

# Reachability in multi-agent transfer systems <sup>★</sup>

Nathalie Bertrand<sup>1[0000–0002–9957–5394]</sup>, Loïc Hélouët<sup>1[0000–0001–7056–2672]</sup>,  
Engel Lefaucheux<sup>2[0000–0003–0875–300X]</sup>, and Luca  
Paparazzo<sup>1[0009–0007–1134–000X]</sup>

<sup>1</sup> University of Rennes, Inria, CNRS, Rennes, France

<sup>2</sup> Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

{nathalie.bertrand, loic.helouet, engel.lefaucheux, luca.paparazzo}@inria.fr

**Abstract.** This paper introduces collaborative reachability games with energy constraints. In the considered arenas, agents can spend or gain energy during moves, or share it with their peers if their current position allows it. We study several variants of energy reachability games where agents move either synchronously or asynchronously, and with/without constraints on energy transfers among peers. We show that these problems have different complexities ranging from NP to EXPSPACE.

**Keywords:** Multi-Agent Systems · Quantitative verification · VASS · Collaborative games · Planning

## 1 Introduction

Cooperation of several agents occurs in a variety of applications, such as robotics, traffic control and aviation to name a few. In contrast to adversarial games, in such cooperative settings, the agents collaborate to achieve a common goal. A typical instantiation of this general framework is the multi-agent path finding problem [11], in which one aims at designing a plan to move multiple agents while avoiding collisions to perform a global task. Beyond Boolean objectives such as coverage of an area, or reachability of a position for each agent, introducing quantities in models for multi-agent systems is crucial to represent energy or financial cost. Quantitative settings where multiple agents interact are for instance useful formalisms to find optimal management strategies to control cyber physical systems (CPS) where objectives are not purely Boolean, but also aim at optimizing some measure. A variety of settings of quantitative multi-player games have been proposed in the literature [4,5,6,12]. Game concepts such as the famous Nash equilibria [20] can then be studied, for instance to efficiently distribute energy in smart grids [5].

In this paper, we introduce a new quantitative multi-agent model, in which agents move on their own local arena and are given a goal, *i.e.*, a particular

---

<sup>★</sup> This work was supported by the project BisoUS (ANR-22-CE48-0012).

The authors would like to thank an anonymous reviewer for their constructive comments on potential links between our model and VASS.

vertex to reach. Local arenas are equipped with integer weights on edges to represent energy variations. Each agent stores energy and, when moving from a vertex to a consecutive one, gains energy if the weight is positive, or loses energy if the weight is negative. Interestingly, agents may cooperate by sending some or all of their stored energy to other agents. The objective is to design a collaborative plan moving each agent to its target vertex while staying within the energy available in the system, possibly using transfers among peers. We coin this model *multi-agent transfer systems*, or simply *transfer systems*.

Several semantics can be considered for transfer systems: either agents move synchronously or asynchronously. In any case, they can only take an edge if their stored energy is sufficient, as their energy level cannot drop below 0. We consider the natural question of global reachability objectives, where all agents must reach their assigned target simultaneously. Our setting thus shares the objective of multi-agent path finding [11]. There are however several crucial differences: first, in transfer systems, agents move on their respective local arenas rather than on a common space; second, transfer systems are equipped with energy variations and agents must move within energy budget; finally, in transfer systems agents can transfer energy one to another.

Transfer systems form a particularly suited model for modern urban transport networks equipped with regenerative braking systems. In these CPSs, the kinetic energy of a braking vehicle can be converted into electric energy, transferred to the power network and used by other close vehicles. Another possible application is the study of the logistics of complex systems in which resources must be provided at specific locations and times for the success of a mission.

Multi-weighted energy games [10] are close to our transfer systems. In multi-weighted energy games, stored quantities are  $k$ -vectors of integers and moves are also labeled by integer vectors of same dimension. Different to our setting, the number of players is fixed to at most two. The objective in these games is to play infinitely while respecting energy bounds on each coordinate: a lower bound or a combination of lower and a weak/strong upper bounds. With a single player, the problem with a lower bound is NP-hard and  $k$ -EXPTIME already, and with two players, the complexity is EXPTIME-hard and in  $k$ -EXPTIME. These complexity proofs build on results of [3]. One can consider transfer systems with  $n$  agents as a reachability question in a multi-weighted game with a single player (representing the coalition of agents) of dimension  $n$ , one dimension for each agent that must remain non-negative. For an arbitrary dimension, existence of an infinite run in multiweighted games with lower bounds is EXPSPACE-complete, and becomes PSPACE-complete if integral upper bound are set for each dimension. Notice that this setting has several differences with our questions in transfer systems; one of the main differences is that [10] considers the existence of infinite runs, while the questions addressed in this paper would be encoded as coverability questions. Most importantly, transfer systems are given succinctly by local arenas for each agent, while multi-weighted energy games are monolithic.

As our model deals with transfer of energy, and is close to Petri nets, a natural question is whether reachability in transfer systems is equivalent to a

reachability or coverability in transfer Petri nets [8]. Transfer Petri nets extend Petri nets with flow relations that can transfer the whole contents of a place  $p$  to another place  $p'$  when firing a transition. Our complexity results on transfer systems prove that reachability for transfer systems and coverability/reachability for transfer nets are different questions. Indeed, transfer Petri nets can easily simulate Reset Petri nets a model where reachability is undecidable [1], and coverability is Ackermann-hard [23]. In contrast, our reachability problems on transfer systems remain decidable in almost all cases, and have at worst complexities in **EXPSPACE** when decidable. From a modeling perspective, transfers in Petri nets and in transfer systems are quite different: in Petri nets the whole contents of a place is transferred in one step while in our model, an agent can share only a part of its energy.

The semantics of transfer systems can be captured by vector addition systems with states (VASS) [14], or equivalently by Petri nets, and our reachability problems as coverability questions. **EXPSPACE**-hardness for coverability in VASS was shown by [18], and the matching **EXPSPACE** upper bound was shown by [21]. A natural question is whether one has to pay the full complexity of VASS to solve our reachability problems in transfer systems. We show in this paper that the answer depends on the chosen characteristics of the model. For instance, reachability for transfer systems with energy transfers always enabled and under asynchronous semantics lies between **NP** and **PSPACE**. Also, under asynchronous semantics with arbitrary transfer groups, the complexity lies between **PSPACE** and **EXPSPACE**. Finally, if one relaxes synchronicity by allowing agents that lack energy to idle (resulting in the so-called weak synchronous semantics), reachability becomes undecidable.

The rest of the paper is organized as follows. Section 2 presents the model and the notations that will be used throughout the paper. Section 3 details the different possible semantics of the model: asynchronous, strongly synchronous, and weakly synchronous and shows the relations between these semantics. Section 4 studies the complexity of reachability under all semantics when transfer of energy can occur at any time between agents. Section 5 considers reachability for systems with restricted local transfers, that can occur only in some states. Due to space constraints, some proofs are omitted and can be found in an extended version of this paper available at [2].

## 2 Transfer systems

Transfer systems are multi-agent systems, in which every agent plays on a local weighted graph, and the communication between agents is limited.

**Definition 1.** *A local arena  $A = (V, E)$  is a directed weighted graph where  $V$  is a finite set of vertices,  $E \subseteq V \times \mathbb{Z} \times V$  describes the edges of the arena.*

Intuitively, the weight on an edge represents the amount of energy an agent gains (if positive) or loses (if negative) while traversing that edge. Communication between agents is limited to energy transfers, and is formalised by transfer groups that specify conditions on the vertices of the agents to enable transfers.

**Definition 2.** Let  $n \in \mathbb{N}$ ,  $\mathcal{A} = \{A_1, \dots, A_n\}$  be a set of local arenas with  $A_i = (V_i, E_i)$  for every  $i \in \llbracket 1, n \rrbracket$  (assuming all  $V_i$  are disjoint sets) and  $\mathcal{T} : \bigcup_{i \in \llbracket 1..n \rrbracket} V_i \rightarrow \mathbb{N}$  is a partial map defining transfer groups.  
 $\mathcal{A}$  and  $\mathcal{T}$  induce the transfer system  $\text{TS} = \langle \mathcal{A}, \mathcal{T} \rangle$ .

We will say that vertices  $v, v'$  belong to the same transfer group if  $\mathcal{T}(v) = \mathcal{T}(v')$ , and impose that transfer groups are not singletons, are disjoint sets and contain states from at least two arenas. For convenience, we will often define transfer groups as sets of states  $T_1, \dots, T_k$  where  $T_i = \{v \mid \mathcal{T}(v) = i\}$ . Writing  $V = \bigcup_{i \in \llbracket 1..n \rrbracket} V_i$ , the size of a transfer arena is defined as  $|\text{TS}| = |V| \cdot (|\mathcal{T}| + 1) + |V|^2 \cdot \log(w_{\max})$  where  $w_{\max}$  is the largest absolute value of a weight appearing in an arena, and  $|\mathcal{T}|$  is bounded  $|V| \cdot \log(|V|)$ .

The semantics of a transfer system  $\text{TS} = \langle \mathcal{A}, \mathcal{T} \rangle$  is given in terms of a transition system. A *configuration* of  $\text{TS}$  consists of the current vertex of each agent and their energy level: we write  $C, C'$ , etc. for a configuration, and  $\Gamma = (\prod_{i=1}^n V_i) \times \mathbb{N}^n$  for the set of all configurations. For a configuration  $C = (S, \vec{e})$ ,  $S \in \prod_{i=1}^n V_i$  is referred to as the *global state* (or simply state) and  $\vec{e}$  as the *energy vector*. When the dimension  $n$  is clear from the context, we use  $\vec{0}$  to denote the null energy vector  $(0, \dots, 0) \in \mathbb{N}^n$ .

Transitions between configurations are induced by moves of the agents on their local arenas, or energy transfers between agents when permitted by the transfer groups. An agent  $A_i$  cannot move along an edge  $q_i \xrightarrow{-w} q'_i$  with negative weight  $-w$  if its energy level  $e_i$  is lower than  $w$ . We will say that edge  $q_i \xrightarrow{w} q'_i$  is *enabled* if  $e_i + w \geq 0$ . For move transitions, we distinguish several semantics, depending on whether the agents move simultaneously or not.

