Computing Embeddings of Directed Bigraphs

Alessio Chiapperini[1], Marino Miculan[1(✉)], and Marco Peressotti[2]

[1] DMIF, University of Udine, Udine, Italy
`marino.miculan@uniud.it`
[2] IMADA, University of Southern Denmark, Odense, Denmark
`peressotti@imada.sdu.dk`

Abstract. *Directed bigraphs* are a meta-model which generalises Milner's bigraphs by taking into account the *request flow* between controls and names. A key problem about these bigraphs is that of *bigraph embedding*, i.e., finding the embeddings of a bigraph inside a larger one. We present an algorithm for computing embeddings of directed bigraphs, via a reduction to a *constraint satisfaction problem*. We prove soundness and completeness of this algorithm, and provide an implementation in jLibBig, a general Java library for manipulating bigraphical reactive systems, together with some experimental results.

1 Introduction

Bigraphical Reactive Systems (BRSs) are a family of graph-based formalisms introduced as a meta-model for distributed, mobile systems [17,22,25]. In this approach, system configurations are represented by *bigraphs*, graph-like data structures capable of describing at once both the locations and the logical connections of (possibly nested) components. The dynamics of a system is defined by means of a set of *graph rewriting rules*, which can replace and change components' positions and connections. BRSs have been successfully applied to the formalization of a wide spectrum of domain-specific models, including context-aware systems, web-service orchestration languages [4,5,20,28]. BRSs are appealing because they provide a range of general results and tools, which can be readily instantiated with the specific model under scrutiny: libraries for bigraph manipulation (e.g., DBtk [1] and jLibBig [23,24]), simulation tools [10,19,21], graphical editors [9], model checkers [27], modular composition [26], stochastic extensions [18], etc.

Along this line, [13,14] introduced *directed bigraphs*, a strict generalization of Milner's bigraphs where the link graph is directed (see Fig. 1). This variant is very suited for reasoning about *dependencies* and *request flows* between components,
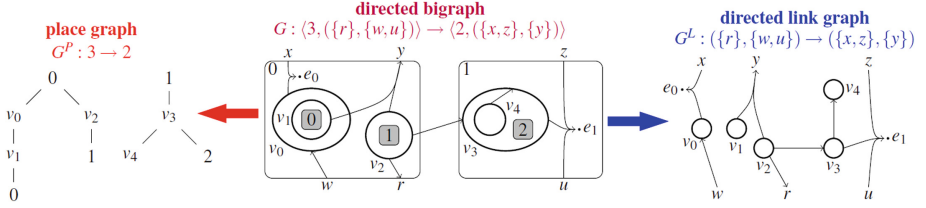
Fig. 1. An example of directed bigraph and its place and link graphs [15].

such as those found in client-server or producer-consumer scenarios. In fact, they have been used to design formal models of security protocols [12], molecular biology [2], access control [15], container-based systems [5], etc.

A key notion at the core of these results and tools is that of *bigraph embedding*. Informally, an embedding is a structure preserving map from a bigraph (called *guest*) to another one (called *host*), akin a subgraph isomorphism. Finding such embeddings is a difficult problem; in fact, the sole embedding of place graphs has been proved to be NP-complete [3]. Several algorithms have been proposed in literature for bigraphs with undirected links (see e.g. [7,11,23,29,30]), but there is no embedding algorithm for the more general case of directed bigraphs, yet.

In this work, we propose an algorithm for computing embedding of directed bigraphs (which subsume traditional ones), laying the theoretical and technical foundations for bringing directed bigraphs to tools like the ones listed above.

More precisely, in Sect. 2 we first introduce directed bigraphs and bigraphic reactive systems, generalizing [5,13]. Then, the notion of directed bigraph embedding is defined in Sect. 3. In Sect. 4 we present a reduction of the embedding problem for directed bigraphs to a constraint satisfaction problem (CSP) and show that it provides a sound and complete algorithm for computing embeddings. This reduction extends our previous (unpublished) work [23] on the embedding problem for undirected bigraphs. We have implemented this algorithm as an extension of jLibBig [24], a general Java library for BRSs; this implementation and some experimental results are reported in Sect. 5. Finally, some conclusions and directions for future work are drawn in Sect. 6.

2   Reactive Systems on Directed Bigraphs

In this section we introduce a conservative extension of the notions of *directed link graphs and bigraphs*, and *directed bigraphical reactive systems*, originally defined in [13,14].

