmic analysis. To begin with, we use the symmetric protocol $\mathcal{P}$ to define a graph $\mathcal{G}_\mathcal{P}$ whose vertices are the states of $\mathcal{P}$ and there is an edge between q and q' in $\mathcal{G}_\mathcal{P}$ if and only if there exists $a \in \Sigma$ such that $(q, a, q') \in R$. The following lemma is immediate from the definition of $\mathcal{P}$.

Lemma 7.5. *There exists $k \in \mathbb{N}$ such that C^{2k}_{init} can reach C^{2k}_{fin} in $\mathcal{P}$ if and only if there is a path from init to fin in the graph $\mathcal{G}_\mathcal{P}$.*

Proof. The left-to-right implication follows from the definition of $\mathcal{G}_\mathcal{P}$. For the other direction, suppose there is a path $\mathit{init}, q_1, q_2, \dots, q_{m-1}, \mathit{fin}$ in the graph $\mathcal{G}_\mathcal{P}$. Then notice that $\wr 2 \cdot \mathit{init} \wr \Rightarrow \wr 2 \cdot q_1 \wr \Rightarrow \wr 2 \cdot q_2 \wr \Rightarrow \cdots \Rightarrow \wr 2 \cdot q_{m-1} \wr \Rightarrow \wr 2 \cdot \mathit{fin} \wr$ is a valid run of $\mathcal{P}$. $\square$

Intuitively, the above lemma takes care of the "even" case in the characterization given in Proposition 7.4. To handle the "odd" case, we first need a couple of definitions.

A state q of $\mathcal{P}$ will be called *good* if there is a path from *init* to q and a path from q to *fin* in the graph $\mathcal{G}_\mathcal{P}$. A state which is not good is *bad*. Given the protocol $\mathcal{P}$, we consider the Petri net $\mathcal{N}_\mathcal{P} = (Q, T, Pre, Post)$ with incidence matrix $\mathcal{A} = Post - Pre$ that we constructed in Section 2. Note that the set of places of $\mathcal{N}_\mathcal{P}$ is the set of states of $\mathcal{P}$. A transition t of $\mathcal{N}_\mathcal{P}$ is called *useless* if ${}^\bullet t \cup t^\bullet$ contains a bad state. We now have the following propositions about good states.

Proposition 7.6. *If q is good, then $⦃2 \cdot init⦄ \xrightarrow{*} ⦃2 \cdot q⦄$ and $⦃2 \cdot q⦄ \xrightarrow{*} ⦃2 \cdot fin⦄$ in $\mathcal{N}_\mathcal{P}$.*

Proof. Since q is good, there are paths $init, p_1, \ldots, p_n, q$ and $q, q_1, \ldots, q_m, fin$ in $\mathcal{G}_\mathcal{P}$. By definition of symmetric protocols, it follows that $⦃2 \cdot init⦄ \Rightarrow ⦃2 \cdot p_1⦄ \Rightarrow \cdots \Rightarrow ⦃2 \cdot q⦄$ and $⦃2 \cdot q⦄ \Rightarrow ⦃2 \cdot q_1⦄ \Rightarrow \cdots \Rightarrow ⦃2 \cdot fin⦄$ are valid runs of $\mathcal{P}$. By Proposition 2.5, these two runs are also valid in the Petri net $\mathcal{N}_\mathcal{P}$. □

The next proposition intuitively asserts that in any run from an initial configuration to a final configuration, only good states may occur.

Proposition 7.7. *Suppose C is such that $C^n_{init} \xrightarrow{*} C$ (resp. $C \xrightarrow{*} C^n_{fin}$) for some n. Then for every q such that $C(q) > 0$, there is a path from init to q (resp. from q to fin) in the graph $\mathcal{G}_\mathcal{P}$.*

Proof. In both of these cases, we can prove the claim by induction on the length of the underlying run. □

The following lemma now allows us to handle the "odd" case in the characterization of symmetric protocols that admit a cut-off.

Lemma 7.8. *There exists $k \in \mathbb{N}$ such that $C^{2k+1}_{init} \xrightarrow{*} C^{2k+1}_{fin}$ if and only if the marking equation for $(\mathcal{N}_\mathcal{P}, ⦃init⦄, ⦃fin⦄)$ has a solution $\mathbf{x}$ over the field $\mathbb{F}_2$ such that $\mathbf{x}[t] = 0$ for every useless transition t.*

Proof. We first provide an intuition behind the proof. The left-to-right implication is true because we can perform a "modulo 2" operation on both sides of the marking equation. For the other direction, we use an idea similar to the Insertion Lemma (Lemma 4.7). Let $\mathbf{x}$ be a solution to the marking equation over $\mathbb{F}_2$ such that $\mathbf{x}[t] = 0$ for every useless transition t. Using Proposition 7.6, we first populate all the good states of Q with enough agents such that all the good states except *init* have an even number of agents. Then, we fire exactly once all the transitions t such that $\mathbf{x}[t] = 1$. Since $\mathbf{x}$ satisfies the marking equation over $\mathbb{F}_2$, we can now argue that in the resulting configuration, the number of agents at each bad state is 0 and the number of agents in each good state except *fin* is even. Hence, we can once again use Proposition 7.6 to conclude that we can move all the agents which are not at *fin* to the final state *fin*. We now proceed to the formal proof.

($\Rightarrow$): Suppose there exists $k \in \mathbb{N}$ and a firing sequence σ such that $C^{2k+1}_{init} \xrightarrow{\sigma} C^{2k+1}_{fin}$. We claim that $\overrightarrow{\sigma}[t] = 0$ for every useless transition t. For the sake of contradiction, suppose $\overrightarrow{\sigma}[t] > 0$ for some useless transition t. Let $\sigma = \sigma' t \sigma''$ and let $C^{2k+1}_{init} \xrightarrow{\sigma'} C \xrightarrow{t} C' \xrightarrow{\sigma''} C^{2k+1}_{fin}$. By definition of a useless transition, it follows that there is a bad state q such that either $C(q) > 0$ or $C'(q) > 0$. However, this is a direct contradiction to Proposition 7.7. Hence, $\overrightarrow{\sigma}[t] = 0$ for every useless transition t.

By the marking equation it follows that $C_{fin}^{2k+1} = C_{init}^{2k+1} + \mathcal{A}\overrightarrow{\sigma}$. Taking modulo 2 on both sides of this equation, we have that $\overrightarrow{\sigma}$ mod 2 is a solution to the marking equation over the field $\mathbb{F}_2$.

($\Leftarrow$): Suppose the marking equation has a solution $\mathbf{x}$ over $\mathbb{F}_2$ such that $\mathbf{x}[t] = 0$ for every useless transition t. Let T' be the set of all transitions t such that $\mathbf{x}[t] = 1$. Let $n = 2|Q| + 3$. We will now construct a run from C_{init}^n to C_{fin}^n as follows:

First stage - Saturate: Let $G \subseteq Q$ be the set of good states and let $\ell = 2|Q| - 2|G|$. By Proposition 7.6 and the monotonicity property, from C_{init}^n we can reach the marking `First` $:= \wr (2l+3) \cdot init \wr + \sum_{q \in G} \wr 2q \wr$.

Second stage - Insert: Since $\mathbf{x}[t] = 0$ for every useless transition t, it follows that for every transition $t \in T'$, ${}^\bullet t \cup t^\bullet \subseteq G$. Hence, every transition $t \in T'$ is enabled at `First`. Therefore, we can fire all these transitions exactly once, in any order, from `First` to reach some marking `Second`. Since we fire only transitions from T', it follows that `Second`$(q) = 0$ for any bad state q.

We now claim that `Second`(q) is even for every $q \neq fin$ and `Second`(fin) is odd. Indeed, suppose $q \notin \{init, fin\}$. Since we fired each transition in T' exactly once to go from `First` to `Second` and since $T' = \{t : \mathbf{x}[t] = 1\}$, by the marking equation for $(\mathcal{N}_\mathcal{P}, \texttt{First}, \texttt{Second})$ it follows that `Second`$(q) -$ `First`$(q) = \sum_{t \in T} \mathcal{A}[q,t] \cdot \mathbf{x}[t]$. Since $\mathbf{x}$ is also a solution to the marking equation for $(\mathcal{N}_\mathcal{P}, \wr init \wr, \wr fin \wr)$ over the field $\mathbb{F}_2$, it follows that $\sum_{t \in T} \mathcal{A}[q,t] \cdot \mathbf{x}[t] \equiv 0 \bmod 2$, and so we can conclude that `Second`$(q) -$ `First`(q) is even. Since `First`(q) is even, it follows that so is `Second`(q). Similarly we can argue that `Second`$(init)$ is even and `Second`(fin) is odd.

Third stage - Desaturate: Hence, we have shown that `Second`$(q) = 0$ for any bad state q and `Second`(q) is even for every good state $q \neq fin$. By Proposition 7.6 and by the monotonicity property, it follows that from `Second`(q) we can reach the configuration C_{fin}^n, thereby constructing the required run. □

Example 7.9. Consider the symmetric protocol $\mathcal{P}$ from Figure 6. We have seen that 2 is a cut-off for this protocol. The Petri net associated with this protocol is given in Figure 7. We

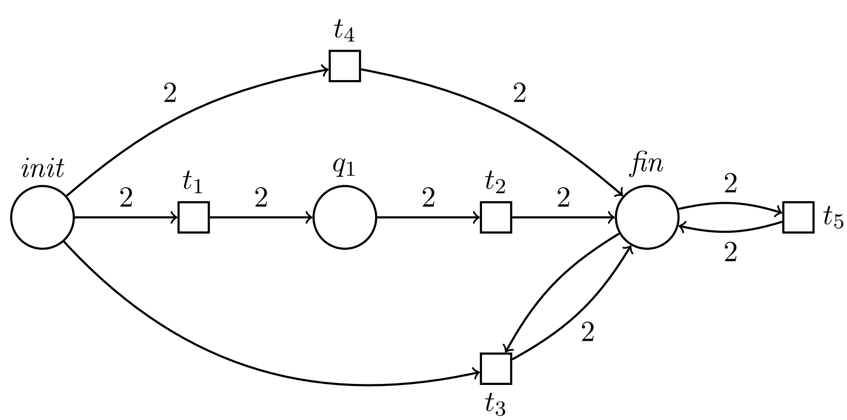

Figure 7: Petri net corresponding to the protocol from Figure 6

now show that the conditions of Lemmas 7.5 and 7.8 are satisfied in this protocol. Indeed, it is easy to see that there is a path from *init* to *fin* in the graph of $\mathcal{P}$ and that there is no bad state. Further, notice that the entries in the incidence matrix $\mathcal{A}$ of this Petri net are all either 0, 2 or -2, except for $\mathcal{A}[init, t_3]$ and $\mathcal{A}[fin, t_3]$ which are -1 and 1 respectively. It follows that if we were to write down the marking equation for the markings $\wr init \int$ and $\wr fin \int$ over the field $\mathbb{F}_2$, then the only non-trivial equations that we would get are $0 = 1 - \mathbf{v}_{t_3}$ and $1 = \mathbf{v}_{t_3}$, which are the equations corresponding to the places *init* and *fin* respectively. Hence if we set $\mathbf{x} = (0, 0, 1, 0, 0)$, then $\mathbf{x}$ is a solution to the marking equation over the field $\mathbb{F}_2$. Since there are no useless transitions in this net, it follows that the conditions of Lemmas 7.5 and 7.8 are satisfied.