**Definition 3.** Consider two configurations  $C = \langle (q_1, \dots, q_n), (e_1, \dots, e_n) \rangle$  and  $C' = \langle (q'_1, \dots, q'_n), (e'_1, \dots, e'_n) \rangle$ .

**move** There is a move transition from  $C$  to  $C'$  if one of the following holds

**asynchronous**  $\exists i \in \llbracket 1, n \rrbracket : q_i \xrightarrow{w} q'_i \in E_i$ ,  $e'_i = e_i + w \geq 0$  and  $\forall j \neq i, (q'_j, e'_j) = (q_j, e_j)$ , corresponding to the single agent  $A_i$  moving along an edge of its local arena. This results in an asynchronous move transition, and is denoted  $C \xrightarrow{a} C'$ .

**synchronous**  $\forall i \in \llbracket 1, n \rrbracket$ ,  $q_i \xrightarrow{w_i} q'_i \in E_i$  and  $e'_i = e_i + w_i \geq 0$ , corresponding to all agents moving simultaneously in their respective local arenas. This results in a strongly synchronous move transition, denoted  $C \xrightarrow{s} C'$ .

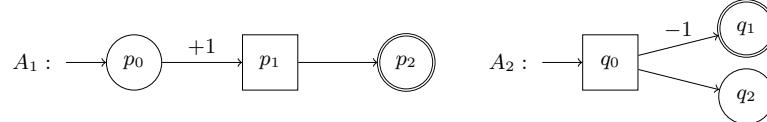
**weakly synchronous**  $\forall i \in \llbracket 1, n \rrbracket$ , either  $q_i \xrightarrow{w_i} q'_i \in E_i$  and  $e'_i = e_i + w_i \geq 0$  or  $(q'_i, e'_i) = (q_i, e_i)$  and  $\forall q_i \xrightarrow{w_i} q''_i \in E_i, e_i + w_i < 0$ , corresponding to a synchronous move of all agents that have an enabled edge. This results in a weakly synchronous move transition, denoted  $C \xrightarrow{w} C'$ .

**transfer** There is a transfer transition from  $C$  to  $C'$  if  $\forall i, q_i = q'_i$ , and  $\exists i, j \in \llbracket 1, n \rrbracket$ ,  $\mathcal{T}(q_i) = \mathcal{T}(q_j)$ ,  $e_i + e_j = e'_i + e'_j$  and  $\forall k \notin \{i, j\}, e_k = e'_k$ , corresponding to a transfer between agents  $A_i$  and  $A_j$  on vertices of a same transfer group. This transition is denoted  $C \xrightarrow{t} C'$ .

We use  $\rightarrow^a$  (resp.  $\rightarrow^s$ , resp.  $\rightarrow^w$ ) to denote a transition that is either a transfer or an asynchronous (resp. strongly synchronous, resp. weakly synchronous) move and call it an asynchronous (resp. strongly synchronous, resp. weakly synchronous) transition for short. For instance  $\rightarrow^a = \rightarrow_m^a \cup \rightarrow_t$ . We also refer to any type of move transition with  $\rightarrow_m$ :  $\rightarrow_m = \rightarrow_m^a \cup \rightarrow_m^s \cup \rightarrow_m^w$ . Finally, an arbitrary transition is simply denoted  $\rightarrow$ . Notice that agents change their local vertex in their arena during moves, and stay on the same vertex during transfers.

As usual, sequences of transitions define runs of the transfer system. A finite/infinite *asynchronous run* (resp. strongly synchronous run, resp. weakly synchronous run) over  $\text{TS}$  is a finite/infinite sequence of asynchronous (resp. strongly synchronous, resp. weakly synchronous) transitions. We will write  $C \rightsquigarrow^a C'$  (resp.  $C \rightsquigarrow^s C'$ ,  $C \rightsquigarrow^w C'$ ) when there exists an asynchronous (resp. synchronous, weakly synchronous) run from  $C$  to  $C'$ . The set of asynchronous (resp. strongly synchronous, resp. weakly synchronous) runs over  $\text{TS}$  is denoted  $\text{Runs}^a(\text{TS})$  (resp.  $\text{Runs}^s(\text{TS})$ , resp.  $\text{Runs}^w(\text{TS})$ ). We refer to them as the asynchronous, strongly synchronous and weakly synchronous semantics of the transfer system, respectively, that we sometimes abbreviate into  $a$ -semantics,  $s$ -semantics and  $w$ -semantics.

We observe the following relations between runs of transfer systems under the various semantics. First of all,  $\text{Runs}^s(\text{TS}) \subseteq \text{Runs}^w(\text{TS})$ . Indeed, by definition, for every two configurations  $C, C'$ , if  $C \rightarrow_m^s C'$  then  $C \rightarrow_m^w C'$ . One can also notice that if  $C \rightarrow_m^w C'$ , then there exists a sequence of move transitions  $C \rightarrow_m^a C_1 \rightarrow_m^a \dots \rightarrow_m^a C'$ . Thus, a run in the weakly synchronous semantics can be simulated by a run in the asynchronous semantics.

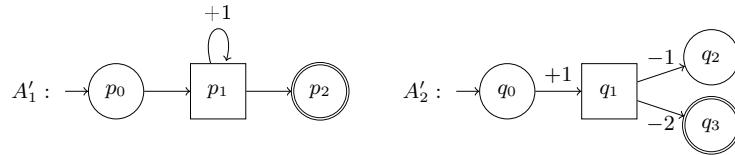


**Fig. 1.** Transfer system with two agents and a single transfer group with  $p_1$  and  $q_0$ :  $\mathcal{T} = \{\{p_1, q_0\}\}$ . Null weights are omitted.

*Example 1.* Consider the transfer system with two agents depicted in Figure 1, where boxed vertices belong to the same transfer group. Let  $C_0 = \langle (p_0, q_0), (0, 0) \rangle$  be the initial configuration. Then, there exists an asynchronous run from  $C_0$  to the target state  $(p_2, q_1)$ , namely  $\langle (p_0, q_0), (0, 0) \rangle \rightarrow_m^a \langle (p_1, q_0), (1, 0) \rangle \rightarrow_t \langle (p_1, q_0), (0, 1) \rangle \rightarrow_m^a \langle (p_1, q_1), (0, 0) \rangle \rightarrow_m^a \langle (p_2, q_1), (0, 0) \rangle$ . Yet, there are no strongly nor weakly synchronous runs to the target.

Consider now the transfer system depicted in Figure 2. Again, there exists an asynchronous execution that reaches the target state:  $\langle (p_0, q_0), (0, 0) \rangle \rightarrow_m^a \langle (p_1, q_0), (0, 0) \rangle \rightarrow_m^a \langle (p_1, q_0), (1, 0) \rangle \rightarrow_m^a \langle (p_1, q_1), (1, 1) \rangle \rightarrow_t \langle (p_1, q_1), (0, 2) \rangle$

$\longrightarrow_m^a \langle (p_1, q_3), (0, 0) \rangle \longrightarrow_m^a \langle (p_2, q_3), (0, 0) \rangle$ . Also, since the only transition fireable from  $(q_1, 1)$  for  $A'_2$  leads to  $q_2$ , there is no synchronous run to  $(p_2, q_3)$ . Finally, even though  $A'_2$  has a move transition available from  $(q_1, 1)$  leading to  $q_2$ , there exists a weakly synchronous run that avoids the sink state  $q_2$  and reaches the global target state:  $\langle (p_0, q_0), (0, 0) \rangle \longrightarrow_m^w \langle (p_1, q_1), (0, 1) \rangle \longrightarrow_t \langle (p_1, q_1), (1, 0) \rangle \longrightarrow_m^w \langle (p_1, q_1), (2, 0) \rangle \longrightarrow_t \langle (p_1, q_1), (0, 2) \rangle \longrightarrow_m^w \langle (p_2, q_3), (0, 0) \rangle$ . Interestingly, in  $q_1$  it is in  $A'_2$ 's interest to send energy to  $A'_1$ , thus not being able to fire any move transition while waiting for  $A'_1$  to accumulate enough energy and send it back so that both reach their target.



**Fig. 2.** Transfer system with two agents and a single transfer group with  $p_1$  and  $q_1$ :  $\mathcal{T} = \{\{p_1, q_1\}\}$ . Null weights are omitted.

*Reachability in transfer systems* The transfer system is equipped with one reachability goal for each agent, given by a set of initial vertices and a set of final vertices. The global objective is then to find a run in which each agent reaches its final vertex, starting from its initial vertex and with initial energy 0. Similarly to vector addition systems with states (VASS), recall that the energy level of agents cannot drop below 0. As agents can reach their final vertices with an arbitrary energy level, we naturally introduce a coverability relation on configurations.

For two configurations  $C = \langle S, (e_1, \dots, e_n) \rangle$  and  $C' = \langle S', (e'_1, \dots, e'_n) \rangle$ , we say  $C'$  covers  $C$ , written  $C \triangleleft C'$ , if  $S = S'$  and  $\forall i \in [1, n], e_i \leq e'_i$ . For any global state  $S$  we define the *covering* of  $S$  as  $S\uparrow = \{C \mid \langle S, \vec{0} \rangle \triangleleft C\}$ . We are now ready to define the verification problems of interest for transfer systems:

Problem  $x$ -REACH

**Input:** A transfer system  $\text{TS}$ , an initial state  $S_0$  and a final state  $S_f$

**Question:** Does there exist  $C_f \in S_f\uparrow$  and a run  $\rho : \langle S_0, \vec{0} \rangle \rightsquigarrow^x C_f$ ?

Note that this defines three decision problems, when one varies the semantics (parameter  $x$ ): asynchronous, strongly synchronous or weakly synchronous. Moreover, we consider arbitrary transfer groups, as well as the special case of a unique trivial transfer group  $T_{\top} = \bigcup_{i=1}^n V_i$ . We later use  $\ell x$ -REACH and  $ux$ -REACH to highlight the *transfer group type* and respectively denote the variant with arbitrary transfer groups or a unique trivial transfer group.