2.1   Directed Bigraphs

Definition 1 (Polarized interface). *A* polarized interface X *is a pair* (X^-, X^+)*, where* X^+ *and* X^- *are sets of names s.t.* $X^- \cap X^+ = \varnothing$*; the two sets are called* downward *and* upward *interfaces respectively.*

Definition 2 (Polarized signature). *A* signature *is a pair* $(\mathcal{K}, ar)$, *where* $\mathcal{K}$ *is the set of* controls, *and* $ar : \mathcal{K} \to \mathbb{N} \times \mathbb{N}$ *is a map assigning to each control its polarized arity, that is, a pair* $\langle n, m \rangle$ *where* n, m *are the numbers of* positive *and* negative *ports of the control, respectively.*

We define $ar^+, ar^- : \mathcal{K} \to \mathbb{N}$ *as shorthand for the* positive *and* negative *ports of controls:* $ar^+ \triangleq \pi_1 \circ ar$, $ar^- \triangleq \pi_2 \circ ar$.

The main difference between this definition and that from [13] is that we allow also for *inward* ports in controls, whereas in [13], like in [25], controls have only outward ports. This turns up also in the definition of *points* and *handles*. The addition of negative ports enables us to represent more faithfully the dependencies between processes, entities and components, according to the micro-services paradigm.

Definition 3 (Directed Link Graph). *A directed link graph* $A : X \to Y$ *is a quadruple* $A = (V, E, ctrl, link)$ *where* X, Y, V, E *and* $ctrl$ *are defined as before, while the link map is defined as* $link : Pnt(A) \to Lnk(A)$ *where*

$$Prt^+(A) \triangleq \sum_{v \in V} ar^+(ctrl(v)) \qquad\qquad Prt^-(A) \triangleq \sum_{v \in V} ar^-(ctrl(v))$$
$$Pnt(A) \triangleq X^+ \uplus Y^- \uplus Prt^+(A) \qquad Lnk(A) \triangleq X^- \uplus Y^+ \uplus E \uplus Prt^-(A)$$

with the following additional constraints:

$$\forall x \in X^-, \forall y \in X^+ . link(y) = x \Rightarrow link^{-1}(x) = \{y\} \tag{1}$$
$$\forall y \in Y^+, \forall x \in Y^- . link(x) = y \Rightarrow link^{-1}(y) = \{x\}. \tag{2}$$

The elements of $Pnt(A)$ *are called the* points *of A; the elements of* $Lnk(A)$ *are called the* handles *of A.*

The constraint (1) means that if there is an upward inner name connected to a downward inner name, then nothing else can be connected to the latter; constraint (2) is similar, on the outer interface. Together, these requirements guarantee that composition of link graphs (along the correct interfaces) is well defined.

Direct link graphs are graphically depicted similarly to ordinary link graphs, with the difference that edges are represented as vertices of the graph and not as hyper-arcs connecting points and names.

Directed bigraphs are composed by a directed link graph and a place graph. Since the definition of place graph is the same as for pure bigraphs, we will omit it and refer the interested reader to [25].

Definition 4 (Directed Bigraph). *An* interface $I = \langle m, X \rangle$ *is composed by a finite ordinal* m, *called the* width, *and by a directed interface* $X = (X^-, X^+)$.

Let $I = \langle m, X \rangle$ *and* $O = \langle n, Y \rangle$ *be two interfaces; a directed bigraph* with *signature* $\mathcal{K}$ *from* I *to* O *is a tuple* $G = (V, E, ctrl, prnt, link) : I \to O$ *where*

– *I and O are the* inner *and* outer *interfaces;*

 – *V and E are the sets of nodes and edges;*
 – *ctrl, prnt, link are the* control, parent *and* link *maps;*

such that $G^L \triangleq (V, E, ctrl, link) : X \to Y$ is a directed link graph and $G^P \triangleq (V, ctrl, prnt) : m \to n$ is a place graph, that is, the map $prnt : m \uplus V \to n \uplus V$ is acyclic. The bigraph G is denoted also as $\langle G^P, G^L \rangle$.

Definition 5 (Composition and identities).