7.2. **An NC algorithm.** We now use Proposition 7.4 and Lemmas 7.5 and 7.8 to give an NC algorithm for the cut-off problem for symmetric protocols. Recall that NC is the class of problems that can be solved by *parallel random access machines* (PRAMs) in polylogarithmic parallel time with a polynomially many processors [Pap07]. Intuitively, a PRAM is a parallel computer with some number of random access machines $M_1, M_2, \dots, M_k$, also known as processors, and a central memory which can be accessed by any of these processors. These processors have random access to any of the cells in the central memory and they can all read and write to these cells as dictated by their instruction set. A single step of the PRAM consists of all of these processors executing a single step, with contradictory writes to the same cell avoided by adopting the convention that the write of the processors with the smallest index prevails. A uniform family of polynomial-sized PRAMS is a set $\{P_n : n \in \mathbb{N}\}$ such that each P_n only has a polynomial number of processors in n and there is a logarithmic space Turing machine, which when given input 1^n can output P_n. A language L is said to be in NC if there is a uniform family of polynomial-sized PRAMS $\{P_n : n \in \mathbb{N}\}$ such that given input x, the PRAM $P_{|x|}$ correctly decides if $x \in L$ and the number of steps $P_{|x|}$ takes is polylogarithmic in the size of $|x|$. Similarly, we can define the notion of a uniform family of polynomial-sized PRAMS computing a function, and hence a reduction between two languages.

From the definition of NC, it follows that this class is closed under intersection of languages. Further, it also follows that if $L \in \mathsf{NC}$ and there is a NC-reduction from another language L' to L, then $L' \in \mathsf{NC}$. With these two facts in mind, by Proposition 7.4, to prove that the cut-off problem for symmetric protocols is in NC, it suffices to give two NC algorithms: one to check if there is an even number e such that C^e_{init} can reach C^e_{fin} and another to check if there is an odd number o such that C^o_{init} can reach C^o_{fin}.

Algorithm for the even case. We now give an NC algorithm to determine if there is an even number e such that C^e_{init} can reach C^e_{fin}. To do this, we show that given the protocol $\mathcal{P}$ as input, we can construct the graph $\mathcal{G}_{\mathcal{P}}$ using an NC algorithm. By Lemma 7.5, we have then reduced the problem to an instance of graph reachability, which is in NC [Pap07]. Since NC is closed under NC-reductions, the required claim will then follow.

Let $\mathcal{P} = (Q, \Sigma, init, fin, R)$. We assume that $\mathcal{P}$ is given as input in the central memory in some fixed encoding, i.e., the first $|Q|$ cells contain the encoding of the states, the next $|\Sigma|$ cells contain the encoding of the alphabet and so on. To construct $\mathcal{G}_{\mathcal{P}}$, we will have $n + n^2 \cdot |\Sigma|$ processors working in parallel, where $n = |Q|$. The i^{th} processor in this collection will simply write down the encoding of the i^{th} state onto the memory. Each of the other

$n^2 \cdot |\Sigma|$ processors uniquely correspond to an element from $Q \times \Sigma \times Q$. A processor responsible for the triple consisting of the i^{th} and the j^{th} states and the k^{th} letter first checks if there is a rule in R for this triple by means of random access to the input. Depending on this check, it will then write down whether or not there is an edge in $\mathcal{G}_\mathcal{P}$ corresponding to the i^{th} and the j^{th} states. Notice that we only have a polynomial number of processors and each of these processors only performs a polylogarithmic amount of work. Hence, we have shown how to construct $\mathcal{G}_\mathcal{P}$ using an NC algorithm, which completes the proof of the claim.

Algorithm for the odd case. We now give an NC algorithm to determine if there is an odd number o such that C^o_{init} can reach C^o_{fin}. First note that similar to the NC algorithm which constructs $\mathcal{G}_\mathcal{P}$ from $\mathcal{P}$, we can prove that there is an NC algorithm which constructs $\mathcal{N}_\mathcal{P}$ from $\mathcal{P}$. So we can assume that we have access to both $\mathcal{G}_\mathcal{P}$ and $\mathcal{N}_\mathcal{P}$ for our NC algorithm. We now show that there is an NC algorithm which allows us to identify the set of all bad states and useless transitions of $\mathcal{N}_\mathcal{P}$.

Let $n = |Q|$. Suppose the NC algorithm to decide graph reachability on n vertices requires $f(n)$ processors and takes $g(n)$ (parallel) steps. Then, to compute the set of bad states of $\mathcal{G}_\mathcal{P}$, we take $2nf(n)$ processors. For each i, the i^{th} collection of $f(n)$ processors will decide if there is a path from *init* to the i^{th} state of $\mathcal{P}$ in $\mathcal{G}_\mathcal{P}$ and the $(n+i)^{th}$ collection of $f(n)$ processors will decide if there is a path from the i^{th} state of $\mathcal{P}$ to *fin* in $\mathcal{G}_\mathcal{P}$. Notice that each collection of $f(n)$ processors can work in a parallel manner and hence the overall (parallel) time of this algorithm will still be $g(n)$. Finally, we will have n processors, one for each state, which will wait for the result from the processors responsible for graph reachability and based on this decide if the state corresponding to themselves is bad or not.

Once we have computed the set of bad states, computing the set of useless transitions is easy. The shared memory will have one bit for each transition of $\mathcal{N}_\mathcal{P}$, initially set to 0. For each transition t and each place p of $\mathcal{N}_\mathcal{P}$, we will have one processor which will check if $p \in {}^\bullet t \cup t^\bullet$ and also check if p is a bad state. If this is the case, then that process changes the bit corresponding to t to 1 and otherwise, it does nothing.

Finally, we show that given $\mathcal{N}_\mathcal{P}$ and its set of useless transitions, there is an NC algorithm which constructs the marking equation for $(\mathcal{N}_\mathcal{P}, \wr init \int, \wr fin \int)$ and additionally adds constraints specifying that the solution must be 0 on all useless transitions. Composing this NC algorithm, with the ones for constructing $\mathcal{N}_\mathcal{P}$ and useless transitions, by Lemma 7.8, we have then reduced the problem to an instance of solving equations over the field $\mathbb{F}_2$, which is in NC [Mul87] and so the required claim will then follow.

To construct the marking equation, we will have one processor responsible for each entry of the equation. We will have n processors for constructing the column corresponding to the marking $\wr init \int$ and another n processors for the column corresponding to the marking $\wr fin \int$. Further, for each transition t and each place p, we will have one processor for outputting $\mathcal{A}[p,t]$. This processor will consult the memory and get the values of $Pre[p,t]$ and $Post[p,t]$ and will then compute $\mathcal{A}[p,t]$. Finally, we will have one processor for each transition t, which will check if t is a useless transition (by consulting the memory) and then depending on this check, will write down a constraint specifying that the solution to the marking equation must have value 0 corresponding to the index t. This completes the construction of the NC algorithm and so we get,

Theorem 7.10. *The cut-off problem for symmetric protocols is in* NC.

7.3. **The bounded-loss cut-off problem.** We can also consider the bounded-loss cut-off problem for symmetric protocols. However, in the case of symmetric protocols, this problem becomes rather easy. More specifically, the bounded-loss cut-off problem is NL-complete for symmetric protocols.

Indeed, for a symmetric protocol to have a bounded-loss cut-off, it is easy to see that there must be a path from *init* to *fin* in the graph $\mathcal{G}_{\mathcal{P}}$. Moreover, this is also a sufficient condition, because if such a path exists, then by Proposition 7.6 and the monotonicity property, for every $k \geq 1$, $\wr (2k-1) \cdot init \wr \xrightarrow{*} \wr 2k-2 \cdot fin \wr + \wr init \wr$ and $\wr 2k \cdot init \wr \xrightarrow{*} \wr 2k \cdot fin \wr$ and so we can simply take the bounded-loss cut-off to be 1. Notice that we can compute $\mathcal{G}_{\mathcal{P}}$ from $\mathcal{P}$ by a reduction which uses a logarithmic amount of space. This proves that the bounded-loss cut-off problem for symmetric protocols reduces to graph reachability which is known to be NL-complete.

Moreover, given a graph reachability instance (G, s, t), we can interpret it in a straightforward manner as the graph of a symmetric protocol $\mathcal{P}$ whose communication alphabet is a single letter and where the initial state is s and the final state is t. This then immediately implies that $\mathcal{P}$ would have a bounded-loss cut-off if and only if s can reach t in the graph G, which leads to NL-hardness of the bounded-loss cut-off problem for symmetric protocols.

8. The cut-off problem for symmetric rendez-vous protocols with a leader

In [HS20], Horn and Sangnier proposed an extension to symmetric rendez-vous protocols by adding a special agent called a leader. They defined the cut-off problem for that extension and showed that it is in PSPACE. We improve on their result and and prove that the problem is NP-complete. We proceed by introducing this new model and its cut-off problem.