*Example 2.* Back to the example of Figures 1 and 2,  $\langle (A_1, A_2), \{\{p_1, q_0\}\} \rangle$  is a positive instance of  $\ell a$ -REACH, and  $\langle (A'_1, A'_2), \{T_{\top}\} \rangle$  is a positive instance of

*uw*-REACH, but there exists only a single positive run for *uw*-REACH which is the same as for *lw*-REACH.

In the rest of the paper, we study the complexity of all variants of the *tx*-REACH problem. The following table summarizes the obtained results:

	semantics		
transfer	asynchronous	strongly synchronous	weakly synchronous
unique group	NP-hard (Th. 3) in PSPACE (Cor. 1)		PSPACE-c. (Th. 4 & Th. 5)
arbitrary groups	PSPACE-c. (Th. 6 & Th. 7)	PSPACE-hard (Cor. 3) in EXPSPACE (Th. 8)	undecidable (Th. 9)

### 3 Relationships between the different semantics

A first, immediate, observation is that the unique variants of our decision problem are special cases of the local ones. Indeed, any instance of a *ux*-REACH with a single trivial transfer group is also an instance of *lx*-REACH. There is thus an immediate polynomial reduction from one to the other:

**Proposition 1.** *For every semantics  $x \in \{a, s, w\}$ ,  $ux$ -REACH  $\preceq_P lx$ -REACH.*

The following theorems relate to the asynchronous, strongly synchronous and weakly synchronous semantics (for a fixed transfer group type):

**Theorem 1.** *For every transfer group type  $t \in \{u, \ell\}$ ,  $ta$ -REACH  $\preceq_P ts$ -REACH.*

*Proof.* Let  $\mathbf{TS} = \langle \{A_1, \dots, A_n\}, \mathcal{T} \rangle$  be a transfer system, and  $S_0, S_f$  initial and final global states. From  $\mathbf{TS}$ , we build the transfer system  $\mathbf{TS}'$  in which every vertex of every agent is added a self-loop with weight 0. Formally,  $\mathbf{TS}' = \langle \{B_1, \dots, B_n\}, \mathcal{T} \rangle$  where for every  $i \in \llbracket 1, n \rrbracket$ , if  $A_i = (V_i, E_i)$  then  $B_i = (V_i, F_i)$  with  $F_i = E_i \cup \{q \xrightarrow{0} q \mid q \in V_i\}$ . We claim that

$$\exists \langle S_0, \vec{0} \rangle \rightsquigarrow_{\mathbf{TS}}^a \langle S_f, \vec{e} \rangle \in \text{Runs}^a(\mathbf{TS}) \quad \text{iff} \quad \exists \langle S_0, \vec{0} \rangle \rightsquigarrow_{\mathbf{TS}'}^s \langle S_f, \vec{e} \rangle \in \text{Runs}^s(\mathbf{TS}').$$

Note that the difference between asynchronous and strongly synchronous semantics only lies in move transitions (and do not concern transfer transitions). Intuitively, the 0-self-loops in  $\mathbf{TS}'$  are used to simulate an asynchronous run over  $\mathbf{TS}$  by a synchronous run over  $\mathbf{TS}'$ . Reciprocally, synchronous transitions over  $\mathbf{TS}'$  can be serialized (and 0-self-loops can be removed) to obtain an asynchronous run over  $\mathbf{TS}$ .  $\square$

**Theorem 2.** *For every transfer group type  $t \in \{u, \ell\}$ ,  $ts$ -REACH  $\preceq_P tw$ -REACH.*

*Proof.* Let  $\mathbf{TS} = \langle \{A_1, \dots, A_n\}, \mathcal{T} \rangle$  be a transfer system, and  $S_0, S_f$  initial and final global states. From  $\mathbf{TS}$ , we build the transfer system  $\mathbf{TS}'$  in which each local arena  $A_i$  is augmented with a fresh vertex  $\text{BAD}_i$  and additional edges from every vertex to  $\text{BAD}_i$  with weight 0. Formally,  $\mathbf{TS}' = \langle \{B_1, \dots, B_n\}, \mathcal{T} \rangle$

where for every  $i \in \llbracket 1, n \rrbracket$ , if  $A_i = (V_i, E_i)$  then  $B_i = (V_i \cup \{\text{BAD}_i\}, F_i)$  with  $F_i = E_i \cup \{q \xrightarrow{0} \text{BAD}_i \mid q \in V_i \cup \{\text{BAD}_i\}\}$ . We claim that

$$\exists \langle S_0, \vec{0} \rangle \rightsquigarrow_{\text{TS}}^s \langle S_f, \vec{e} \rangle \in \text{Runs}^s(\text{TS}) \text{ iff } \exists \langle S_0, \vec{0} \rangle \rightsquigarrow_{\text{TS}'}^w \langle S_f, \vec{e} \rangle \in \text{Runs}^w(\text{TS}').$$

Note that the difference between strongly and weakly synchronous semantics only lies in move transitions (and do not concern transfer transitions). Since the additional edges have weight 0, every agent always has an enabled edge. This means that for  $\text{TS}'$ , the strongly synchronous runs and weakly synchronous runs coincide. Moreover, every vertex  $\text{BAD}_i$  is a sink. Hence, a run reaching the global final states cannot visit  $\text{BAD}_i$ . Finally, by construction, the runs of  $\text{Runs}^w(\text{TS}')$  that avoid all vertices  $\text{BAD}_i$  are exactly the runs of  $\text{Runs}^s(\text{TS})$ .  $\square$

*Remark 1 (Transfer systems and VASS.).* In general, reachability in transfer systems can be cast into state-coverability of VASS. The state-space of the VASS is the product of sets of vertices for each agent, thus exponential in the transfer system size. The VASS transitions are induced by transfer transitions and move transitions, and their precise definition depends on the semantics of move transitions. In the case of a unique transfer group, for asynchronous and strongly synchronous semantics, 1-dim VASS even suffice, since intuitively, a unique counter is needed to store the total energy amount shared by the agents. For the weakly synchronous semantics however, it is less obvious how to represent the energy levels of agents with a single counter. Indeed, a global energy level exceeding the energies required to allow one move per agent is not a sufficient condition for all agents to move. For instance, an agent may have the incentive to transfer energy to another agent in order to be temporarilly blocked (see Examples 1 and 2). This suggests that the encoding in 1-dim VASS is not immediate for transfer systems under the weakly synchronous semantics, and that 1 dimension per agent may be nedded. Further, up to our knowledge, the state of the art on state-coverability in 1-dim VASS [13,17] yields worse complexity results than the direct proofs we present in the coming section, since the obtained VASS is exponential in the transfer system size. For arbitrary transfer groups, the situation is even worse since the reduction would be to an exponential size n-dim VASS.

## 4 Unique trivial transfer group

Let us start with the particular case of a unique and trivial transfer group:  $\text{TS} = \langle \{A_1, \dots, A_n\}, \{T_{\top}\} \rangle$ . In such transfer systems, in every configuration, agents can transfer energy to others, regardless of their respective local vertices.

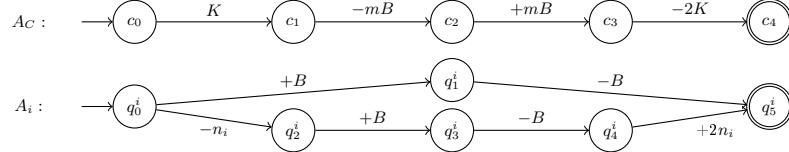
### 4.1 Asynchronous Semantics

For the asynchronous semantics, we prove the following complexity lower-bound:

**Theorem 3.** *ua-REACH is NP-hard.*

*Proof.* We perform a reduction from the SUBSETSUM problem, that we recall now. Given a finite set of integers  $\mathcal{S} = \{n_1, \dots, n_m\}$  and a target integer  $K \in \mathbb{N}$ , the subset sum problem consists in determining whether there exists a subset  $I \subseteq \llbracket 1, m \rrbracket$  such that  $\sum_{i \in I} n_i = K$ . This problem is known to be NP-complete [15].

From an instance  $\mathcal{S} = \langle \{n_1, \dots, n_m\}, K \rangle$  of SUBSETSUM, we build a transfer system  $\mathsf{TS} = \langle (A_C, A_1, \dots, A_m), \{T_\top\} \rangle$  together with initial and final states  $S_0, S_f$  as represented in Figure 3, and such that  $\mathcal{S}$  is a positive instance of SUBSETSUM iff there exists  $C_f \in S_f \uparrow$  and  $\langle S_0, \vec{0} \rangle \rightsquigarrow^a C_f$  in  $\mathsf{Runs}^a(\mathsf{TS})$ .



**Fig. 3.** Transfer system for the NP-hardness of *ua*-REACH. Incoming arrows point to initial vertices, and doubly circled vertices are final.  $B = 2mK$ .

Before giving the formal proof, let us explain the intuition of the reduction. The system is formed of one "control" agent  $A_C$ , as well as one agent  $A_i$  for each integer  $n_i$ . The transfer system starts with  $K$  energy units (gained by the controller  $A_C$ ). In a first step, the agents  $A_i$  will have to select whether or not the solution set  $I$  contains  $i$ . This choice corresponds to two branches of the local arena  $A_i$ . If they decide  $i \in I$ , they have to pay  $n_i$ , which ensures that the sum of all the values selected by the agents is at most  $K$ . After this choice, each  $A_i$  agent receives a very large value  $B$ . This is a synchronization point:  $A_C$  needs all the agents to have received  $B$  energy units before it can progress, thus ensuring that every agent made their choice. In the third step, each agent  $A_i$  which decided that  $i \in I$  receives  $2n_i$  before reaching its target. As  $A_C$  needs  $2K$  to reach its target, this will require that the sum of all the values selected by the agents is at least  $K$ . All in all, the sum has to be exactly  $K$ . Note that without the synchronization point, some agent  $A_j$  could wait for the  $2n_i$  to be produced by  $A_i$  before making their initial choice.