 – *The composition of two place graphs $F : k \to m$ and $G : m \to n$, is defined in the same way as pure bigraphs (i.e., suitable grafting of forests);*
 – *If $F : X \to Y$ and $G : Y \to Z$ are two link graphs, their composition is the link graph $G \circ F \triangleq (V, E, ctrl, link) : X \to Z$ such that $V = V_F \uplus V_G$, $E = E_F \uplus E_G$, $ctrl = ctrl_F \uplus ctrl_G$, and $link : Pnt(G \circ F) \to Lnk(G \circ F)$ is defined as follows:*

$$Pnt(G \circ F) = X^+ \uplus Z^- \uplus Prt^+(F) \uplus Prt^+(G)$$
$$Lnk(G \circ F) = X^- \uplus Z^+ \uplus Prt^-(F) \uplus Prt^-(G) \uplus E$$
$$link(p) \triangleq \begin{cases} prelink(p) & if\ prelink(p) \in Lnk(G \circ F) \\ link(prelink(p)) & otherwise \end{cases}$$

 where $prelink : Pnt(G \circ F) \uplus Y^+ \uplus Y^- \to Lnk(G \circ F) \uplus Y^+$ is $link_F \uplus link_G$. The identity link graph at X is $id_X \triangleq (\varnothing, \varnothing, \varnothing_{\mathcal{K}}, Id_{X^- \uplus X^+}) : X \to X$.
 – *If $F : I \to J$ and $G : J \to K$ are two bigraphs, their composite is*

$$G \circ F \triangleq \langle G^P \circ F^P, G^L \circ F^L \rangle : I \to K$$

 and the identity bigraph at $I = \langle m, X \rangle$ is $\langle id_m, id_{X^- \uplus X^+} \rangle$.

Definition 6 (Juxtaposition).

 – *For place graphs, the juxtaposition of two interfaces m_0 and m_1 is $m_0 + m_1$; the unit is 0. If $F_i = (V_i, ctrl_i, prnt_i) : m_i \to n_i$ are disjoint place graphs (with $i = 0, 1$), their juxtaposition is defined as for pure bigraphs;*
 – *For link graphs, the juxtaposition of two (directed) link graph interfaces X_0 and X_1 is $(X_0^- \uplus X_1^-, X_0^+ \uplus X_1^+)$. If $F_i = (V_i, E_i, ctrl_i, link_i) : X_i \to Y_i$ are two link graphs (with $i = 0, 1$), their juxtaposition is*

$$F_0 \otimes F_1 \triangleq (V_0 \uplus V_1, E_0 \uplus E_1, ctrl_0 \uplus ctrl_1, link_0 \uplus link_1) : X_0 \otimes X_1 \to Y_0 \otimes Y_1$$

 – *For bigraphs, the juxtaposition of two interfaces $I_i = \langle m_i, X_i \rangle$ (with $i = 0, 1$) is $\langle m_0 + m_1, (X_0^- \uplus X_1^-, X_0^+ \uplus X_1^+) \rangle$ (the unit is $\epsilon = \langle 0, (\varnothing, \varnothing) \rangle$). If $F_i : I_i \to J_i$ are two bigraphs (with $i = 0, 1$), their juxtaposition is*

$$F_0 \otimes F_1 \triangleq \langle F_0^P \otimes F_1^P, F_0^L \otimes F_1^L \rangle : I_0 \otimes I_1 \to J_0 \otimes J_1.$$

Polarized interfaces and directed bigraphs over a given signature $\mathcal{K}$ form a monoidal category $\mathrm{DBIG}(\mathcal{K})$.

Milner's pure bigraphs [25] correspond precisely to directed bigraphs with positive interfaces only and over signatures with only positive ports. We observe also that the introduction of negative ports is more important than adding directions to interfaces: directed bigraphs as per [13] can be obtained as a traced category over the category of pure bigraphs, while we cannot properly represent controls with negative ports using those with positive ports only.

2.2   Reactive Systems over Directed Bigraphs

In order to define reactive systems over bigraphs, we need to define how a parametric reaction rule (i.e., a pair of "redex-reactum" bigraphs) can be instantiated. Essentially, in the application of the rule, the "sites" of the reactum must be filled with the parameters appearing in the redex. This relation can be expressed by specifying an *instantiation map* in the rule.

Definition 7 (Instantiation map). *An* instantiation map $\eta :: \langle m, X \rangle \rightarrow \langle m', X' \rangle$ *is a pair* $\eta = (\eta^P, \eta^L)$ *where*

- $\eta^P : m' \rightarrow m$ *is a function which maps sites of the reactum to sites of the redex; for each* $j \in m'$, *it determines that the j-th site of the reactum is filled with the $\eta(j)$-th parameter of the redex.*
- $\eta^L : \left(\sum_{i=0}^{m'-1} X \right) \rightarrow X'$ *is a wiring (i.e., a link graph without nodes nor edges), which is responsible for mapping names of the redex to names of the reactum. This can be described as a pair of functions* $\eta^L = (\eta^+, \eta^-)$ *where* $\eta^+ : \left(\sum_{i=0}^{m'-1} X^+ \right) \rightarrow X'^+$ *and* $\eta^- : X'^- \rightarrow \sum_{j=0}^{m'-1} X^-$.