Definition 8.1. A symmetric rendez-vous protocol with a leader, or simply a *symmetric leader protocol*, is a pair of symmetric protocols $\mathcal{P} = (\mathcal{P}^L, \mathcal{P}^F)$ where $\mathcal{P}^L = (Q^L, \Sigma, init^L, fin^L, R^L)$ is the *leader protocol* and $\mathcal{P}^F = (Q^F, \Sigma, init^F, fin^F, R^F)$ is the *follower protocol* such that $Q^L \cap Q^F = \emptyset$.

Intuitively, we will have exactly one agent in our collection executing the leader protocol and all the other agents will execute the follower protocol. To this end, we define a *configuration* of a symmetric leader protocol $\mathcal{P}$ as a multiset over $Q^L \cup Q^F$ such that $\sum_{q \in Q^L} C(q) = 1$. For each $n \in \mathbb{N}$, let C^n_{init} (resp. C^n_{fin}) denote the initial (resp. final) configuration of $\mathcal{P}$ given by $C(init^L) = 1$ (resp. $C(fin^L) = 1$) and $C(init^F) = n$ (resp. $C(fin^F) = n$).

Suppose a is a message in Σ and $r = (p, a, p')$ and $r' = (q, a, q')$ are two rules of $R^L \cup R^F$. For any two configurations C and C', we say that $C \xRightarrow{r,r'} C'$ if $C \geq \wr p, q \wr$ and $C' = C - \wr p, q \wr + \wr p', q' \wr$. Since we allow exactly one agent to execute the leader protocol, given a configuration C, we let $\texttt{lead}(C)$ denote the unique state q of the leader protocol such that $C(q) > 0$. *The cut-off problem for symmetric leader protocols* is defined as the following decision problem.

Given: A symmetric leader protocol $\mathcal{P} = (\mathcal{P}^L, \mathcal{P}^F)$.
Decide: If there exists $B \in \mathbb{N}$ such that for all $n \geq B$, $C^n_{init} \xRightarrow{*} C^n_{fin}$.

If such a B exists, then we say that $\mathcal{P}$ admits a cut-off and that B is a cut-off for $\mathcal{P}$.

Example 8.2. Let us consider the symmetric leader protocol $\mathcal{P} = (\mathcal{P}^L, \mathcal{P}^F)$ where $\mathcal{P}^L$ is given in Figure 8 and $\mathcal{P}^F$ is given in Figure 9. Note that the follower protocol is obtained from the one in Figure 6, by removing the self-loop at the state *fin*.

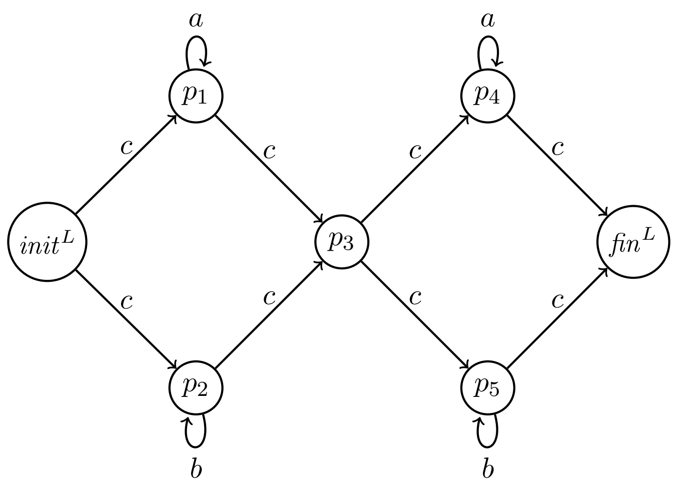

Figure 8: The leader protocol

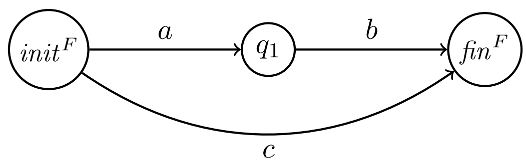

Figure 9: The follower protocol

We claim that 4 is a cut-off for this protocol. Indeed, notice that by only using the message c we have the run $\wr\, init^L + 4\cdot init^F \,\wr \Rightarrow \wr\, p_1 + 3\cdot init^F + fin^F \,\wr \Rightarrow \wr\, p_3 + 2\cdot init^F + 2\cdot fin^F \,\wr \Rightarrow \wr\, p_4 + init^F + 3\cdot fin^F \,\wr \Rightarrow \wr\, fin^L + 4\cdot init^F \,\wr$. In addition to these steps, if we also make the leader take the rules corresponding to the messages a and b at p_1 and p_5 respectively we have the run $\wr\, init^L + 5\cdot init^F \,\wr \Rightarrow \wr\, p_1 + 4\cdot init^F + fin^F \,\wr \Rightarrow \wr\, p_1 + 3\cdot init^F + q_1 + fin^F \,\wr \overset{*}{\Rightarrow} \wr\, p_5 + init^F + q_1 + 3\cdot fin^F \,\wr \Rightarrow \wr\, p_5 + init^F + 4\cdot fin^F \,\wr \Rightarrow \wr\, fin^L + 5\cdot fin^F \,\wr$.

If $n \geq 4$ and n is even (resp. odd), notice that C^n_{init} can reach the configuration $C^4_{init} + \wr\, n - 4\cdot fin^F \,\wr$ (resp. $C^5_{init} + \wr\, n - 5\cdot fin^F \,\wr$) by using the rules in the follower protocol corresponding to the message c. From this observation and from the above paragraph, it follows that 4 is a cut-off for this symmetric leader protocol.

Further, suppose $\rho := C^{2k+1}_{init} \xrightarrow{r_1, r'_1} C_1 \xrightarrow{r_2, r'_2} C_2 \dots \xrightarrow{r_w, r'_w} C^{2k+1}_{fin}$ is a valid run for some k. We shall now give an informal sketch of a proof that there must be an occurrence of at least one of the rules in $\{(p_1, a, p_1), (p_4, a, p_4)\}$ along ρ. Let $S := \{(init^L, c, p_i), (p_i, c, p_3) : 1 \leq i \leq 2\} \cup \{(p_3, c, p_i), (p_i, c, fin^L) : 4 \leq i \leq 5\}$. By the construction of the leader protocol, it follows that there are exactly 4 occurrences of rules from S in the run ρ. For every i, notice that the rule $(init^F, c, fin^F)$ can appear exactly once in the pair (r_i, r'_i) if and only if the other rule which appears in the pair is in S. It then follows that the number of occurrences of $(\mathit{init}^F, c, \mathit{fin}^F)$ along ρ is even. Since we begin with an odd number of agents in init^F, it follows that an odd number of agents must have taken the rule $(\mathit{init}^F, a, q_1)$ along ρ. Once again notice that for every i, the rule $(\mathit{init}^F, a, q_1)$ can appear exactly once in the pair (r_i, r_i') if and only if the other rule which appears in the pair is in $\{(p_1, a, p_1), (p_4, a, p_4)\}$. Since $(\mathit{init}^F, a, q_1)$ is fired by an odd number of agents, it follows that there must be an occurrence of at least one of the rules in $\{(p_1, a, p_1), (p_4, a, p_4)\}$ along ρ.

A generalization of this protocol along with the argument given above will be useful to prove the lower bound on the cut-off problem in Subsection 8.2.

The main theorem of this section is that

Theorem 8.3. *The cut-off problem for symmetric leader protocols is* NP*-complete.*

The following characterization of symmetric leader protocols that admit a cut-off, proved by Horn and Sangnier, will be central to us for proving both the upper bound and the lower bound from Theorem 8.3.

Proposition 8.4 [HS20, Lemma 18]**.** *A symmetric leader protocol admits a cut-off if and only if there exist an even number e and an odd number o such that C^e_{init} can reach C^e_{fin} and C^o_{init} can reach C^o_{fin}.*

8.1. **The upper bound: An NP algorithm.** Let $\mathcal{P}_1 = (\mathcal{P}_1^L, \mathcal{P}_1^F)$ be a symmetric leader protocol where $\mathcal{P}_1^L = (Q_1^L, \Sigma, \mathit{init}_1^L, \mathit{fin}_1^L, R_1^L)$ and $\mathcal{P}^F = (Q_1^F, \Sigma, \mathit{init}_1^F, \mathit{fin}_1^F, R_1^F)$.

Preprocessing the protocol. As a first step, we preprocess the protocol to remove certain states, whilst preserving the cut-off property. Since the protocols $\mathcal{P}_1^L$ and $\mathcal{P}_1^F$ are symmetric, we can consider the graphs $\mathcal{G}_{\mathcal{P}_1^L}$ and $\mathcal{G}_{\mathcal{P}_1^F}$ as defined in Section 7. Recall that a bad state of a symmetric protocol is one which either does not have a path from the initial state or to the final state in the graph of its protocol. Similar to Proposition 7.7, we have the following proposition for symmetric leader protocols, which intuitively states that we can discard all the bad states of $\mathcal{P}_1^L$ and $\mathcal{P}_1^F$ without sacrificing the validity of the cut-off property.