Let us now formally describe the reduction and establish its correctness. Without loss of generality we assume that for every  $i \in \llbracket 1, m \rrbracket$ ,  $n_i \leq K$ , and we set  $B = 2mK$ . The local arenas are defined by:

- $A_C = (\{c_i \mid i \in \llbracket 0, 4 \rrbracket\}, E)$  with  $E = \{c_j \xrightarrow{w_j} c_{j+1} \mid j \in \llbracket 0, 3 \rrbracket\}$ , and  $w_0 = K, w_1 = -mB, w_2 = mB, w_3 = -2K$ .
- for every  $i \in \llbracket 1, m \rrbracket$ ,  $A_i = (\{q_j^i \mid j \in \llbracket 0, 5 \rrbracket\}, E_i)$  with :
$$E_i = \left\{ (q_0^i \xrightarrow{w_{01}^i} q_1^i), (q_1^i \xrightarrow{w_{15}^i} q_5^i) \right\} \cup \left\{ (q_0^i \xrightarrow{w_{02}^i} q_2^i), (q_2^i \xrightarrow{w_{23}^i} q_3^i), (q_3^i \xrightarrow{w_{34}^i} q_4^i), (q_4^i \xrightarrow{w_{45}^i} q_5^i) \right\};$$
and  $w_{01}^i = w_{23}^i = B, w_{15}^i = w_{34}^i = -B, w_{02}^i = -n_i$  and  $w_{45}^i = 2n_i$ .

Since we consider *ua*-REACH, the only transfer group is  $T_{\top}$  that contains every vertex of every agent. The initial and target states are  $S_0 = (c_0, q_0^1, \dots, q_0^m)$  and  $S_f = (c_4, q_5^1, \dots, q_5^m)$ .

We claim that some  $C_f \in S_f \uparrow$  is reachable from  $\langle S_0, \overrightarrow{0} \rangle$  by an asynchronous run if and only if there exists  $I \subseteq \llbracket 1, m \rrbracket$  with  $\sum_{i \in I} n_i = K$ .

As a first step, let us show that if an agent  $A_i$  takes an edge with cost  $-B$  before  $A_C$  reaches  $c_2$ , then the controller cannot reach its target. The maximum amount of energy that can be collected in the system before  $A_C$  reaches  $c_2$  is bounded by  $mB + K + \sum_{i=1}^m 2n_i$ . Assume an edge with weight  $-B$  is taken, the total energy is now bounded by  $(m-1)B + K + \sum_{i=1}^m 2n_i$ . Due to our choice of  $B$ , this value is strictly less than  $mB$  and  $A_C$  cannot take the edge to  $c_2$ .

The previous reasoning implies that every available edge with weight  $+B$  must be taken by the agents  $A_i$  before  $A_C$  can reach  $c_2$ . Following this observation, fix a run and assume that every agent took an edge with weight  $+B$  but did not take its edge with weight  $-B$  yet; assume further that  $A_C$  has not reached  $c_2$  yet; finally, without loss of generality, assume that  $A_C$  reached  $c_1$ . For every  $i \in \llbracket 1, m \rrbracket$ , agent  $A_i$  is thus either in  $q_1^i$  or in  $q_3^i$ . Let  $H$  be defined as the set of indices  $i$  such that  $A_i$  is in  $q_3^i$ .

Assume that  $D = \sum_{i \in H} n_i > K$ , then the current total energy is  $mB + K - D < mB$ , thus  $A_C$  cannot reach  $c_2$  and will never reach its target. So  $D \leq K$ . From this point, as  $A_C$  needs to traverse to  $c_2$  not to be blocked, we can assume it goes immediately to  $c_3$ . Agent  $A_C$  and the agents  $A_i$  with  $i \notin H$  will no longer gain energy before reaching their target, so we can assume the agents in  $H$  act first. They lose an amount of energy of  $|H|B$  and gain  $2D$ . Thus, the total energy in the system is  $(m - |H|)B + 2D + (K - D)$ . Exactly  $(m - |H|)B + 2K$  energy units are required for the remaining agents to reach their target. This is only possible if  $D \geq K$ . As we already showed that  $D \leq K$ , this means that  $D = K$ .

This concludes the proof.  $\square$

Note that the hardness proof of Theorem 3 uses acyclic arenas. Moreover, given a transfer system where each local arena is acyclic, the reachability problems under each semantics is in **NP**. Indeed, the length of a path for each agent from its initial vertex to a final vertex is bounded by the number of vertices. To derive a non-deterministic polynomial time algorithm, one can thus guess for each agent a linear length path, and then check whether they can be combined into a complete run of the transfer system. The latter can be done in polynomial time by checking that at every step the global energy exceeds the one needed for the next transition. Therefore, for acyclic local arenas, *ua*-REACH is **NP**-complete.

Towards a complexity upper-bound beyond the acyclic case, we observe that thanks to Theorem 1, *ua*-REACH reduces in polynomial time to *us*-REACH. We will state in Theorem 4 that the latter is solvable in polynomial space.

**Corollary 1.** *ua*-REACH is in PSPACE.

## 4.2 Strongly and Weakly Synchronous Semantics

**Theorem 4.** *uw-REACH is in PSPACE.*

*Proof.* To prove membership in PSPACE, we show a small witness property. Precisely, in the *uw*-semantics, there exist exponential bounds  $B_{\max}$  and  $L_{\max}$  such that: if there is a run to the target global state, then there is one (1) of length at most  $L_{\max}$  and (2) along which if the energy level of the agents reach  $B_{\max}$ , then the energy requirements can be ignored for the rest of the run. Further, both bounds are exponential in the size of the input transfer system.

Consider the instance  $\text{TS} = \langle \{A_1, \dots, A_n\}, \{T_{\top}\} \rangle$  of *uw*-REACH together with initial and final states  $S_0$  and  $S_f$ . Denote by  $w_{\max}$  the largest absolute value among the weights of the edges in all  $A_i$ 's. Suppose agent  $A_k$  has  $B_{\max} = n \cdot |\text{TS}|^n \cdot w_{\max}$  energy units. It can transfer  $K = |\text{TS}|^n \cdot w_{\max}$  energy units to each other agent, and still have energy level  $K$ . Now, focusing on states only, not on energy vectors, the length of a cycle-free path in the transfer system is at most  $\prod_{i=1}^n |V_i| \leq |\text{TS}|^n$ . With  $K$  energy units, each agent is hence able to take an acyclic path to its target, losing at most  $w_{\max}$  energy units at each step. Hence, from a configuration storing  $B_{\max}$  energy units, the energy can be distributed in such a way that each agent reaches its goal assuming it is reachable from its current vertex.

When exploring runs with bounded energy levels, one thus only needs to look for relatively short runs. The number of useful configurations is bounded by  $L_{\max} = \prod_{i=1}^n |V_i| \cdot (B_{\max} + 1)$ , and each of these is visited at most once in a useful run. Therefore, useful runs are then of length at most  $L_{\max}$ , a value that is exponential in  $|\text{TS}|$ .

Let us thus first consider runs with energy levels bounded by  $B_{\max}$ . The number of configurations such runs visit is bounded by  $\prod_{i=1}^n |V_i| \cdot (B_{\max} + 1)$ . One can also notice that a run from  $S_0$  to  $S_f$  does not need to contain cycles. Hence configurations need only be visited at most once. This induces a bound on the length of relevant runs:  $L_{\max} = \prod_{i=1}^n |V_i| \cdot (B_{\max} + 2)$  (notice here that  $\prod_{i=1}^n |V_i|$  steps can be required to reach the target when energy level is above  $B_{\max}$ ).

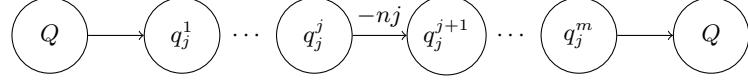
Using this bound on the maximum length of runs to reach the goal, we can design a non-deterministic algorithm that starts from the initial configuration, and explores runs of length at most  $L_{\max}$  among configurations that store at most  $B_{\max}$  energy. The algorithm returns yes if the final state  $S_f$  is reached, and fails if the length of the run exceeds  $L_{\max}$  or if the current configuration is a deadlock. It requires polynomial space to store a configuration and the step-counter. Indeed, a configuration is represented by storing for each agent its vertex and its energy level. When energy levels are bounded by  $B_{\max}$ , the space needed to store them is logarithmic in  $B_{\max}$ . Moreover, the length of runs can be encoded by a counter taking values up to  $L_{\max}$ , which can be encoded in space logarithmic in  $L_{\max}$ . Since both bound are exponential in  $\mathcal{A}$ , polynomial space in  $|\text{TS}|$  is sufficient. By Savitch's theorem [22], this non-deterministic polynomial space algorithm proves membership in PSPACE.  $\square$

**Theorem 5.** *us-REACH is PSPACE-hard.*

*Proof.* To prove PSPACE-hardness, we reduce the reachability problem for 1-safe Petri nets which is PSPACE-complete. More precisely, w.l.o.g., we consider 1-safe Petri nets in which no place is in the postset and the preset of the same transition; reachability is known to be PSPACE-complete for this class [7].

A Petri net is a tuple  $\mathcal{N} = \langle P, T; F \rangle$  where  $P = \{p_1, \dots, p_n\}$  is a set of places,  $T = \{t_1, \dots, t_m\}$  is a set of transitions, and  $F \subseteq P \times T \cup T \times P$  is a flow relation. A marking of a Petri net is a map  $M : P \rightarrow \mathbb{N}$  that associates a number of tokens to each place. The preset of a transition is the set of places  $\bullet t = \{p \in P \mid (p, t) \in F\}$  and the postset of  $t$  is the set of places  $t^\bullet = \{p \in P \mid (t, p) \in F\}$ . A transition is firable from marking  $M$  if, for every place  $p \in \bullet t$ ,  $M(p) > 0$ . Firing a transition  $t$  from marking  $M$  decrements  $M(p)$  by 1 for every place of  $p$  the preset, and increments  $M(p')$  for each  $p'$  in its postset. We write  $M[t]M'$  when  $M'$  is the marking obtained by firing  $t$  from  $M$ . Given a Petri net  $\mathcal{N}$ , one can define the set of reachable markings  $\text{Reach}(\mathcal{N}, M_0)$  that are reachable from  $M_0$ . The net  $\mathcal{N}$  is *1-safe* if, for every marking  $M$  in  $\text{Reach}(\mathcal{N}, M_0)$  and every place  $p$ ,  $M(p) \leq 1$ . The reachability problem for Petri nets consists in deciding whether a given input marking  $M$  belongs to  $\text{Reach}(\mathcal{N}, M_0)$ .