We can now define the dynamics of directed bigraphs, starting with the formal definition of parametric reaction rules.

Definition 8 (Parametric reaction rule). *A* parametric reaction rule *for bigraphs is a triple of the form* $(R : I \rightarrow J, R' : I' \rightarrow J, \eta :: I \rightarrow I')$ *where R is the parametric redex, R' the parametric reactum and η is an instantiation map.*

We can now define the key notion of reactive systems over directed bigraphs, which is a generalization of that in [14,25]. Let $Ag(\mathcal{K})$ be the set of *agents* (i.e., bigraphs with no inner names nor sites) over a signature $\mathcal{K}$.

Definition 9 (DBRS). *A directed bigraphical reactive system $DBG(\mathcal{K}, \mathcal{R})$ is defined by a signature $\mathcal{K}$ and a set $\mathcal{R}$ of rewriting rules.*

A DBRS $DBG(\mathcal{K}, \mathcal{R})$ induces a rewriting relation $\twoheadrightarrow \subseteq Ag(\mathcal{K}) \times Ag(\mathcal{K})$ according to the following rule:

$$\frac{\begin{array}{c} (R_L, R_R, \eta) \in \mathcal{R} \\ A = C \circ (R_L \otimes Id_Z) \circ \omega \circ (D_0 \otimes \ldots \otimes D_{m-1}) \\ A' = C \circ (R_R \otimes Id_Z) \circ \omega' \circ (D_{\eta^P(0)} \otimes \ldots \otimes D_{\eta^P(m'-1)}) \end{array}}{A \rightarrow A'}$$

where ω and ω' are called wiring maps *and are defined as follows:*

$$\omega : \sum_{i=0}^{m-1} X_i \to X \oplus Z \qquad\qquad \omega' : \sum_{j=0}^{m'-1} X_{\eta^P(j)} \to X' \oplus Z$$

$$\omega^+ : \sum_{i=0}^{m-1} X_i^+ \to X^+ \uplus Z^+ \qquad\qquad \omega'^+ : \sum_{j=0}^{m'-1} X_{\eta^P(j)}^+ \to X'^+ \uplus Z^+$$

$$\omega^- : X^- \uplus Z^- \to \sum_{i=0}^{m-1} X_i^- \qquad\qquad \omega'^- : X'^- \uplus Z^- \to \sum_{j=0}^{m'-1} X_{\eta^P(j)}^-$$

$$\omega'^+(j,x) \triangleq \begin{cases} \eta^+(j, \omega^+(\eta(j), x)) & if\ \omega^+(\eta^P(j), x) \in X^+ \\ \omega^+(\eta^P(j), x) & if\ \omega^+(\eta(j), x) \in Z^+ \end{cases}$$

$$\omega'^-(x) \triangleq (j, y)\ for\ j \in \eta^{P^{-1}}(i)\ and\ (i, y) \in \eta^-(x)$$

The difference with respect to the previous versions of BRS is that now links can descend from the redex (and reactum) into the parameters, as it is evident from the fact that redexes and reactums in rules may have generic inner interfaces (I and I'). This is very useful for representing a request flow which goes "downwards", e.g. connecting a port of a control in the redex to a port of an inner component (think of, e.g., a linked library).

However, this poses some issues when the rules are not linear. If any of D_i's is cancelled by the rewriting, the controls in it disappear as well, and we may be not able to connect some name descending from R_L or Id_Z anymore. More formally, this means that the map ω^- can be defined only if for every $x \in (X'^- \uplus Z^-)$ there are j, y such that $(\eta^P(j), y) = \eta^-(x)$. We can have two cases:

1. for some x, there are no such j, y. This means that ω is not defined and hence the rule cannot be applied.
2. for each x, there are one or more pairs (j, y) such that $(\eta^P(j), y) = \eta^-(x)$. This means that for a given source agent decomposition, there can be several ways to define ω^-, each yielding a different application of the same rule.

Overall, the presence of downward names in parameters adds a new degree of non-determinism to Directed BRSs, with respect to previous versions of BRSs.

3   Directed Bigraph Embeddings

As we have seen in the previous section, to execute or simulate a BRS it is necessary to solve the *bigraph matching problem*, that is, finding the occurrences of a redex R within a given bigraph A. More formally, this translates to finding C, Z, ω and $D = (D_0 \otimes \ldots \otimes D_{m-1})$ such that $A = C \circ (R \otimes Id_Z) \circ \omega \circ D$. C and D are called *context* and *parameter*, respectively.