Proposition 8.5. *Suppose C is such that $C^n_{\mathit{init}} \xrightarrow{*} C$ (resp. $C \xrightarrow{*} C^n_{\mathit{fin}}$) for some n. Then for every $q \in Q_1^F$ such that $C(q) > 0$, there is a path from init_1^F to q (resp. from q to fin_1^F) in the graph $\mathcal{G}_{\mathcal{P}_1^F}$. Similarly for every $q \in Q_1^L$ such that $C(q) > 0$, there is a path from init_1^L to q (resp. from q to fin_1^L) in the graph $\mathcal{G}_{\mathcal{P}_1^L}$.*

Proof. Straightforward induction on the length of the underlying run. □

Let $\mathcal{P}^L$ (resp. $\mathcal{P}^F$) be the symmetric protocol obtained from $\mathcal{P}_1^L$ (resp. $\mathcal{P}_1^F$) by removing all of its bad states (and the associated rules) and let $\mathcal{P} = (\mathcal{P}^L, \mathcal{P}^F)$. Note that $\mathcal{P}$ can be constructed from $\mathcal{P}_1$ in polynomial time. By Proposition 8.5, it follows that,

Lemma 8.6. *$\mathcal{P}$ admits a cut-off if and only if $\mathcal{P}_1$ admits a cut-off.*

Proof. The left-to-right implication is true because $\mathcal{P}$ is obtained from $\mathcal{P}_1$ by potentially discarding some of its transitions and so if a configuration C can reach a configuration C' in $\mathcal{P}$, then the same is true for $\mathcal{P}_1$ as well. For the other direction, suppose $\rho := C^n_{\mathit{init}} \to C_1 \to C_2 \to \ldots \to C_k \to C^n_{\mathit{fin}}$ is a valid run in $\mathcal{P}_1$. By Proposition 8.5, it follows that if

$C_i(q) > 0$ for any $1 \le i \le n$ and any state q, then q is not a bad state of either $\mathcal{P}_1^L$ or $\mathcal{P}_1^F$. Since we obtained $\mathcal{P}$ from $\mathcal{P}_1$ by only pruning the bad states of the underlying protocols, it follows that ρ is also a valid run in $\mathcal{P}$. Hence, if $\mathcal{P}_1$ admits a cut-off, then $\mathcal{P}$ also admits a cut-off. □

Because $\mathcal{P}^F$ is symmetric and does not contain any bad states, we have the following fact, which is easy to verify.

Proposition 8.7. *If q^L and q^F are states of $\mathcal{P}^L$ and $\mathcal{P}^F$ respectively, then $\wr q^L + 2 \cdot init^F \wr \xrightarrow{*} \wr q^L + 2 \cdot q^F \wr \xrightarrow{*} \wr q^L + 2 \cdot fin^F \wr$.*

In the rest of this subsection, we will concern ourselves with only $\mathcal{P} = (\mathcal{P}^L, \mathcal{P}^F)$ and show how to decide if it admits a cut-off.

Constructing a Petri net. Similar to the case of rendez-vous protocols, we will construct a Petri net $\mathcal{N}_\mathcal{P}$ from the symmetric leader protocol $\mathcal{P} = (\mathcal{P}^L, \mathcal{P}^F)$. However, the construction is a bit different now because of the leader protocol.

Let $\mathcal{P}^L = (Q^L, \Sigma, init^L, fin^L, R^L)$ and $\mathcal{P}^F = (Q^F, \Sigma, init^F, fin^F, R^F)$. Our Petri net $\mathcal{N}_\mathcal{P} = (P, T, Pre, Post)$ will be constructed as follows: The set of places P will be $Q^L \cup Q^F$. For each $a \in \Sigma$ and $r = (q, a, s), r' = (q', a, s') \in R^L \cup R^F$ such that *not both r and r' belong to R^L*, we will have a transition $t_{r,r'}$ in $\mathcal{N}_\mathcal{P}$ satisfying

- $Pre[p, t_{r,r'}] = 0$ for every $p \notin \{q, q'\}$, $Post[p, t_{r,r'}] = 0$ for every $p \notin \{s, s'\}$
- If $q = q'$ then $Pre[q, t_{r,r'}] = 2$, otherwise $Pre[q, t_{r,r'}] = Pre[q', t_{r,r'}] = 1$
- If $s = s'$ then $Post[s, t_{r,r'}] = 2$, otherwise $Post[s, t_{r,r'}] = Post[s', t_{r,r'}] = 1$.

The transitions defined here have the same intuitive meaning as the ones defined in Subsection 2.3. The reason we only consider pairs of rules (r, r') such that not both of them belong to R^L is that since we will only have one agent executing the leader protocol, we can never have a step between configurations in which two rules in R^L are executed.

If $t_{r,r'}$ is a transition such that *exactly one of r and r'* is in R^L, then it will be called a *leader transition*. All other transitions will be called *follower-only transitions*. Notice that if t is a leader transition, then there is a unique place $t.from \in {}^\bullet t \cap Q^L$ and a unique place $t.to \in t^\bullet \cap Q^L$.

As usual, we let $\mathcal{A} = Post - Pre$ denote the incidence matrix of $\mathcal{N}_\mathcal{P}$. The next proposition follows from the construction of $\mathcal{N}_\mathcal{P}$.

Proposition 8.8. *For any two configurations C and C' we have that $C \overset{*}{\Rightarrow} C'$ over the protocol $\mathcal{P}$ if and only if $C \xrightarrow{*} C'$ over the Petri net $\mathcal{N}_\mathcal{P}$.*

Because of Proposition 8.8 and Proposition 8.4 to decide whether $\mathcal{P}$ admits a cut-off, it suffices to give an NP algorithm which decides if there is an even number e and an odd number o such that C^e_{init} can reach C^e_{fin} and C^o_{init} can reach C^o_{fin} over $\mathcal{N}_\mathcal{P}$. The forthcoming subsections are dedicated to proving that such an NP algorithm exists.

The notion of compatibility. To give our NP algorithm, we define the notion of *compatibility* between a configuration C and a vector $\mathbf{x} \in \mathbb{N}^T$, where T is the set of transitions of $\mathcal{N}_\mathcal{P}$. We show that compatible pairs of the form $(C, \mathbf{x})$ admit certain nice properties in terms of runs in the net $\mathcal{N}_\mathcal{P}$.

For any set of leader transitions S of $\mathcal{N}_\mathcal{P}$, define a graph $\mathcal{G}(S)$ as follows: The set of vertices of $\mathcal{G}(S)$ is given by $\{t.from : t \in S\} \cup \{t.to : t \in S\}$ and its set of edges is given by $\{(t.\mathit{from}, t.\mathit{to}) : t \in S\}$. Notice that every edge in $\mathcal{G}(S)$ corresponds to a unique leader transition in S. For this reason, we will often identify each edge in $\mathcal{G}(S)$ with its corresponding leader transition in S.

For any vector $\mathbf{x} \in \mathbb{N}^T$, define $\mathtt{lset}(\mathbf{x})$ to be the set of all leader transitions such that $\mathbf{x}[t] > 0$. The graph of the vector $\mathbf{x}$, denoted by $\mathcal{G}(\mathbf{x})$ is defined as the graph $\mathcal{G}(\mathtt{lset}(\mathbf{x}))$. Recall that for a configuration C, there is exactly one state q of the leader protocol for which $C(q) > 0$ and this state is denoted by $\mathtt{lead}(C)$. We are now in a position to define the notion of compatibility.

Definition 8.9. Let C be a configuration and let $\mathbf{x} \in \mathbb{N}^T$. We say that the pair $(C, \mathbf{x})$ is *compatible* if $C + \mathcal{A}\mathbf{x} \geq \mathbf{0}$ and every vertex in $\mathcal{G}(\mathbf{x})$ is reachable from the state $\mathtt{lead}(C)$.

Note that if $C \xrightarrow{\sigma} C'$ is a valid run in $\mathcal{N}_{\mathcal{P}}$, then $(C, \overrightarrow{\sigma})$ is a compatible pair. Indeed, we have $\mathbf{0} \leq C' = C + \mathcal{A}\overrightarrow{\sigma}$ and so the first condition of being compatible is satisfied. The other condition can be shown by induction on the length of σ. The following lemma acts as a sort of converse to this property. It states that *as long as there are enough followers in every state of the follower protocol*, it is possible to come up with a firing sequence from a compatible pair.

Lemma 8.10 (**Compatibility Lemma**). *Suppose $(C, \mathbf{x})$ is a compatible pair such that $C(q) \geq 2\|\mathbf{x}\|$ for every $q \in Q^F$. Then there is a firing sequence ξ such that $C \xrightarrow{\xi}$ and $\overrightarrow{\xi} = \mathbf{x}$.*

Proof. First, we give an intuition behind the proof. The proof proceeds by induction on $\|\mathbf{x}\|$. Suppose $\mathbf{x}[t] > 0$ for some follower-only transition. In this case, we simply execute t from C to reach C', and we reduce the value corresponding to t in $\mathbf{x}$ by 1 to get a vector $\mathbf{x}'$. We can then show that $(C', \mathbf{x}')$ is compatible and $C'(q) \geq 2\|\mathbf{x}'\|$ for every $q \in Q^F$ and so we can apply the induction hypothesis to handle this case.

Suppose $\mathbf{x}[t] > 0$ for some leader transition. Let $p = \mathtt{lead}(C)$. Suppose p belongs to some cycle $S = p, r_1, p_1, r_2, p_2, \ldots, p_k, r_{k+1}, p$ in the graph $\mathcal{G}(\mathbf{x})$, where $r_1, r_2, \ldots, r_{k+1}$ are leader transitions, $p = r_1.\mathit{from} = r_{k+1}.\mathit{to}$ and for each $1 \leq i \leq k$, we have $p_i = r_i.\mathit{to} = r_{i+1}.\mathit{from}$. We then let $C \xrightarrow{r_1} C'$ and let $\mathbf{x}'$ be the same vector as $\mathbf{x}$, except that $\mathbf{x}'[r_1] = \mathbf{x}[r_1] - 1$. We can then verify that $C' + \mathcal{A}\mathbf{x}' \geq \mathbf{0}$, $C'(q) \geq 2\|\mathbf{x}'\|$ for every $q \in Q^F$ and $\mathtt{lead}(C') = p_1$. Any path W in $\mathcal{G}(\mathbf{x})$ from p to some vertex s either goes through p_1 or we can use the cycle S to go from p_1 to p first and then use W to reach s. This gives a path from p_1 to every vertex s in $\mathcal{G}(\mathbf{x}')$, which will enable us to prove that $(C', \mathbf{x}')$ is also compatible.

If p does not belong to any cycle in $\mathcal{G}(\mathbf{x})$, then using the fact that $C + \mathcal{A}\mathbf{x} \geq 0$, we can show that there is exactly one out-going edge t from p in $\mathcal{G}(\mathbf{x})$. We then let $C \xrightarrow{t} C'$ and let $\mathbf{x}'$ be the same vector as $\mathbf{x}$, except that $\mathbf{x}'[t] = \mathbf{x}[t] - 1$. Since any path in $\mathcal{G}(\mathbf{x})$ from p has to necessarily use this edge t, it follows that in $\mathcal{G}(\mathbf{x}')$ there is a path from $t.\mathit{to} = \mathtt{lead}(C')$ to every vertex. This will then enable us to prove that $(C', \mathbf{x}')$ is also compatible. We now proceed to the formal proof.