Let  $\mathcal{N} = \langle P, T; F \rangle$  be a 1-safe Petri net. Consider the transfer system with a trivial transfer group  $\text{TS} = \langle \{A_C, A_1, \dots, A_n\}, \{T_\top\} \rangle$  represented in Figures 4 and 5. This arena is composed of one control agent  $A_C$ , and one agent  $A_i$  per place  $p_i \in P$ .



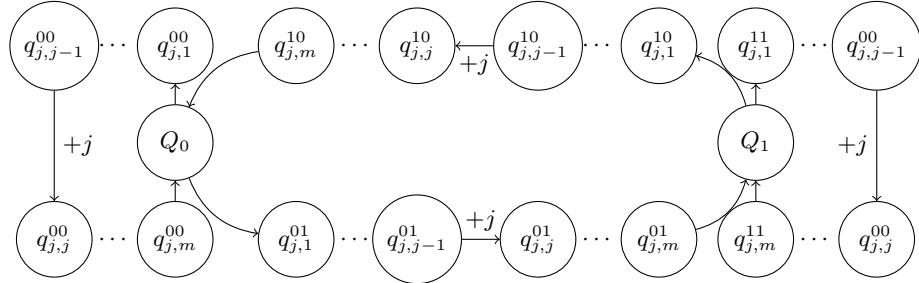
**Fig. 4.** Local arena of agent  $A_C$ . Only vertices relevant for transition  $t_j$  are depicted.

Formally, the control agent is  $A_C = (V_C, E_C)$  with

$$\begin{aligned} V_C &= \{Q\} \cup \left\{ q_j^{j'} \mid j, j' \in \llbracket 1, m \rrbracket \right\}; \\ E_C &= \left\{ q_j^{j'} \xrightarrow{w_j^{j'}} q_j^{j'+1} \mid j' \in \llbracket 1, m-1 \rrbracket, j \in \llbracket 1, m \rrbracket \right\} \\ &\cup \left\{ Q \xrightarrow{0} q_j^1, q_j^m \xrightarrow{w_j^m} Q \mid j \in \llbracket 1, m \rrbracket \right\} \end{aligned}$$

and  $\forall j, w_j^j = -n \cdot j$  and all other weights are null.

Intuitively, the controller chooses to fire  $t_j$  by moving to vertex  $q_j$  and expects to get enough energy in time. We will see that the only way for  $A_C$  to pay  $nj$  energy units at step  $j$  is if other agents have also chosen to fire  $t_j$ .



**Fig. 5.** Local arena of agent  $A_i$ . Only edges relevant for transition  $t_j$  are depicted.

Then, for every  $i \in \llbracket 1, n \rrbracket$ , there is an agent  $A_i = (V_i, E_i)$  with  $V_i = \{Q_0, Q_1\} \cup \{q_{j,j'}^{AB} \mid j, j' \in \llbracket 1, m \rrbracket, A, B \in \{0, 1\}\}$ , and

$$E_i = \left\{ q_{j,j'-1}^{AB} \xrightarrow{w_{j,j'}^{AB}} q_{j,j'}^{AB} \mid j' \in \llbracket 2, m \rrbracket, j \in \llbracket 1, m \rrbracket, A, B \in \{0, 1\} \right\} \cup \left\{ Q_A \xrightarrow{w_{j,1}^{AB}} q_{j,1}^{AB}, q_{j,m}^{AB} \xrightarrow{0} Q_B \mid j \in \llbracket 1, m \rrbracket, A, B \in \{0, 1\} \right\}$$

and for every transition  $t_j \in T$ , if  $p_i \in \bullet t_j$ , then  $w_{jj}^{00} = w_{jj}^{01} = w_{jj}^{11} = 0$  and  $w_{jj}^{10} = j$ ; else if  $p_i \in t_j^\bullet$ , then  $w_{jj}^{00} = w_{jj}^{10} = w_{jj}^{11} = 0$  and  $w_{jj}^{01} = j$ ; otherwise  $w_{jj}^{10} = w_{jj}^{01} = 0$  and  $w_{jj}^{11} = w_{jj}^{00} = j$ . All other weights are null. Intuitively, agent  $A_i$  at  $Q_0$  represents place  $p_i$  having no tokens and agent  $A_i$  at  $Q_1$  represents  $p_i$  having one token.  $A_i$  can provide for the  $j$  energy units needed by  $A_C$  through simulating the firing of  $t_j$  only if  $A_i$  is on a path corresponding to the effect of  $t_j$  on  $p_i$ : if  $p_i$  loses one token ( $p_i \in \bullet t_j$ ), the only correct path is through the 10-vertex; if  $p_i$  gains one token ( $p_i \in t_j^\bullet$ ), the only correct path is via the 01-vertex; otherwise  $t_j$  has no effect on  $p_i$ : the correct paths are via the 00- or 11-vertex depending on the current marking.

Obviously, if firing transition  $t_j$  from the marking  $M$  leads to the marking  $M'$ , there is a run from  $C(M)$  to  $C(M')$ : For all  $p_i \in \bullet t_j$ ,  $A_i$  follows the path 10, for all  $p_i \in t_j^\bullet$ ,  $A_i$  follows the path 01 and each other agent  $A_i$  follows the path  $M(p_i)M(p_i)$ . Note that each agent  $A_i$  follows the path  $M(p_i)M'(p_i)$ . With these choices, before states  $q_{jj}$  each agent except  $A_C$  gains  $j$  energy units. On  $q_{jj}$  they all transfer that amount to  $A_C$  which can leave  $q_{jj}$  with exactly enough energy to pay the  $-nj$ .

Now suppose that there is a run  $\rho : C(M) \rightsquigarrow_{TS}^a C'$  with  $m + 1$  move transitions. We show that there exists  $M'$  such that  $C' = C(M')$ . If agent  $A_C$  follows states  $q_{jj}$ . Suppose agents  $A_i$  follow states  $q_{j_i}$ , they will receive at most  $j_i$  energy units through the  $j_i$ -th edge of their path. The amount of energy available to  $A_C$  before going through its  $j + 1$ -th edge is at most  $\sum_{j_i \leq j} j_i$  because if  $j_i > j$  this energy has not been gained before the  $j + 1$ -th edge. The only way for  $\sum_{j_i \leq j} j_i$  to be greater than or equal to  $nj$  is if for all  $i$ ,  $j_i = j$ . Thus agents  $A_i$  follow

$q_j$  states. But because  $A_i$  must gain  $j$  energy units, it must follow a path that represents a possible behavior of the firing of  $t_j$  on place  $p_i$ . Thus  $\rho$  is exactly as described in the first part of this proof,  $t_j$  is enabled by  $M$  and  $C'$  represents the marking  $M'$  resulting from firing  $t_j$  in  $M$ :  $C' = C(M')$ .

Finally, this reduction is polynomial :  $|A_C| = O(m(m + \log(nm)))$  and  $\forall i, |A_i| = O(m(m + \log m))$ .  $|\text{TS}| = O(nm^2 \log(mn))$ .  $\square$

Thanks to Theorem 2, we deduce:

**Corollary 2.** *us-REACH and uw-REACH are PSPACE-complete.*

## 5 Arbitrary transfer groups

### 5.1 Asynchronous semantics

Let us now consider the complexity of  $\ell a$ -REACH, i.e., reachability for transfer systems with local transfer groups, and under asynchronous semantics. We show below that this problem is PSPACE-complete. The PSPACE membership is shown by exhibiting an algorithm that requires polynomial space to reach a target configuration. The PSPACE-hardness is proved by a reduction from a reachability problem for safe Petri nets. We give a construction that builds for each safe Petri net, a transfer system of polynomial size w.r.t. the original net, and whose runs simulate that net. We already mentioned that [7] proved PSPACE-completeness of reachability for safe Petri nets. Later, [9] has showed that reachability is NP-complete for free-choice safe Petri nets. However, the encoding shown below applies to any 1-safe Petri net. Let us start with the PSPACE membership.

**Theorem 6.**  *$\ell a$ -REACH is in PSPACE.*

*Proof.* Consider  $\text{TS} = \langle \{A_1, \dots, A_n\}, \mathcal{T} \rangle$ . Call  $e_{\max}$  the biggest weight on an edge in  $\text{TS}$  and  $S_{\max}$  the biggest size among sets  $S_i$ . From each vertex of its graph, an agent with  $B_{\max} = S_{\max} \cdot e_{\max}$  energy units does not need to receive energy from an other agent to reach any other vertex of its graph. In the sequel, we give an upper bound on the useful energy level of an agent, taking into account that it may transfer energy to others to help them achieve their reachability objective. We show that if  $\text{TS}$  is a positive instance of  $\ell a$ -REACH, then there exists a witness execution  $\rho$  in which the energy of each agent is bounded by  $B_{\max}(2n - 1)$ .

Fix  $\rho \in \text{Runs}^a(\text{TS})$ . We say that  $A_i$  helps  $A_j$  by  $e$  energy units along  $\rho$  with 0 intermediary if  $A_i$  sends at least  $e$  energy units to  $A_j$  through a transfer transition in  $\rho$  and the energy level of  $A_j$  is at least  $e$  right after the last transfer transition involving this agent in  $\rho$ . We say that  $A_i$  helps  $A_j$  by  $e$  energy units along  $\rho$  with  $k \geq 1$  intermediaries if  $A_i$  transfers at least  $e$  energy units to an other agent  $A_p$  that helps  $A_j$  along  $\rho$  by  $e$  energy units with  $k - 1$  intermediaries. If  $A_i$  helps  $A_j$  along  $\rho$  by  $e$  energy units with any number of intermediaries, we say that  $A_i$  helps  $A_j$  by  $e$  energy units for short. This can be thought of as if  $A_i$  has ultimately sent  $e$  energy units to  $A_j$  that it can keep for itself.