If we abstract from the decomposition of the agent A in context, redex and parameter we can see how the matching problem is related to the *subgraph isomorphism* problem. Therefore, in this section we define the notions of *directed bigraph embedding*. The following definitions are taken from [16], modified to suit the definition of directed bigraphs.

Directed Link Graph. Intuitively an embedding of link graphs is a structure preserving map from one link graph (the *guest*) to another (the *host*). As one would expect from a graph embedding, this map contains a pair of injections: one for the nodes and one for the edges (i.e., a support translation). The remaining of the embedding map specifies how names of the inner and outer interfaces should be mapped into the host link graph. Outer names can be mapped to any link; here injectivity is not required since a context can alias outer names. Dually, inner names can be mapped to hyper-edges linking sets of points in the host link graph and such that every point is contained in at most one of these sets.

Definition 10 (Directed link graph embedding). *Let $G : X_G \to Y_G$ and $H : X_H \to Y_H$ be two directed link graphs. A directed link graph embedding $\phi : G \hookrightarrow H$ is a map $\phi \triangleq \phi^v \uplus \phi^e \uplus \phi^i \uplus \phi^o$, assigning nodes, edges, inner and outer names with the following constraints:*

(L1) $\phi^v : V_G \rightarrowtail V_H$ *and* $\phi^e : E_G \rightarrowtail E_H$ *are injective;*

(L2) $ctrl_G = ctrl_H \circ \phi^v$;

(L3) $\phi^i : Y_H^- \uplus X_H^+ \uplus P_H^+ \rightharpoonup X_G^+ \uplus Y_G^- \uplus P_G^+$ *defined as follows*

$$\phi^i(x) \triangleq \begin{cases} \phi^{i^-}(x) & if \ x \in Y_H^- \uplus P_H^+ \\ \phi^{i^+}(x) & if \ x \in X_H^+ \uplus P_H^+ \end{cases} \quad where \quad \begin{array}{l} \phi^{i^-} : Y_H^- \uplus P_H^+ \rightharpoonup Y_G^- \uplus P_G^+ \\ \phi^{i^+} : X_H^+ \uplus P_H^+ \rightharpoonup X_G^+ \uplus P_G^+ \\ dom(\phi^{i^+}) \cap dom(\phi^{i^-}) = \varnothing \end{array}$$

(L4) $\phi^o : X_G^- \uplus Y_G^+ \rightharpoonup E_H \uplus X_H^- \uplus Y_H^+ \uplus P_H^-$ *is a partial map s.t.:*

$$\phi^o(y) \triangleq \begin{cases} \phi^{o^-}(y) & if \ y \in X_G^- \\ \phi^{o^+}(y) & if \ y \in Y_G^+ \end{cases} \quad where \quad \begin{array}{l} \phi^{o^-} : X_G^- \rightharpoonup E_H \uplus X_H^- \uplus P_H^- \\ \phi^{o^+} : Y_G^+ \rightharpoonup E_H \uplus Y_H^+ \uplus P_H^- \end{array}$$

(L5a) $img(\phi^e) \cap img(\phi^o) = \varnothing$;

(L5b) $\forall v \in V_G, \forall j \in ar(ctrl(v)) \ . \ \phi^i((\phi^v(v), j)) = \bot$;

(L6a) $\phi^p \circ link_G^{-1}|_{E_G} = link_H^{-1} \circ \phi^e$;

(L6b) $\forall v \in V_G, \forall i \in ar(ctrl(v)) \ . \ \phi^p \circ link_G^{-1}((v, i)) = link_H^{-1} \circ \phi^{port}((v, i))$;

(L7) $\forall p \in dom(\phi^i) : link_H(p) = (\phi^o \uplus \phi^e)(link_G \circ \phi^i(p))$.

where $\phi^p \triangleq \phi^{i^+} \uplus \phi^{o^-} \uplus \phi^{port}$ and $\phi^{port} : P_G \rightarrowtail P_H$ is $\phi^{port}(v, i) \triangleq (\phi^v(v), i)$.

The first three conditions are on the single sub-maps of the embedding. Conditions (L5a) and (L5b) ensures that no components (except for outer names) are identified; condition (L6a) imposes that points connected by the image of an edge are all covered. Finally, conditions (L2), (L6b) and (L7) ensure that the guest structure is preserved i.e. node controls and point linkings are preserved.

Place Graph. Like link graph embeddings, place graph embeddings are just a structure preserving injective map from nodes along with suitable maps for the inner and outer interfaces. In particular, a site is mapped to the set of sites and nodes that are "put under it" and a root is mapped to the host root or node that is "put over it" splitting the host place graphs in three parts: the guest image, the context and the parameter (which are above and below the guest image).