For any transition $t \in T$, we let $\boldsymbol{t}$ denote the vector in $\mathbb{N}^T$ which has a 1 in the co-ordinate corresponding to t and 0 everywhere else. We recall from the construction of $\mathcal{N}_{\mathcal{P}}$ that its weight is 2, i.e., the largest absolute value appearing in the *Pre* and *Post* matrices of $\mathcal{N}_{\mathcal{P}}$ is 2. We will use this fact implicitly throughout the proof.

Let $n = \|\mathbf{x}\|$. We will construct a sequence of pairs $(C_0, \mathbf{x}_0), (C_1, \mathbf{x}_1), \ldots, (C_n, \mathbf{x}_n)$, each of which are compatible, and also a sequence of transitions $t_1, \ldots, t_n$ satisfying the following conditions.

(1) $C_0 = C,\ \mathbf{x}_0 = \mathbf{x}$

(2) $C_i \xrightarrow{t_{i+1}} C_{i+1}$ for every $i < n$.

(3) $\mathbf{x}_{i+1} = \mathbf{x}_i - \boldsymbol{t_{i+1}}$ for every $i < n$.

(4) $C_i(q) \geq 2n - 2i$ for every $q \in Q^F$ and every $i < n$.

Assume that we have already constructed the pair $(C_i, \mathbf{x}_i)$ for some $i < n$. We now show how to construct the pair $(C_{i+1}, \mathbf{x}_{i+1})$. We consider three cases.

Case 1: Suppose $\mathbf{x}_i[t] > 0$ for some follower-only transition t. By assumption $C_i(q) \geq 2n - 2i \geq 2$ for every $q \in Q^F$. Hence, it follows that the transition t can be fired from C_i. We let $t_{i+1} := t$, C_{i+1} be the configuration satisfying $C_i \xrightarrow{t} C_{i+1}$ and $\mathbf{x}_{i+1} = \mathbf{x}_i - \boldsymbol{t}$. By construction, $(C_{i+1}, \mathbf{x}_{i+1})$ satisfies conditions (1), (2) and (3). Further, since C_i satisfies condition (4), and since the weight of $\mathcal{N}_\mathcal{P}$ is at most 2, it follows that C_{i+1} also satisfies condition (4). Also, $C_{i+1} + \mathcal{A}\mathbf{x}_{i+1} = C_i + \mathcal{A}\boldsymbol{t} + \mathcal{A}(\mathbf{x}_i - \boldsymbol{t}) = C_i + \mathcal{A}\mathbf{x}_i \geq \mathbf{0}$. Finally, note that $\mathcal{G}(\mathbf{x}_{i+1}) = \mathcal{G}(\mathbf{x}_i)$ and $\texttt{lead}(C_{i+1}) = \texttt{lead}(C_i)$ and since $(C_i, \mathbf{x}_i)$ is compatible, it follows that $(C_{i+1}, \mathbf{x}_{i+1})$ is also compatible.

Case 2: Suppose $\mathbf{x}_i[t] > 0$ for some leader transition t. Therefore the graph $\mathcal{G}(\mathbf{x}_i)$ has at least one edge. Let $p = \texttt{lead}(C_i)$. We split this case into two further subcases:

Subcase 2a: Suppose p is part of some cycle in $\mathcal{G}(\mathbf{x}_i)$. Let $p, r_1, p_1, r_2, p_2, \ldots, p_k, r_{k+1}, p$ be a shortest cycle containing p in $\mathcal{G}(\mathbf{x}_i)$, where $r_1, r_2, \ldots, r_{k+1}$ are leader transitions, $p = r_1.\mathit{from} = r_{k+1}.\mathit{to}$ and for each $1 \leq j \leq k$, we have $p_j = r_j.\mathit{to} = r_{j+1}.\mathit{from}$. By construction of $\mathcal{G}(\mathbf{x}_i)$, $\mathbf{x}_i[r_j] > 0$ for each j. Since $p = \texttt{lead}(C_i)$ it follows that $C_i(p) > 0$. Further, by assumption $C_i(q) \geq 2n - 2i \geq 2$ for every $q \in Q^F$. It follows then that r_1 is a transition which can be fired at the configuration C_i. We let $t_{i+1} := r_1$, C_{i+1} be the configuration satisfying $C_i \xrightarrow{r_1} C_{i+1}$ and $\mathbf{x}_{i+1} := \mathbf{x}_i - \boldsymbol{r_1}$. By arguments that are similar to the one given in Case 1, we can conclude that $(C_{i+1}, \mathbf{x}_{i+1})$ satisfies Conditions (1), (2), (3), (4) and also that $C_{i+1} + \mathcal{A}\mathbf{x}_{i+1} = C_i + \mathcal{A}\mathbf{x}_i \geq \mathbf{0}$.

Hence, all that is left to prove is that every vertex in $\mathcal{G}(\mathbf{x}_{i+1})$ is reachable from $\texttt{lead}(C_{i+1})$ which by construction is p_1. Note that $\mathcal{G}(\mathbf{x}_{i+1})$ is necessarily a subgraph of $\mathcal{G}(\mathbf{x}_i)$ which further satisfies the property that if $tr \neq r_1$ is some transition in $\mathcal{G}(\mathbf{x}_i)$, then tr is also present in $\mathcal{G}(\mathbf{x}_{i+1})$. Now, let s be any vertex of $\mathcal{G}(\mathbf{x}_{i+1})$. Since $\mathcal{G}(\mathbf{x}_{i+1})$ is a subgraph of $\mathcal{G}(\mathbf{x}_i)$, it follows that s is also present in $\mathcal{G}(\mathbf{x}_i)$. By compatibility of $(C_i, \mathbf{x}_i)$, s is reachable from p in $\mathcal{G}(\mathbf{x}_i)$. Let $p, tr_1, s_1, tr_2, s_2, \ldots, s_l, tr_{l+1}, s$ be a path in $\mathcal{G}(\mathbf{x}_i)$.

If $tr_1 = r_1$ then $s_1 = p_1$ and hence $p_1, tr_2, s_2, \ldots, s_l, tr_{l+1}, s$ is a path in $\mathcal{G}(\mathbf{x}_{i+1})$. If $tr_1 \neq r_1$ then $p_1, r_2, p_2, \ldots, p_k, r_{k+1}, p, tr_1, s_1, \ldots, s_l, tr_{l+1}, s$ is a walk in $\mathcal{G}(\mathbf{x}_{i+1})$ and so there must be a path from p_1 to s in $\mathcal{G}(\mathbf{x}_{i+1})$. In either case, we are done.

Subcase 2b: Suppose p is not part of any cycle in $\mathcal{G}(\mathbf{x}_i)$. Let $\mathit{Out} = \{\mathit{out}_1, \ldots, \mathit{out}_b\}$ be the set of leader transitions t such that $t.\mathit{from} = p$ and let $\mathit{In} = \{\mathit{in}_1, \ldots, \mathit{in}_a\}$ be the set of leader transitions t such that $t.\mathit{to} = p$. Notice that Out and In depend only on $\mathcal{P}^L$ and *not* on $\mathcal{G}(\mathbf{x}_i)$.

We claim that for every $\mathit{in} \in \mathit{In}$, $\mathbf{x}_i[\mathit{in}] = 0$ and there is exactly one $\mathit{out} \in \mathit{Out}$ such that $\mathbf{x}_i[\mathit{out}] > 0$. Since $(C_i, \mathbf{x}_i)$ is compatible, every vertex in $\mathcal{G}(\mathbf{x}_i)$ is reachable from p and by assumption p is not part of any cycle in $\mathcal{G}(\mathbf{x}_i)$. It follows then that there are no incoming edges to p in $\mathcal{G}(\mathbf{x}_i)$ and hence $\mathbf{x}_i[\mathit{in}] = 0$ for every $\mathit{in} \in \mathit{In}$. Also since every vertex in $\mathcal{G}(\mathbf{x}_i)$ is reachable from p and since $\mathcal{G}(\mathbf{x}_i)$ has at least one edge, it follows that there is at least one $out \in Out$ such that $\mathbf{x}_i[out] > 0$. Finally, since $C_i + \mathcal{A}\mathbf{x}_i \geq \mathbf{0}$ it follows that $C_i(p) + \sum_{in \in In} \mathbf{x}_i[in] - \sum_{out \in Out} \mathbf{x}_i[out] \geq 0$. Since $\mathbf{x}_i[in] = 0$ for every $in \in In$ and since $C_i(p) = 1$, it follows that there is exactly one $out \in Out$ such that $\mathbf{x}_i[out] > 0$. Hence, by construction of $\mathcal{G}(\mathbf{x}_i)$, there is exactly one outgoing edge from p in $\mathcal{G}(\mathbf{x}_i)$, which we will denote by r.

Let $t_{i+1} := r$, C_{i+1} be the configuration satisfying $C_i \xrightarrow{r} C_{i+1}$ and $\mathbf{x}_{i+1} := \mathbf{x}_i - \boldsymbol{r}$. By arguments that are similar to the one given in Case 1, we can conclude that $(C_{i+1}, \mathbf{x}_{i+1})$ satisfies Conditions (1), (2), (3), (4) and also that $C_{i+1} + \mathcal{A}\mathbf{x}_{i+1} = C_i + \mathcal{A}\mathbf{x}_i \geq \mathbf{0}$.