We show by induction on  $k$  that if an agent starts  $\rho$  with  $(2k-1)B_{\max}$  energy units, it may help up to  $k$  other agents by  $B_{\max}$  energy units and have at least  $B_{\max}$  energy units after its last transfer transition. The case  $k=0$  is immediate. Suppose the property holds until  $k \geq 0$ . If agent  $A_i$  has  $(2k+1)B_{\max}$  energy units, it may try to meet an other agent  $A_j$  at cost at most  $B_{\max}$  leaving at least  $2kB_{\max}$  energy units. If  $A_i$  sends  $(2k'-1)B_{\max}$  energy units to  $A_j$  for some  $k' \in \llbracket 0, k \rrbracket$ , then  $A_j$  may help up to  $k'$  other agents by  $B_{\max}$  and  $A_i$  may help up to  $k-k'-1$  other agents by  $B_{\max}$ . Note that  $A_i$  helps all the up to  $k'$  agents that  $A_j$  helps (with an additional intermediary) and because  $A_j$  has at least  $B_{\max}$  energy units after its last transfer transition,  $A_i$  also helps  $A_j$  which adds up to a total of  $k$  agents helped. As a consequence, an agent never needs to have more than  $(2n-1)B_{\max}$  energy units since helping every other agents by  $B_{\max}$  while still having that much afterwards is enough for  $\mathsf{TS}$  to reach the final state.

The same way as we showed it in the proof of Theorem 4, we have now an exponential bound in  $O(n|\mathsf{TS}|2^{|\mathsf{TS}|})$  on useful energy levels which results in a polynomial bound in  $O(|\mathsf{TS}| \log(n|\mathsf{TS}|))$  on the space needed to store a useful configuration and an exponential bound  $L_{\max} \in O(n|\mathsf{TS}|2^{|\mathsf{TS}|})$  on the length of useful runs. In the end, there exists an  $\mathbf{NPSPACE}$  algorithm that explores runs of size at most  $L_{\max}$  and either fails if the final state  $S_f$  is not reached in  $L_{\max}$  steps (the current length of the run can be stored in space  $O(|\mathsf{TS}| \log(n|\mathsf{TS}|))$ ) or if a deadlock is reached, and succeeds otherwise. By Savitch's theorem, we get that  $\ell a\text{-REACH}$  is in  $\mathbf{PSPACE}$ .  $\square$

**Theorem 7.**  $\ell a\text{-REACH}$  is  $\mathbf{PSPACE}$ -hard.

*Proof (sketch).* We encode a reachability problem for safe Petri nets in a  $\ell a$ -REACH problem with a transfer system whose size is linear in the size of the considered net. Let  $\mathcal{N} = (P, T; F)$  be a safe Petri net with initial marking  $M_0$ . We build a transfer system composed of  $n+1$  agents,  $\mathsf{TS}_{\mathcal{N}} = \langle \{A_C, A_1, \dots, A_n\}, \mathcal{T} \rangle$  simulating the behavior of  $\mathcal{N}$ . We do not give the whole construction here, and refer to [2] for details. The first agent  $A_C$  is a controller that initiates the simulation of a transitions firing. Agents of the form  $A_i$  encode the contents of place  $p_i$  through their states, and simulate the effect of a transition firing via sequences of moves. We distinguish in particular two states  $A_{i,0}$  and  $A_{i,1}$ , used to encode  $M(p_i) = 0$  and  $M(p_i) = 1$  in order to represent the marking  $M$ . Then we set an ordering on places, and ensure that when the controller agent chooses a particular transition  $t$ , all place agents choose the transition they simulate, but have to wait for energy from their predecessor to progress in this simulation. If two agents choose different transitions, the system deadlocks. Upon agreement on the chosen transition to simulate, the last agent  $A_n$  eventually sends back energy to the controller, acknowledging the fact that all places are engaged in the simulation of the same transition from the same marking. The controller then launches another round among place agents (still by transferring energy) who successively update their state to encode the effect of  $t$  on their place contents before acknowledging all changes to the controller. For instance, if transition  $t$

consumes a token from place  $p_i$  and  $M(p_i) = 1$ , then agent  $A_i$  will start its interactions from state  $A_{i,1}$  and will end the simulation of  $t$ 's firing in state  $A_{i,0}$ . If a transition  $t$  is chosen,  $p_i$  is in the preset of  $t$ , but  $A_i$  started from state  $A_{i,0}$ , then choosing to simulate  $t$  will send  $A_i$  to a deadlock state, and will prevent reaching global states that encode markings. During this simulation process, if a place agent does not transfer energy to its successor and moves to its next state, then the system necessarily deadlocks or can return to the situation where the transfer was missed. When an agent keeps energy for its future moves, it can only repeat the choice of a new transition to simulate, hence canceling its previous choice. The only way to simulate properly a Petri net transition is if all agents choose the same transition and transfer their energy to their successor. Other choices lead either to deadlocks or livelocks in configurations that do not encode markings.

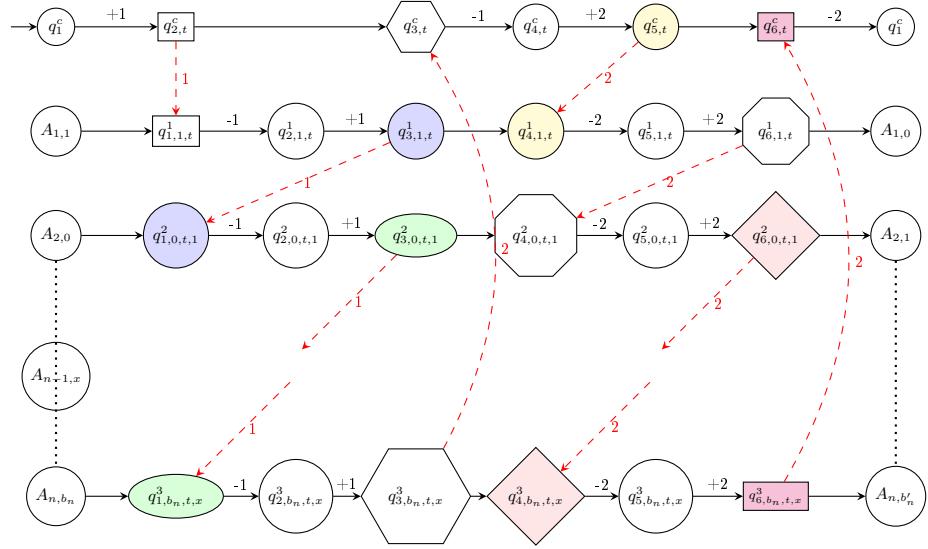
Configurations of  $\text{TS}_{\mathcal{N}}$  of the form  $\langle (q_1^C, A_{1,b_1}, \dots, A_{n,b_n}), \overrightarrow{0} \rangle$  will be called *stable* configurations (and other configurations *unstable*). Stable configurations represent an encoding of a marking of  $\mathcal{N}$ . A first step in the proof is to show that, for a pair of markings  $M, M'$  of  $\mathcal{N}$ , such that  $M[t]M'$ , there exists a run from  $C(M) = \langle (q_1^C, A_{1,M(p_1)}, \dots, A_{n,M(p_n)}), \overrightarrow{0} \rangle$  to  $C(M')$  in  $\text{TS}_{\mathcal{N}}$ . The shape of such runs is depicted in Figure 6. In this figure, agents steps are organized as local sequences, red dashed arrows depict energy transfers, and vertices that belong to the same transfer group have identical shape and color. This first part of the proof shows that a transfer system can simulate a safe Petri net, as for  $M[t]M'$ , there exists a "canonical" run  $\rho_{M[t]M'}$  from stable configuration  $C(M)$  to stable configuration  $C(M')$  that does not visit any other stable configuration.

It then remains to show that  $\text{TS}_{\mathcal{N}}$  does not allow the reachability of stable configurations that are not encodings of reachable markings. To this extent, we look at the transition system composed of possible configurations and moves of the transfer system and highlight its properties. One can show that the runs encoding firing of a transition  $t$  of a safe net follow a particular pattern: Agents choose their transition and guess their predecessor's bit. When all agents agree on a common transition, one unit of energy flows from  $A_c$  to  $A_1, A_2 \dots A_n$  and then back to  $A_c$ . A second round then starts, with two units of energy transferred successively to  $A_c$  to  $A_1, A_2 \dots A_n$ . If two agents did incompatible choices of simulated transition or control bit, then a deadlock (without reaching any stable configuration) is unavoidable. Similarly, if an agent does not transfer all its energy to its successor, then the system either deadlocks, or enters an infinite sequence of moves that can be only exited by sending back the faulty agent to the state from which the wrong choice was performed, still without visiting a stable configuration.

We can hence conclude that a marking  $M$  of the safe Petri net  $\mathcal{N}$  is reachable from marking  $M_0$  if and only if the stable configuration  $C(M)$  is reachable from the configuration  $C(M_0)$  in  $\text{TS}_{\mathcal{N}}$ . Hence,  $\ell a\text{-REACH}$  is PSPACE-hard.  $\square$

## 5.2 Strongly synchronous semantics

**Theorem 8.**  $\ell s\text{-REACH}$  is in EXPSPACE.



**Fig. 6.** Simulating a transition  $t$  moving a token from  $p_1$  to  $p_2$

*Proof.* The exponential space algorithm we exhibit to show membership in EX-**PSPACE** involves the construction of a VASS  $\mathcal{V}$  of exponential size yet polynomial dimension. Rackoff's backward algorithm for VASS coverability requires  $2^{O(d)} \cdot \log |\mathcal{V}|$  space [16] where  $d$  is the dimension of  $\mathcal{V}$ . This results in an overall exponential space algorithm for  $\ell s$ -**REACH**.

The above-mentioned VASS  $\mathcal{V}$  is built as follows. Given an instance  $\text{TS} = \langle \{A_1, \dots, A_n\}, \mathcal{T} \rangle$  with  $\forall i, A_i = (V_i, E_i)$ , we define  $\mathcal{V} = \langle \mathcal{S}, \delta \rangle$  such that:

- $\mathcal{S} = \prod_{i=1}^n V_i$
- $\forall (q_1 \xrightarrow{w_1} q'_1, \dots, q_n \xrightarrow{w_n} q'_n) \in \prod_{i=1}^n E_i, (q_1, \dots, q_n) \xrightarrow{w_1, \dots, w_n} (q'_1, \dots, q'_n) \in \delta$
- $\forall \vec{q} = (q_1, \dots, q_n) \in \mathcal{S}, \forall q_i \neq q_j, \left[ \exists T \in \mathcal{T} \mid q_i, q_j \in T \implies \vec{q} \xrightarrow{\vec{w}_{ij}} \vec{q} \in \delta \right]$

where  $\vec{w}_{ij}$  is the vector with only zeros except +1 at index  $i$  and -1 at index  $j$ .