Definition 11 (Place graph embedding [16, Def 7.5.4]). *Let $G : n_G \to m_G$ and $H : n_H \to m_H$ be two place graphs. A place graph embedding $\phi : G \hookrightarrow H$ is a map $\phi \triangleq \phi^{\mathsf{v}} \uplus \phi^{\mathsf{s}} \uplus \phi^{\mathsf{r}}$ (assigning nodes, sites and roots respectively) such that:*

(P1) $\phi^{\mathsf{v}} : V_G \rightarrowtail V_H$ *is injective;*
(P2) $\phi^{\mathsf{s}} : n_G \rightarrowtail \wp(n_H \uplus V_H)$ *is fully injective;*
(P3) $\phi^{\mathsf{r}} : m_G \to V_H \uplus m_H$ *in an arbitrary map;*
(P4) $img(\phi^{\mathsf{v}}) \cap img(\phi^{\mathsf{r}}) = \emptyset$ *and* $img(\phi^{\mathsf{v}}) \cap \bigcup img(\phi^{\mathsf{s}}) = \emptyset$*;*
(P5) $\forall r \in m_G : \forall s \in n_G : prnt^*_H \circ \phi^{\mathsf{r}}(r) \cap \phi^{\mathsf{s}}(s) = \emptyset$*;*
(P6) $\phi^{\mathsf{c}} \circ prnt_G^{-1}\big|_{V_G} = prnt_H^{-1} \circ \phi^{\mathsf{v}}$*;*
(P7) $ctrl_G = ctrl_H \circ \phi^{\mathsf{v}}$*;*
(P8) $\forall c \in n_G \uplus V_G : \forall c' \in \phi^{\mathsf{c}}(c) : (\phi^{\mathsf{f}} \circ prnt_G)(c) = prnt_H(c')$*;*

where $prnt^*_H(c) = \bigcup_{i<\omega} prnt^i(c)$, $\phi^{\mathsf{f}} \triangleq \phi^{\mathsf{v}} \uplus \phi^{\mathsf{r}}$, and $\phi^{\mathsf{c}} \triangleq \phi^{\mathsf{v}} \uplus \phi^{\mathsf{s}}$.

These conditions follow the structure of Definition 10, the main difference is (P5) which states that the image of a root cannot be the descendant of the image of another. Conditions (P1), (P2) and (P3) are on the three sub-maps composing the embedding; (P4) and (P5) ensure that no components are identified; (P6) imposes surjectivity on children and the last two conditions require the guest structure to be preserved by the embedding.

Directed Bigraph. Finally, a directed bigraph embedding can be defined as a pair composed by an directed link graph embedding and a place graph embedding, with a consistent interplay of these two structures. The interplay is captured by two additional conditions ensuring that points (resp. handles) in the image of guest upward (resp. downward) inner names reside in some parameter defined by the place graph embedding (i.e. descends from the image of a site).

Definition 12 (Directed bigraph embedding). *Let $G : \langle n_G, X_G \rangle \to \langle m_G, Y_G \rangle$ and $H : \langle n_H, X_H \rangle \to \langle m_H, Y_H \rangle$ be two directed bigraphs. A directed bigraph embedding is a map $\phi : G \hookrightarrow H$ given by a place graph embedding $\phi^P : G^P \hookrightarrow H^P$ and a link graph embedding $\phi^L : G^L \hookrightarrow H^L$ subject to the following constraints:*

(B1) $dom(\phi^{i^+}) \subseteq X_H^+ \uplus \{(v,i) \in P_H^+ \mid \exists s \in n_G, k \in \mathbb{N} : prnt_H^k(v) \in \phi^s(s)\}$*;*
(B2) $img(\phi^{o^-}) \subseteq X_H^- \uplus \{(v,i) \in P_H^- \mid \exists s \in n_G, k \in \mathbb{N} : prnt_H^k(v) \in \phi^s(s)\}$*.*

4 Implementing the Embedding Problem in CSP

In this Section we present a constraint satisfaction problem that models the directed bigraph embedding problem. The encoding is based solely on integer linear constraints and is proven to be sound and complete.

Initially, we present the encoding for the directed link graph embedding problem and for the place graph embedding problem. Then we combine them providing some additional "gluing constraints" to ensure the consistency of the two sub-problems. The resulting encoding contains 37 constraint families (reflecting the complexity of the problem definition, see Sect. 3); hence we take advantage of the orthogonality of link and place structures for the sake of both exposition and adequacy proofs. We observe that the overall number of variables and constraints produced by the encoding is polynomially bounded with respect to the size of the involved bigraphs, i.e., the number of nodes and edges.