All that is left to prove is that every vertex in $\mathcal{G}(\mathbf{x}_{i+1})$ is reachable from $\texttt{lead}(C_{i+1})$ which by construction is $r.to$. Note that $\mathcal{G}(\mathbf{x}_{i+1})$ is necessarily a subgraph of $\mathcal{G}(\mathbf{x}_i)$ which further satisfies the property that if $tr \neq r$ is some transition in $\mathcal{G}(\mathbf{x}_i)$, then tr is also present in $\mathcal{G}(\mathbf{x}_{i+1})$. Now, let s be any vertex of $\mathcal{G}(\mathbf{x}_{i+1})$. Since $\mathcal{G}(\mathbf{x}_{i+1})$ is a subgraph of $\mathcal{G}(\mathbf{x}_i)$, it follows that s is also present in $\mathcal{G}(\mathbf{x}_i)$. By compatibility of $(C_i, \mathbf{x}_i)$, s is reachable from p in $\mathcal{G}(\mathbf{x}_i)$. Let $p, tr_1, s_1, tr_2, s_2, \ldots, s_l, tr_{l+1}, s$ be a path in $\mathcal{G}(\mathbf{x}_i)$. Since p has only one outgoing edge in $\mathcal{G}(\mathbf{x}_i)$ (which is r), it follows that $tr_1 = r$ and $s_1 = r.to$. Hence $s_1, tr_2, s_2, \ldots, s_l, tr_{l+1}, s$ is a path in $\mathcal{G}(\mathbf{x}_{i+1})$ and so the proof of this case is also complete.

Let $\xi := t_1, t_2, \ldots, t_n$. By construction $C_0 \xrightarrow{\xi} C_n$. Further, since each $\mathbf{x}_{i+1} = \mathbf{x}_i - \boldsymbol{t_{i+1}}$, it follows that $\mathbf{x}_n = \mathbf{x}_0 - \overrightarrow{\sigma}$. Since $n = \|\mathbf{x}\|$, it follows that $\mathbf{x}_n = \mathbf{0}$. Hence, $\overrightarrow{\sigma} = \mathbf{x}_0 = \mathbf{x}$ and so the claim of the lemma is true. □

Characterization of symmetric leader protocols that admit a cut-off. We use the Compatibility lemma to prove another characterization of symmetric leader protocols that admit a cut-off, which will help us construct our final NP algorithm.

Lemma 8.11. *For every par* $\in \{0,1\}$*, there exists* $k \in \mathbb{N}$ *such that* $C^k_{init} \xrightarrow{*} C^k_{fin}$ *and* $k \equiv par \bmod 2$ *if and only if there exist* $n \in \mathbb{N}$*,* $\mathbf{x} \in \mathbb{N}^T$ *such that* $n \equiv par \bmod 2$*,* $(C^n_{init}, \mathbf{x})$ *is compatible and* $C^n_{fin} = C^n_{init} + \mathcal{A}\mathbf{x}$*.*

Proof. ($\Rightarrow$): Suppose there exist $k \in \mathbb{N}$ and a firing sequence σ such that $k \equiv par \bmod 2$ and $C^k_{init} \xrightarrow{\sigma} C^k_{fin}$. Let $\mathbf{x} = \overrightarrow{\sigma}$. By the marking equation $C^k_{fin} = C^k_{init} + \mathcal{A}\mathbf{x}$. Since $C^k_{fin} \geq \mathbf{0}$, to prove that $(C^k_{init}, \mathbf{x})$ is compatible it is enough to prove that every vertex in $\mathcal{G}(\mathbf{x})$ is reachable from $\texttt{lead}(C^k_{init}) = init^L$.

Let $t_1, \ldots, t_l$ be leader transitions sorted by the order in which they first appear in σ. We have to show that for every i, the states $t_i.from$ and $t_i.to$ are reachable from $init^L$ in $\mathcal{G}(\mathbf{x})$. Since σ is a firing sequence from C^k_{init} to C^k_{fin}, it must be the case that for each t_i, the place $t_i.from \in \{init^L, t_1.to, t_2.to, \ldots, t_{i-1}.to\}$. Our claim then follows by means of this observation and induction on i.

($\Leftarrow$): Suppose there exist $n \in \mathbb{N}$, $\mathbf{x} \in \mathbb{N}^T$ such that $n \equiv par \bmod 2$, $(C^n_{init}, \mathbf{x})$ is compatible and $C^n_{fin} = C^n_{init} + \mathcal{A}\mathbf{x}$. We will now find a $k \in \mathbb{N}$ such that $k \equiv par \bmod 2$ and $C^k_{init} \xrightarrow{*} C^k_{fin}$. Let $\lambda = 2\|\mathbf{x}\| \cdot |Q^F|$ and let $k = \lambda + n$. Note that $k \equiv par \bmod 2$. Similar to the proof of the Insertion Lemma (Lemma 4.7), we now construct the required run in three stages.

First, by Proposition 8.7 and the monotonicity property, from C^k_{init} we can reach the marking $\texttt{First} := C^n_{init} + \sum_{q \in Q^F} \wr 2\|\mathbf{x}\| \cdot q \wr$. Next, notice that since $(C^n_{init}, \mathbf{x})$ is compatible, so is $(\texttt{First}, \mathbf{x})$. By the Compatibility lemma (Lemma 8.10), it follows that there is a firing sequence ξ and a configuration `Second` such that $\overrightarrow{\xi} = \mathbf{x}$ and $\texttt{First} \xrightarrow{\xi} \texttt{Second}$. By the marking equation, it follows that $\texttt{Second} = C^n_{fin} + \sum_{q \in Q^F} \wr 2\|\mathbf{x}\| \cdot q \wr$. Finally, by Proposition 8.7 and the monotonicity property, we can reach the marking C^k_{fin} from `Second`, which concludes the construction. □

Lemma 8.12. *Given a symmetric leader protocol, deciding whether it admits a cut-off can be done in* NP.

Proof. By Proposition 8.4, it suffices to decide if there exist an even number e and an odd number o such that $C^e_{init} \xrightarrow{*} C^e_{fin}$ and $C^o_{init} \xrightarrow{*} C^o_{fin}$. Suppose we want to check that there exists $k \in \mathbb{N}$ such that $C^{2k}_{init} \xrightarrow{*} C^{2k}_{fin}$. By Lemma 8.11, this is possible if and only if there exist $k \in \mathbb{N}$ and $\mathbf{x} \in \mathbb{N}^T$ such that $(C^{2k}_{init}, \mathbf{x})$ is compatible and $C^{2k}_{fin} = C^{2k}_{init} + \mathcal{A}\mathbf{x}$. By definition of compatibility, this is equivalent to saying that there exist $k \in \mathbb{N}$, $\mathbf{x} \in \mathbb{N}^T$ and a subset S of leader transitions such that $[\![\mathbf{x}]\!] = S$, $C^{2k}_{fin} = C^{2k}_{init} + \mathcal{A}\mathbf{x}$ and every vertex is reachable from the vertex $init^L$ in the graph $\mathcal{G}(S)$.

This characterization immediately suggests the following NP algorithm. We first non-deterministically guess a set S of leader transitions and check if every vertex in $\mathcal{G}(S)$ is reachable from $init^L$. Then, we write a polynomial sized non-negative integer linear program as follows: We let $\mathbf{v}$ denote $|T|$ variables, one for each transition in T and we let n be another variable, with all of them ranging over the non-negative integers. The constraints of the linear program are given by $C^{2n}_{fin} = C^{2n}_{init} + \mathcal{A}\mathbf{v}$ and $\mathbf{v}[t] = 0 \iff t \notin S$. Once we have constructed this linear program, we solve it, which we can do in non-deterministic polynomial time [Pap81]. If there exists a solution, then we accept. Otherwise, we reject.

From our characterization and the construction of our algorithm, it follows that at least one of the runs of our non-deterministic algorithm is accepting if and only if there exists $k \in \mathbb{N}$ such that $C^{2k}_{init} \xrightarrow{*} C^{2k}_{fin}$. Similarly we can check if there exists $l \in \mathbb{N}$ such that $C^{2l+1}_{init} \xrightarrow{*} C^{2l+1}_{fin}$. □

8.2. NP-hardness.

We now complement our upper bound by proving NP-hardness of the cut-off problem for symmetric leader protocols.

Lemma 8.13. *The cut-off problem for symmetric leader protocols is* NP*-hard.*

Proof. We prove NP-hardness by giving a reduction from 3-SAT. Let $\varphi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a 3-CNF formula with variables $x_1, \ldots, x_n$, where each clause C_j is of the form $C_j := \ell_{j_1} \vee \ell_{j_2} \vee \ell_{j_3}$ for some literals $\ell_{j_1}, \ell_{j_2}, \ell_{j_3}$. We now construct a symmetric leader protocol $\mathcal{P} = (\mathcal{P}^L, \mathcal{P}^F)$ as follows.

Communication alphabet. First, we specify our alphabet Σ. For each clause C_j, we will have a letter c_j. Further, we will also have a special letter α.

States of the leader protocol. For each variable x_j, the leader will have three states $p_j, \top_j$ and $\bot_j$. The leader's initial and final states will be p_0 and p_n respectively.

States of the follower protocol. For each clause C_j, the follower will have a state q_j. The follower's initial and final states will be q_0 and q_m respectively.

Rules of the leader protocol. For each $0 \leq j \leq n-1$, the rules for the leader at the state p_j are given by the gadget in Figure 10. Intuitively, if the leader decides to move to $\top_{j+1}$ (resp. $\bot_{j+1}$) then she has decided to set the variable x_{j+1} in the formula φ to true (resp. false). Hence, upon moving to $\top_{j+1}$ (resp. $\bot_{j+1}$), the leader tries to let the followers know of the clauses which become true because of setting x_{j+1} to true (resp. false).

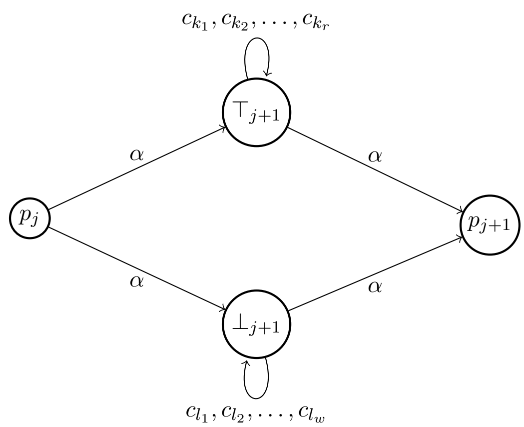

Figure 10: Gadget at state p_j. Here $C_{k_1}, \ldots, C_{k_r}$ denote the set of all clauses in which the literal x_j appears and $C_{l_1}, \ldots, C_{l_w}$ denote the set of all clauses in which the literal $\overline{x_j}$ appears.