With this construction, whether there exists a run  $\rho_{\text{TS}} : \langle \mathcal{S}, \vec{e} \rangle \rightsquigarrow_{\text{TS}}^s \langle \mathcal{S}', \vec{e}' \rangle$  is equivalent to whether there exists a run  $\rho_{\mathcal{V}} : \langle \mathcal{S}, \vec{e} \rangle \rightsquigarrow_{\mathcal{V}} \langle \mathcal{S}', \vec{e}' \rangle$ . We show the direct implication by induction on the length of  $\rho_{\text{TS}}$ :

- If  $\rho_{\text{TS}}$  is empty,  $\langle \mathcal{S}, \vec{e} \rangle = \langle \mathcal{S}', \vec{e}' \rangle$  and the property holds.
- Let  $\rho_{\text{TS}}$  be of length  $l + 1$ . Let  $\langle \mathcal{S}^{-1}, \vec{e}^{-1} \rangle$  be the penultimate configuration of  $\rho_{\text{TS}}$ . By induction, there exists  $\rho_{\mathcal{V}} : \langle \mathcal{S}, \vec{e} \rangle \rightsquigarrow_{\mathcal{V}} \langle \mathcal{S}^{-1}, \vec{e}^{-1} \rangle$ . If the last transition of  $\rho_{\text{TS}}$  is a move transition, for all  $i$  there is a transition  $q_i^{-1} \xrightarrow{w_i} q'_i \in E_i$  such that all energy levels  $e'_i = e_i^{-1} + w_i$  are non-negative.  $\rho_{\mathcal{V}}$

may reach  $\langle S', \vec{e}^j \rangle$  with the transition  $S^{-1} \xrightarrow{w_1, \dots, w_n} S'$  induced by these transitions. Otherwise, there are some  $i, j \in \llbracket 1, n \rrbracket$  and some  $T \in \mathcal{T}$  with  $q'_i, q'_j \in T$ ,  $e_i^{-1} + e_j^{-1} = e'_i + e'_j$  and  $\forall k \notin \{i, j\}$ ,  $e_k^{-1} = e'_k$ . The transfer from coordinate  $i$  to coordinate  $j$  may be decomposed using the transition  $S' \xrightarrow{\vec{w}_{ij}} S'$  since  $q'_i$  and  $q'_j$  share the same group  $T$ . In all cases,  $\langle S', \vec{e}^j \rangle$  is reachable.

Conversely, by induction on the length of  $\rho_V$ :

- The case of  $\rho_V$  empty is immediate.
- Let  $\rho_V$  be of length  $l+1$ . Let  $\langle S^{-1}, \vec{e}^{-1} \rangle$  be the penultimate configuration of  $\rho_V$ . By induction, there exists  $\rho_{TS} : \langle S, \vec{e} \rangle \rightsquigarrow_{TS}^s \langle S^{-1}, \vec{e}^{-1} \rangle$ . The last transition of  $\rho_V$  has two possible forms. Either there is  $(q_1 \xrightarrow{w_1} q'_1, \dots, q_n \xrightarrow{w_n} q'_n) \in \prod_{i=1}^n E_i$  such that this last transition is  $(q_1, \dots, q_n) \xrightarrow{w_1, \dots, w_n} (q'_1, \dots, q'_n)$ , in which case  $\rho_{TS}$  can be extended by the move transition induced by these edges to reach  $S'$ . Note that because the coordinates are maintained non-negative in  $V$ , so will the energy levels. Or, there are some coordinates  $i, j \in \llbracket 1, n \rrbracket$  such that the last transition of  $\rho_V$  is  $S' \xrightarrow{\vec{w}_{ij}} S'$  with some transfer group  $T \in \mathcal{T}$  such that  $q'_i, q'_j \in T$ . In that case, a transfer transition from  $\rho_A$  that sends 1 energy unit from  $A_i$  to  $A_j$  reaches  $\langle S', \vec{e}^j \rangle$ .

According to the previous result, we conclude that if an instance is positive for  $\ell s\text{-REACH}$  then its joined instance for VASS-cover is also positive.

Furthermore,  $V$  is of size  $O(|TS|^{2n})$  since there are  $O(|TS|^n)$  states,  $O(n^2)$  loops on each state and  $O((|TS|^n)^2)$  other transitions.

□

For the complexity lower-bound, recall that  $\ell a\text{-REACH}$  is **PSPACE**-hard (Theorem 7) and conclude with Theorem 1 that:

**Corollary 3.**  $\ell s\text{-REACH}$  is **PSPACE**-hard.

### 5.3 Weakly synchronous semantics

Perhaps surprisingly, the relaxation of agents synchronization from strong to weak synchronous semantics, *i.e.* the fact that agents with no enabled edges may not move simultaneously with other agents, leads to undecidability. The main reason is the following. Both the asynchronous and (strongly) synchronous semantics enjoy a monotonicity property: higher energy levels can only enable more transitions and allow to reach more configurations. This monotonicity does not hold under the  $w$ -semantics. Indeed, it can be profitable for an agent to reach a vertex with a low energy level, so that it is allowed to “wait” for other agents to move and later gain energy through a transfer. Intuitively, this behaviour allows one to test whether an agent has energy left, thus encoding a zero test.

**Theorem 9.**  $\ell w\text{-REACH}$  is undecidable.

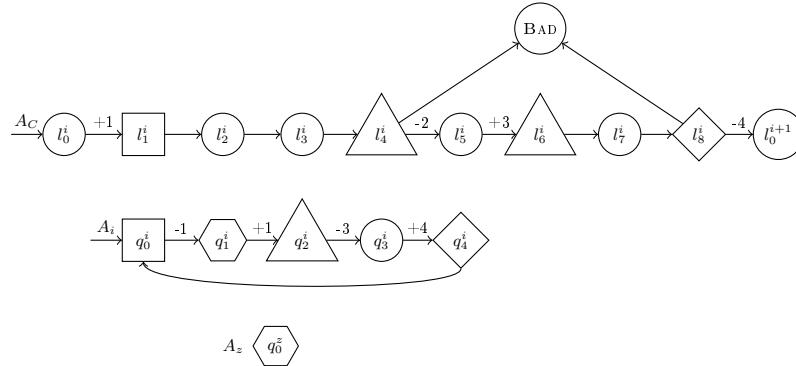
*Proof.* We give a reduction from the termination problem of Minsky machines, which is known to be undecidable [19]. Let us start with a quick recall on Minsky machines. A Minsky machine  $\mathcal{M}$  is described by two counters  $x$  and  $y$ , as well as a sequence of commands  $l_0, \dots, l_m$  where  $l_0$  is the starting command,  $l_m$  ends the run of the system, and every command  $l_0$  to  $l_{m-1}$  is of one of the following three types:

- increment counter  $c \in \{x, y\}$ , move to the next command;
- decrement counter  $c \in \{x, y\}$ , move to the next command<sup>3</sup>;
- if counter  $c \in \{x, y\}$  is equal to 0, move to command  $l_k$ , otherwise move to command  $l_j$ .

In summary, the machine goes through a list of commands starting with  $l_0$ , incrementing, decrementing counters, or testing whether a counter is equal to 0 in order to select the new command to jump to—and it terminates whenever it reaches  $l_m$ . The *termination problem* for Minsky machines consists in deciding, given a machine  $\mathcal{M}$ , whether  $\mathcal{M}$  terminates.

In our reduction, we use two agents with local arenas  $A_x$  and  $A_y$ , storing as energy level the current value of each counter, one control agent  $A_C$  that encodes the control flow of the Minsky machine, and one additional agent  $A_i$  for each command  $l_i$  to simulate the effect of  $l_i$  when it is activated by  $A_C$ . Sink vertices  $\text{BAD}$  are used to punish agents that reach a vertex with an energy level that differs from the one that is expected in the Minsky machine simulation.

We now detail how to encode a decrement: assume command  $l_i$  decrements counter  $z \in \{x, y\}$ . Only three agents take part in the simulation of  $l_i$ : the control agent  $A_C$ , agent  $A_z$  associated to counter  $z$  and agent  $A_i$  dedicated to  $l_i$ .



**Fig. 7.** Encoding a decrement. To the exception of circles, states with the same shape belong to the same transfer group.

<sup>3</sup> Note that the machine must be designed so that decrement can only occur when the counter value is positive.

Formally, we define the transfer system associated with command line  $l_i$  as  $\text{TS}_i = \langle \{A_z, A_C, A_i\}, \mathcal{T}_i \rangle$ , depicted in Figure 7, with:

- $A_z = (\{q_0^z\}, \emptyset)$ ;
- $A_i = (V_i, E_i)$ , where  $V_i = \{q_j^i \mid j = 0 \dots 4\}$ ;  $E_i = \{q_j^i \xrightarrow{w_j^i} q_k^i \mid j \in \llbracket 0, 4 \rrbracket \wedge (k \equiv j + 1 \pmod{5})\}$ ; with  $w_0^i = -1, w_1^i = 1, w_2^i = -3, w_3^i = 4$  and  $w_4^i = 0$ .
- $A_C = (V_i^C, E_i^C)$ , where  $V_i^C = \{l_j^i \mid j \in \llbracket 0, 8 \rrbracket\} \cup \{\text{BAD}, l_0^{i+1}\}$  and  $E_i^C = \{l_j^i \xrightarrow{w^i} l_r^i \mid j = 0 \dots 7 \wedge r = j + 1\} \cup \{l_8^i \xrightarrow{w^8} l_0^{i+1}, l_4^i \xrightarrow{0} \text{BAD}, l_8^i \xrightarrow{0} \text{BAD}\}$ ; and  $w^0 = 1, w^4 = -2, w^5 = 3, w^8 = -4$  and other weights are null.
- $\mathcal{T}_i$  contains the groups  $\{l_1^i, q_0^i\}$ ,  $\{q_0^z, q_1^i\}$ ,  $\{l_4^i, l_6^i, q_2^i\}$  and  $\{l_8^i, q_4^i\}$ .