4.1 Directed Link Graphs

Let us fix the guest and host bigraphs $G : X_G \to Y_G$ and $H : X_H \to Y_H$. We characterize the embeddings of G into H as the solutions of a suitable *multi-flux problem* which we denote as $\mathrm{DLGE}[G, H]$. The main idea is to see the host points (i.e. positive ports, upward inner names and downward outer names) as sources, and the handles (i.e. edges, negative ports, upward outer names and downward inner names) as sinks (see Fig. 2). Each point outputs a flux unit and each handle inputs one unit for each point it links. Units flow towards each point handle following H edges and optionally taking a "detour" along the linking structure of the guest G (provided that some conditions about structure preservation are met). The formal definition of the flux problem is in Fig. 3.

The flux network reflects the linking structure and contains an edge connecting each point to its handle; these edges have an integer capacity limited to 1 and
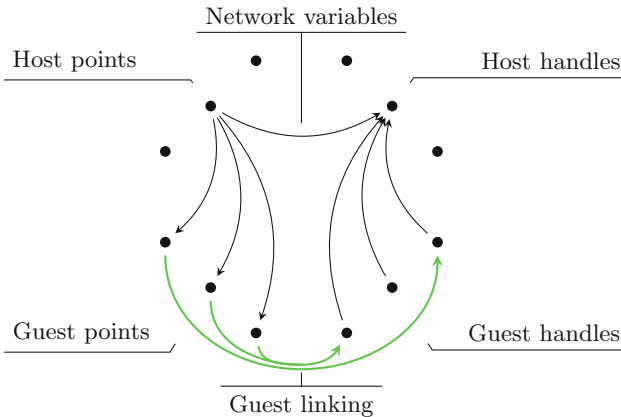
Fig. 2. Schema of the multi-flux network encoding.

$$N_{h,h'} \in \{0,\dots,|link_H^{-1}(h')|\} \qquad \begin{array}{l} h \in E_G \uplus Y_G^+ \uplus X_G^- \uplus P_G^-, \\ h' \in E_H \uplus Y_H^+ \uplus X_H^- \uplus P_H^- \end{array} \qquad (3)$$

$$N_{p,h'} \in \{0,1\} \qquad \begin{array}{l} h' \in E_H \uplus Y_H^+ \uplus X_H^- \uplus P_H^-, \\ p \in link_H^{-1}(h') \end{array} \qquad (4)$$

$$N_{p,p'} \in \{0,1\} \qquad \begin{array}{l} p' \in X_G^+ \uplus P_G^+ \uplus Y_G^-, \\ p \in X_H^+ \uplus P_H^+ \uplus Y_H^- \end{array} \qquad (5)$$

$$F_{h,h'} \in \{0,1\} \qquad \begin{array}{l} h \in E_G \uplus Y_G^+ \uplus X_G^- \uplus P_G^-, \\ h' \in E_H \uplus Y_H^+ \uplus X_H^- \uplus P_H^- \end{array} \qquad (6)$$

$$\sum_k N_{p,k} = 1 \qquad p \in X_H^+ \uplus P_H^+ \uplus Y_H^- \qquad (7)$$

$$\sum_k N_{k,h} = |link_H^{-1}(h)| \qquad h \in E_H \uplus Y_H^+ \uplus X_H^- \uplus P_H^- \qquad (8)$$

$$\sum_k N_{h,k} = \sum_{p \in link_G^{-1}(h)} \sum_k N_{k,p} \qquad h \in E_G \uplus Y_G^+ \uplus X_G^- \uplus P_G^- \qquad (9)$$

$$\sum_k N_{k,p} \leq 1 \qquad p \in X_G^+ \uplus P_G^+ \uplus Y_G^- \qquad (10)$$

$$N_{p,p'} = 0 \qquad p' \in P_G^+,\ p \in X_H^+ \uplus Y_H^- \qquad (11)$$

$$\frac{N_{h,h'}}{|link_H^{-1}(h')|} \leq F_{h,h'} \leq N_{h,h'} \qquad \begin{array}{l} h \in E_G \uplus Y_G^+ \uplus X_G^- \uplus P_G^-, \\ h' \in E_H \uplus Y_H^+ \uplus X_H^- \uplus P_H^-, \\ link_G^{-1}(h) \neq \varnothing,\ link_H^{-1}(h') \neq \varnothing \end{array} \qquad (12)$$

$$N_{p,p'} \leq F_{h,h'} \qquad \begin{array}{l} h \in E_G \uplus Y_G^+ \uplus X_G^- \uplus P_G^-, \\ h' \in E_H \uplus Y_H^+ \uplus X_H^- \uplus P_H^-, \\ p \in link_G^{-1}(h),\ p' \in link_H^{-1}(h') \end{array} \qquad (13)$$