Rules of the follower protocol. For each $0 \leq j \leq m-1$, there is a rule (q_j, c_{j+1}, q_{j+1}). Finally, there is a special rule (q_0, α, q_m). Intuitively, by taking a rule of the form (q_j, c_{j+1}, q_{j+1}) a follower either moves with another follower taking the same rule or the follower has heard from the leader that the clause c_{j+1} becomes true because of the choices that the leader has made. Further, if a follower takes the special rule (q_0, α, q_m), then either that follower is moving with another follower taking the same rule or the follower is helping the leader traverse some gadget of the leader protocol.

Intuition behind the construction. We now give an intuition behind the construction of the protocol. By a straightforward argument, we can show that there *always exists* an even number e such that C^e_{init} can reach C^e_{fin} in this protocol. Hence, the interesting part of the proof is to show that the formula φ is satisfiable if and only if there exists an *odd* number o such that C^o_{init} can reach C^o_{fin}. Now, assume that we have an odd number o of followers which start at q_0 and the leader starts at p_0. By construction, whenever the leader traverses a gadget, two followers would have to take the special rule (q_0, α, q_m) in order for the leader to take the two rules associated with the letter α in each gadget. Further, in addition to this, if at any given point in time, a follower takes the special rule (q_0, α, q_m) then another follower would be forced to take the same rule at that point in time as well. Hence, it follows that if C^o_{init} can reach C^o_{fin}, then only an even number of followers along this run can use the special rule (q_0, α, q_m). Since all the followers must reach q_m at the end, it must be the case that at least an odd number of followers go from q_0 to q_m via the "long" path, i.e., using the rules $(q_0, c_1, q_1), (q_1, c_2, q_2), \ldots, (q_{m-1}, c_m, q_m)$.

Now, for each rule (q_i, c_i, q_{i+1}), either a follower can "pair up" with another follower and take this rule or a follower can wait for the leader to take any one of her rules corresponding to the letter c_i. Since an odd number of followers take this long path, not all of them can pair up with another follower. Hence, it must be the case that for each c_i, at least one follower waits for the leader to take a rule corresponding to c_i. Since this can happen only if the leader has made a sequence of choices for the variables $x_1, \dots, x_m$ which make the clause c_i true, it would then follow that C^o_{init} can reach C^o_{fin} if and only if the formula φ is satisfiable.

Proof of correctness of the reduction. We claim that there is a cut-off for this symmetric leader protocol if and only if the formula φ is satisfiable. First, note that from the configuration C^{2n}_{init} we can reach C^{2n}_{fin}. Indeed, the following run, which only uses rules corresponding to the letter α, is a valid run from C^{2n}_{init} to C^{2n}_{fin} : $\wr p_0,\ 2n \cdot q_0 \int \to \wr \top_1,\ (2n-1)\cdot q_0,\ q_m \int \to\ \wr p_1,\ (2n-2)\cdot q_0,\ 2\cdot q_m \int \to \wr \top_2,\ (2n-3)\cdot q_0,\ 3\cdot q_m \int \to \wr p_2,\ (2n-4)\cdot q_0,\ 4\cdot q_m \int \to \cdots \to \wr \top_{n-1},\ q_0,\ (2n-1)\cdot q_m \int \to \wr p_n,\ 2n\cdot q_m \int$.

Therefore, for our reduction to be correct, by Proposition 8.4, it suffices to prove that there exists k such that there is a run from C^{2k+1}_{init} to C^{2k+1}_{fin} if and only if the formula φ is satisfiable.

($\Leftarrow$): Suppose φ is satisfiable. Let T be a satisfying assignment of φ. Let $var(x_j)$ be $\top_j$ if $T(x_j)$ is true and let it be $\bot_j$ otherwise. Let $k := m + n$. We now construct a run from C^{2k+1}_{init} to C^{2k+1}_{fin}. By construction, the follower protocol does not have any bad states. Hence, by Proposition 8.7 and the monotonicity property, from C^{2k+1}_{init} we can reach the configuration $\texttt{First} := C^{2n+1}_{init} + \sum_{1\le i\le m} \wr 2\cdot q_i \int$. Hence, it suffices to construct a run from $\texttt{First}$ to C^{2k+1}_{fin}.

Let U be the empty set. From the configuration $\texttt{First}$, we construct a run which moves the leader from p_0 to p_n. To begin with, the leader is at the state p_0 in the configuration $\texttt{First}$. Now, suppose we are currently in some configuration where the leader is at the state p_j for some index j and we want the leader to go to the state p_{j+1}. First, we move the leader from p_j to $var(x_{j+1})$ and a single follower from q_0 to q_m by using the rules $(p_j, \alpha, var(x_{j+1}))$ and (q_0, α, q_m) respectively. Then as long as the leader is at the state $var(x_{j+1})$, we do the following:

- If there is a rule $(var(x_{j+1}), c_k, var(x_{j+1}))$ such that $c_k \notin U$, then we keep the leader at $var(x_{j+1})$ by using the rule $(var(x_{j+1}), c_k, var(x_{j+1}))$ and move one follower from q_{k-1} to q_k by using the rule (q_{k-1}, c_k, q_k). Then, we add the letter c_k to the set U.
- If there is no such rule, then we move the leader from $var(x_{j+1})$ to p_{j+1} by using the rule $(var(x_{j+1}), \alpha, p_{j+1})$ and move one follower from q_0 to q_m by using the rule (q_0, α, q_m).

Let us analyze this run a bit more closely. Fix some letter c_k. Notice that the number of times a rule corresponding to the message c_k is fired by the leader is at most once. Indeed, if we consider the first point in our run when such a rule is fired, a necessary precondition for that to happen is that c_k is not present in the contents of U at that point. However, once we fire such a rule, we immediately add c_k to our set U, thereby preventing the firing of any such rule in the future.

Further, since T is a satisfying assignment to φ, there must be a smallest index j such that setting the value of the variable x_j to $T(x_j)$ satisfies the clause C_k. By construction of the gadgets of the leader protocol, this means that when the leader moves to the state

$var(x_j)$ along this run, the letter c_k is not present in the contents of U at that time and there is a rule $(var(x_j), c_k, var(x_j))$ in the leader protocol. According to our construction of the run, this means that the leader fires the rule $(var(x_j), c_k, var(x_j))$. Hence, it follows that for every $1 \leq k \leq m$, a leader rule corresponding to the letter c_k is fired exactly once during this run.

With this observation, let us examine the effect of this run on the followers. For each $0 \leq j \leq n-1$, for the leader to traverse the gadget from p_j to p_{j+1}, exactly two followers fire the rule (q_0, α, q_m). In addition to this, we know that for each c_k, the leader fires exactly one rule corresponding to c_k during the run. Whenever, the leader fires such a rule, the run also forces a follower to fire the rule (q_{k-1}, c_k, q_k). Hence, it follows that for each k, exactly one follower moves from q_{k-1} to q_k. Therefore, at the end of this run, we would have reached the configuration $\texttt{Second} := C_{fin}^{2n+1} + \sum_{1 \leq i \leq m} \wr 2 \cdot q_i \int$. Using Proposition 8.7 and the monotonicity property, from $\texttt{Second}$ we can reach C_{fin}^{2k+1}, thereby completing the construction of the required run.

($\Rightarrow$): Suppose there exists k such that there is a run ρ from C_{init}^{2k+1} to C_{fin}^{2k+1}. First, observe that for every $1 \leq j \leq n$, by construction of the protocol, the leader has to visit exactly one of the states $\top_j$ or $\bot_j$ along this run. To be more precise, there is exactly one state $s_j \in \{\top_j, \bot_j\}$ for which there exists a configuration C_j along this run such that $C_j(s_j) > 0$. This suggests a natural assignment T for the variables of φ: Set $T(x_j)$ to true if $s_j = \top_j$ and otherwise, set $T(x_j)$ to false. The rest of the proof is devoted to proving that T is a satisfying assignment of φ. To prove this, we will actually show that for every $1 \leq j \leq m$, there is an occurrence of a leader rule corresponding to the message c_j at some point along the run ρ. By construction of our assignment T and the leader protocol, this would immediately imply that T is a satisfying assignment for φ.

Let $\rho := C_{init}^{2k+1} \xRightarrow{r_1, r'_1} C_1 \xRightarrow{r_2, r'_2} C_2 \ldots \xRightarrow{r_w, r'_w} C_{fin}^{2k+1}$. For every $0 \leq j \leq n-1$, let $R_\alpha^j = \{(p_j, \alpha, \top_j), (p_j, \alpha, \bot_j), (\top_j, \alpha, p_{j+1}), (\bot_j, \alpha, p_{j+1})\}$. We claim that for every j, there are exactly two occurrences of rules from R_α^j in ρ. First, only at most two rules of R_α^j can appear in ρ, since otherwise it would mean that the leader visited both $\top_j$ and $\bot_j$, which we have already established as a contradiction. Further, for the leader to move out of the state p_j and go to p_n, by construction of the protocol, at least two of these rules must be fired. It then follows that if we set $R_\alpha := \bigcup_{0 \leq j \leq n-1} R_\alpha^j$, then there are exactly $2n$ occurrences of rules from R_α in ρ.

Let us now observe the parity of the number of occurrences of the rule $r := (q_0, \alpha, q_m)$ along ρ. Notice that for every i, the rule r can appear exactly once in the pair (r_i, r'_i) if and only if the other rule in the pair belongs to R_α. Since the number of occurrences of rules from R_α is exactly $2n$, it follows that the number of occurrences of the rule r in ρ is even. Since the initial configuration has an odd number of follower agents in q_0, it follows that the number of occurrences of the rule $r_{c_1} := (q_0, c_1, q_1)$ is odd and hence non-zero. By definition of the run ρ and the follower protocol, for every i, the number of times the rule $r_{c_i} := (q_{i-1}, c_i, q_i)$ occurs in ρ must be equal to the number of times the rule $r_{c_{i+1}} := (q_i, c_{i+1}, q_{i+1})$ occurs in ρ. It then follows that the all the rules in the set $\{r_{c_i} : 1 \leq i \leq m\}$ occur an odd number of times in ρ.