Recall that this addresses a single command line  $l_i$ . To encode a complete Minsky machine  $\mathcal{M}$  with  $k$  instructions, we assemble the arenas for all command lines into the transfer system  $\text{TS}_{\mathcal{M}} = \langle (A_C, A_x, A_y, A_1, \dots, A_k), \bigcup \mathcal{T}_i \rangle$   $A_C$ , in which  $A_x$  and  $A_y$  are the unions of sets of vertices and edges of the arenas of every command line. In particular, the vertex  $l_0^{i+1}$  is shared with the local arena associated with the command line  $l_{i+1}$ .

Let us show that if the agents  $A_C, A_i$  and  $A_z$  start in  $\langle (l_0^i, q_0^i, q_0^z), (0, 0, n) \rangle$  with  $n > 0$ , then the only way to avoid BAD leads them to the configuration  $\langle (l_0^{i+1}, q_0^z), (0, 0, n - 1) \rangle$ . Hence, executing the command line  $l_i$  will indeed decrement the energy of  $A_z$  by 1.

In the first step, only  $A_C$  can move, reaching  $l_1^i$  with 1 unit of energy. There it can either transfer this energy to  $A_i$  or keep it. If it chooses not to transfer, it remains the only agent able to move, and when reaching  $l_4^i$ , it will have 1 unit of energy, forcing it to go to BAD. Assume thus that  $A_C$  transfers its energy unit to  $A_i$ . Both then move synchronously to  $l_2^i$  and  $q_1^i$ . In  $q_1^i$ ,  $A_i$  can interact with  $A_z$ . Let  $m$  be the energy level of  $A_i$  at that point. Agents  $A_C$  and  $A_i$  then reach  $l_3^i$  and  $q_2^i$  with energy levels 0 and  $m + 1$ . If  $m + 1 \geq 3$ , then both agents move to  $l_4^i$  and  $q_3^i$  and in the next step,  $A_C$  is forced to go to BAD. In order for  $A_C$  to avoid its BAD vertex, it must be the case that  $m + 1 < 3$ . In this case,  $A_C$  progresses to  $l_4^i$  while  $A_i$  remains in  $q_2^i$ , because the only available edge consumes 3 units of energy. With this move,  $A_C$  can receive energy from  $A_i$ , as their current vertices belong to the same transfer group. Now  $A_C$  needs 2 energy units to avoid going to BAD, which requires  $m + 1 \geq 2$ . Hence  $m + 1 = 2$  which implies that during their interaction,  $A_z$  transferred 1 energy unit to  $A_i$ , leaving  $A_z$  with  $n - 1$  units of energy. After transferring 2 energy units to  $A_C$ ,  $A_i$  remains stuck in  $q_2^i$  while  $A_C$  progresses to  $l_5^i$  and then  $l_6^i$  with 3 energy units. Again,  $A_C$  can choose to move on its own without transferring energy to  $A_i$ , but it will eventually reach vertex  $l_8^i$ , and lacking 4 energy units will be forced to go to BAD. If  $A_C$  transfers 3 energy units to  $A_i$  they both move to  $l_8^i$  and  $q_4^i$  where  $A_i$  can then transfer to  $A_C$  the 4 energy units required to avoid BAD. This then leads to a configuration where  $A_C$  is in state  $l_0^{i+1}$  with no energy left,  $A_i$  is back in vertex  $q_0^i$  also with no energy left, and  $A_z$  is in vertex  $q_0^z$  with an energy level  $n - 1$ . In this construction, transferring other quantities of energy always leads to deadlock configurations.

The encoding of the increment is similar. The zero test however is even more involved, allowing the agents to remain stuck in some vertices and wait for the other agents iff the counter value is 0. The constructions for these two operations are provided in details in [2]. Altogether, the three constructions ensure that in order to avoid the BAD vertices, the agents must correctly implement the command of the Minsky machine.

To complete the reduction, the reachability objective for the transfer system is defined as follows. The last command  $l_m$  of the Minsky machine, is represented by a single vertex  $l_0^m$  which is the target vertex of the control agent. The targets of other agents are the set of vertices  $l_0^m, q_0^i$  for  $i \in \llbracket 0, m-1 \rrbracket$  and  $q_0^z$  for  $z \in \{x, y\}$ . As the agents must avoid the BAD vertices, the above constructions ensure that the target state is covered in the transfer system if and only if the Minsky machine terminates.  $\square$

## 6 Conclusion

This paper introduced and studied a cooperative game model, in which agents move on local weighted arenas, and can help each other by transferring energy to their peers. We considered a global reachability question, *i.e.*, whether it is possible to reach a system configuration where each agent is in their goal vertex, while always keeping all energy levels non-negative. While transfer systems can be easily encoded as vector addition systems with states, and our reachability problems as a coverability question, we showed that the energy transfer feature induces a complexity drop, with complexities ranging from PSPACE to EXPSPACE. For asynchronous and strongly synchronous semantics, we exploited a form of monotonicity and a small witness property. However, monotonicity does not hold under weak synchronous semantics, leading to undecidability.

An obvious future work is to close the complexity gaps for  $ua$ -REACH and  $\ell s$ -REACH. The similarities between transfer systems and subclasses of VASS, in particular 1-dim VASS for  $ua$ -REACH might help solving this issue. Considering extensions of transfer systems is another interesting research direction, for instance with features that have been considered for Petri nets while maintaining decidability such as transfers or resets. Finally, beyond the purely cooperative question we tackled here, it would also be interesting to consider alternative problems in which the agents have conflicting objectives.

## References

1. Toshiro Araki and Tadao Kasami. Some decision problems related to the reachability problem for Petri nets. *Theoretical Computer Science*, 3(1):85–104, 1976.
2. Nathalie Bertrand, Loïc Hélouët, Engel Lefaucheux, and Luca Paparazzo. Reachability in multi-agent transfer systems (extended version). Technical report, HAL Inria <https://inria.hal.science/hal-05366409>, 2025.

3. Tomás Brázdil, Petr Jančar, and Antonín Kucera. Reachability games on extended vector addition systems with states. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP'10)*, volume 6199 of *Lecture Notes in Computer Science*, pages 478–489. Springer, 2010.
4. Thomas Brihaye, Véronique Bruyère, Aline Goeminne, Jean-François Raskin, and Marie van den Bogaard. The complexity of subgame perfect equilibria in quantitative reachability games. In *Proceedings of the 30th International Conference on Concurrency Theory (CONCUR'19)*, volume 140 of *LIPICS*, pages 13:1–13:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
5. Thomas Brihaye, Amit Kumar Dhar, Gilles Geeraerts, Axel Haddad, and Benjamin Monmege. Efficient energy distribution in a smart grid using multi-player games. In *Proceedings of Cassting Workshop on Games for the Synthesis of Complex Systems and 3rd International Workshop on Synthesis of Complex Parameters (Cassting/SynCoP'16)*, volume 220 of *EPTCS*, pages 1–12, 2016.
6. Nils Bulling and Valentin Goranko. Combining quantitative and qualitative reasoning in concurrent multi-player games. *Autonomous Agents and Multi-Agent Systems*, 36(1):2, 2022.
7. Allan Cheng, Javier Esparza, and Jens Palsberg. Complexity results for 1-safe nets. *Theoretical Computer Science*, 147(1&2):117–136, 1995.
8. Gianfranco Ciardo. Petri nets with marking-dependent arc cardinality: Properties and analysis. In *Proceedings of the 15th International Conference on Application and Theory of Petri Nets 1994 (PetriNets'94)*, volume 815 of *Lecture Notes in Computer Science*, pages 179–198. Springer, 1994.
9. Javier Esparza. Reachability in live and safe free-choice Petri nets is NP-complete. *Theoretical Computer Science*, 198(1-2):211–224, 1998.
10. Uli Fahrenberg, Line Juhl, Kim G. Larsen, and Jirí Srba. Energy games in multi-weighted automata. In *Proceedings of the 8th International Colloquium on Theoretical Aspects of Computing (ICTAC'11)*, volume 6916 of *Lecture Notes in Computer Science*, pages 95–115. Springer, 2011.
11. Ariel Felner, Roni Stern, Solomon Eyal Shimony, Eli Boyarski, Meir Goldenberg, Guni Sharon, Nathan R. Sturtevant, Glenn Wagner, and Pavel Surynek. Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges. In *Proceedings of the 10th International Symposium on Combinatorial Search (SOCS'17)*, pages 29–37. AAAI Press, 2017.
12. Julian Gutierrez, Aniello Murano, Giuseppe Perelli, Sasha Rubin, Thomas Steeples, and Michael J. Wooldridge. Equilibria for games with combined qualitative and quantitative objectives. *Acta Informatica*, 58(6):585–610, 2021.
13. Christoph Haase, Stephan Kreutzer, Joël Ouaknine, and James Worrell. Reachability in succinct and parametric one-counter automata. In *Proceeding of the 20th International Conference on Concurrency Theory (CONCUR'09)*, volume 5710 of *Lecture Notes in Computer Science*, pages 369–383. Springer, 2009.
14. John E. Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979.
15. Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer, 1972.
16. Marvin Künemann, Filip Mazowiecki, Lia Schütze, Henry Sinclair-Banks, and Karol Wegrzycki. Coverability in VASS revisited: Improving Rackoff's bound to obtain conditional optimality. In *Proceeding of the 50th International Colloquium on Automata, Languages, and Programming (ICALP'23)*, volume 261 of *LIPICS*, pages 131:1–131:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

17. Jérôme Leroux. Petri net reachability problem (invited talk). In *Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science (MFCS'19)*, volume 138 of *LIPICS*, pages 5:1–5:3. Schloss Dagstuhl - Leibniz-Zentrum f"ur Informatik, 2019.
18. Richard Lipton. The reachability problem requires exponential space. Technical report, Yale University, 1976.
19. Marvin L. Minsky. *Computation: Finite and infinite machines*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.
20. John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.
21. Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6:223–231, 1978.
22. Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.
23. Philippe Schnoebelen. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In *Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science 2010 (MFCS'10)*, volume 6281 of *Lecture Notes in Computer Science*, pages 616–628. Springer, 2010.