Let us fix some $1 \leq j \leq m$ and let us consider the rule r_{c_j}. Since it appears an odd number of times in ρ, there must be an index i such that exactly one rule in the pair (r_i, r'_i)

is r_{c_j}. The only possibility for the other rule in that pair is a leader rule of the form (p, c_j, p) for some state p. Hence, we have shown that there is at least one occurrence of a leader rule corresponding to the message c_j along the run ρ. Since j was an arbitrary number between 1 and m, the proof is complete. □

9. Conclusion

We have shown that the cut-off problem for Petri nets and rendez-vous protocols is P-complete. For the special case of symmetric rendez-vous protocols we have proved that the cut-off problem is in NC for the leaderless case and is NP-complete in the presence of a leader. Further, we have also studied the bounded-loss cut-off problem and shown that it is P-complete and NL-complete for leaderless rendez-vous and leaderless symmetric protocols respectively. Many of these results follow from two lemmas, the Scaling and Insertion lemmas, which we believe might be of independent interest. As future work, it might be worth studying other variants of the cut-off problem dealing with different types of properties such as liveness specifications.

References

[AAD+06] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Comput.*, 18(4):235–253, 2006. doi:10.1007/s00446-005-0138-3.

[AD98] Hassane Alla and René David. Continuous and hybrid Petri nets. *J. Circuits Syst. Comput.*, 8(1):159–188, 1998. doi:10.1142/S0218126698000079.

[BER21] A. R. Balasubramanian, Javier Esparza, and Mikhail A. Raskin. Finding cut-offs in leaderless rendez-vous protocols is easy. In Stefan Kiefer and Christine Tasson, editors, *Foundations of Software Science and Computation Structures - 24th International Conference, FOSSACS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings*, volume 12650 of *Lecture Notes in Computer Science*, pages 42–61. Springer, 2021. doi:10.1007/978-3-030-71995-1_3.

[BFHH17] Michael Blondin, Alain Finkel, Christoph Haase, and Serge Haddad. The logical view on continuous Petri nets. *ACM Trans. Comput. Log.*, 18(3):24:1–24:28, 2017. doi:10.1145/3105908.

[BJK+15] Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder. *Decidability of Parameterized Verification.* Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2015. doi:10.2200/S00658ED1V01Y201508DCT013.

[Blo20] Michael Blondin. The ABCs of Petri net reachability relaxations. *ACM SIGLOG News*, 7(3):29–43, 2020. doi:10.1145/3436980.3436984.

[BMWK09] Gérard Basler, Michele Mazzucchi, Thomas Wahl, and Daniel Kroening. Symbolic counter abstraction for concurrent software. In Ahmed Bouajjani and Oded Maler, editors, *Computer Aided Verification, 21st International Conference, CAV 2009, Grenoble, France, June 26 - July 2, 2009. Proceedings*, volume 5643 of *Lecture Notes in Computer Science*, pages 64–78. Springer, 2009. doi:10.1007/978-3-642-02658-4_9.

[CO21] Wojciech Czerwinski and Lukasz Orlikowski. Reachability in vector addition systems is Ackermann-complete. In *FOCS 2021*, pages 1229–1240. IEEE, 2021. doi:10.1109/FOCS52979.2021.00120.

[DSTZ12] Giorgio Delzanno, Arnaud Sangnier, Riccardo Traverso, and Gianluigi Zavattaro. On the complexity of parameterized reachability in reconfigurable broadcast networks. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPIcs*, pages 289–300. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.289.

[EFM99] Javier Esparza, Alain Finkel, and Richard Mayr. On the verification of broadcast protocols. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 352–359. IEEE Computer Society, 1999. doi:10.1109/LICS.1999.782630.

[EGLM17] Javier Esparza, Pierre Ganty, Jérôme Leroux, and Rupak Majumdar. Verification of population protocols. *Acta Informatica*, 54(2):191–215, 2017. doi:10.1007/s00236-016-0272-3.

[EK02] E. Allen Emerson and Vineet Kahlon. Model checking large-scale and parameterized resource allocation systems. In Joost-Pieter Katoen and Perdita Stevens, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 8th International Conference, TACAS 2002, Held as Part of the Joint European Conference on Theory and Practice of Software, ETAPS 2002, Grenoble, France, April 8-12, 2002, Proceedings*, volume 2280 of *Lecture Notes in Computer Science*, pages 251–265. Springer, 2002. doi:10.1007/3-540-46002-0_18.

[EN94] Javier Esparza and Mogens Nielsen. Decidability issues for Petri nets - a survey. *J. Inf. Process. Cybern.*, 30(3):143–160, 1994.

[Esp96] Javier Esparza. Decidability and complexity of Petri net problems - an introduction. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*, volume 1491 of *Lecture Notes in Computer Science*, pages 374–428. Springer, 1996. doi:10.1007/3-540-65306-6_20.

[Esp16] Javier Esparza. Parameterized verification of crowds of anonymous processes. In Javier Esparza, Orna Grumberg, and Salomon Sickert, editors, *Dependable Software Systems Engineering*, volume 45 of *NATO Science for Peace and Security Series - D: Information and Communication Security*, pages 59–71. IOS Press, 2016. doi:10.3233/978-1-61499-627-9-59.

[FH15] Estíbaliz Fraca and Serge Haddad. Complexity analysis of continuous Petri nets. *Fundam. Informaticae*, 137(1):1–28, 2015. doi:10.3233/FI-2015-1168.

[GKS+14] Annu Gmeiner, Igor Konnov, Ulrich Schmid, Helmut Veith, and Josef Widder. Tutorial on parameterized model checking of fault-tolerant distributed algorithms. In Marco Bernardo, Ferruccio Damiani, Reiner Hähnle, Einar Broch Johnsen, and Ina Schaefer, editors, *Formal Methods for Executable Software Models - 14th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2014, Bertinoro, Italy, June 16-20, 2014, Advanced Lectures*, volume 8483 of *Lecture Notes in Computer Science*, pages 122–171. Springer, 2014. doi:10.1007/978-3-319-07317-0_4.

[GS92] Steven M. German and A. Prasad Sistla. Reasoning about systems with many processes. *J. ACM*, 39(3):675–735, 1992. doi:10.1145/146637.146681.

[HS20] Florian Horn and Arnaud Sangnier. Deciding the existence of cut-off in parameterized rendez-vous networks. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPIcs*, pages 46:1–46:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CONCUR.2020.46.

[KB79] Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979. doi:10.1137/0208040.

[KKW10] Alexander Kaiser, Daniel Kroening, and Thomas Wahl. Dynamic cutoff detection in parameterized concurrent programs. In Tayssir Touili, Byron Cook, and Paul B. Jackson, editors, *Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, volume 6174 of *Lecture Notes in Computer Science*, pages 645–659. Springer, 2010. doi:10.1007/978-3-642-14295-6_55.

[KM78] Ravindran Kannan and Clyde L. Monma. On the computational complexity of integer programming problems. In Rudolf Henn, Bernhard Korte, and Werner Oettli, editors, *Optimization and Operations Research*, pages 161–172, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg. doi:10.1007/978-3-642-95322-4_17.

[Lad75] Richard E. Ladner. The circuit value problem is log space complete for P. *SIGACT News*, 7(1):18–20, 1975. doi:10.1145/990518.990519.

[Las22] Slawomir Lasota. Improved Ackermannian lower bound for the Petri nets reachability problem. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual*

Conference), volume 219 of *LIPIcs*, pages 46:1–46:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.STACS.2022.46.

[Ler21] Jérôme Leroux. The reachability problem for Petri nets is not primitive recursive. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1241–1252. IEEE, 2021. doi:10.1109/FOCS52979.2021.00121.

[Mul87] Ketan Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Comb.*, 7(1):101–104, 1987. doi:10.1007/BF02579205.

[Mur89] Tadao Murata. Petri nets: Properties, analysis and applications. *Proc. IEEE*, 77(4):541–580, 1989. doi:10.1109/5.24143.

[NB15] Saket Navlakha and Ziv Bar-Joseph. Distributed information processing in biological and computational systems. *Commun. ACM*, 58(1):94–102, 2015. doi:10.1145/2678280.

[Pap81] Christos H. Papadimitriou. On the complexity of integer programming. *J. ACM*, 28(4):765–768, 1981. doi:10.1145/322276.322287.

[Pap07] Christos H. Papadimitriou. *Computational complexity.* Academic Internet Publ., 2007.

[PXZ02] Amir Pnueli, Jessie Xu, and Lenore D. Zuck. Liveness with (0, 1, ∞)-counter abstraction. In Ed Brinksma and Kim Guldstrand Larsen, editors, *Computer Aided Verification, 14th International Conference, CAV 2002,Copenhagen, Denmark, July 27-31, 2002, Proceedings*, volume 2404 of *Lecture Notes in Computer Science*, pages 107–122. Springer, 2002. doi:10.1007/3-540-45657-0_9.

[PZ89] Michael E. Pohst and Hans Zassenhaus. *Algorithmic algebraic number theory*, volume 30 of *Encyclopedia of mathematics and its applications.* Cambridge University Press, 1989.

[RHS10] Laura Recalde, Serge Haddad, and Manuel Silva Suárez. Continuous Petri nets: Expressive power and decidability issues. *Int. J. Found. Comput. Sci.*, 21(2):235–256, 2010. doi:10.1142/S0129054110007222.

[SCWB08] David Soloveichik, Matthew Cook, Erik Winfree, and Jehoshua Bruck. Computation with finite stochastic chemical reaction networks. *Nat. Comput.*, 7(4):615–633, 2008. doi:10.1007/s11047-008-9067-y.

[Son85] Eduardo D. Sontag. Real addition and the polynomial hierarchy. *Inf. Process. Lett.*, 20(3):115–120, 1985. doi:10.1016/0020-0190(85)90076-6.