$$F_{h,h'} \leq \sum_{\substack{p \in link_G^{-1}(h) \\ p' \in link_H^{-1}(h')}} N_{p,p'} \qquad \begin{array}{l} h \in E_G \uplus Y_G^+ \uplus X_G^- \uplus P_G^-, \\ h' \in E_H \uplus Y_H^+ \uplus X_H^- \uplus P_H^-, \\ link_G^{-1}(h) \neq \varnothing, link_H^{-1}(h') \neq \varnothing \end{array} \qquad (14)$$

$$\sum_k F_{h,k} = 1 \qquad h \in E_G \uplus Y_G^+ \uplus X_G^- \uplus P_G^- \qquad (15)$$

$$N_{p,h'} + F_{h,h'} \leq 1 \qquad \begin{array}{l} h \in E_G,\ h' \in E_H \uplus Y_H^+ \uplus X_H^- \uplus P_H^-, \\ p \in link_H^{-1}(h') \end{array} \qquad (16)$$

$$F_{h,h'} + F_{h'',h'} + F_{h''',h'} + F_{h^{iv},h'} \leq 1 \qquad \begin{array}{l} h \in E_G,\ h' \in Y_H^+ \uplus X_H^- \uplus P_H^-, \\ h'' \in Y_G^+,\ h''' \in X_G^-,\ h^{iv} \in P_G^- \end{array} \qquad (17)$$

$$F_{h,h'} = 0 \qquad h \in E_G,\ h' \in Y_H^+ \uplus X_H^- \uplus P_H^- \qquad (18)$$

$$F_{h,h'} \leq 1 \qquad h \in E_G \uplus Y_G^+ \uplus X_G^- \uplus P_G^-,\ h' \in E_H \qquad (19)$$

$$N_{p,p'} = 0 \qquad \begin{array}{l} v \in V_G,\ v' \in V_H, \\ ctrl_G(v) = ctrl_H(v) = c,\ i \neq i' \leq c, \\ p = (v,i) \in P_G^+ \uplus P_G^-, \\ p' = (v',i') \in P_H^+ \uplus P_H^- \end{array} \qquad (20)$$

$$N_{p,p'} = 0 \qquad \begin{array}{l} v \in V_G,\ v' \in V_H, \\ ctrl_G(v) \neq ctrl_H(v) \\ p = (v,i) \in P_G^+ \uplus P_G^-, \\ p' = (v',i') \in P_H^+ \uplus P_H^- \end{array} \qquad (21)$$

$$\sum_{j \leq c} N_{(v,j),(v',j)} = c \cdot N_{p,p'} \qquad \begin{array}{l} v \in V_G,\ v' \in V_H, \\ ctrl_G(v) = ctrl_H(v) = c,\ i \leq c, \\ p = (v,i) \in P_G^+ \uplus P_G^-, \\ p' = (v',i') \in P_H^+ \uplus P_H^- \end{array} \qquad (22)$$

$$N_{p,p'} = 0 \qquad p \in P_H^+,\ p' \in X_G^+ \uplus Y_G^- \qquad (23)$$

Fig. 3. Constraints of DLGE$[G,H]$.

are represented by the variables defined in (4). The remaining edges of the network are organised in two complete biparted graphs: one between guest and host handles and one between guest and host points. Edges of the first sub-network are described by the variables in (3) and their capacity is bounded by the number of points linked by the host handle since this is the maximum acceptable flux and corresponds to the case where each point passes through the same hyper-edge of the guest link graph. Edges of the second sub-network are described by the variables in (5) and, like the first group of links, have their capacity limited to 1; to be precise, some of these variables will never assume a value different from 0 because guest points can receive flux from anything but the host ports (as expressed by constraint (11)). Edges for the link structure of the guest are presented implicitly in the flux preservation constraints (see constraint (9)). In order to fulfil the injectivity conditions of link embeddings, some additional *flux variables* (whereas the previous are *network variables*) are defined by (6). These are used to keep track and separate each flux on the bases of the points handle.

The constraint families (7) and (8) define the outgoing and ingoing flux of host points and handles respectively. The former has to send exactly one unit considering every edge they are involved with and the latter receive one unit for each of their point regardless if this unit comes from the point directly or from a handle of the guest.

The linking structure of the guest graph is encoded by the constraint family (9) which states that flux is preserved while passing through the guest i.e. the output of each handle has to match the overall input of the points it connects.

Constraints (10), (11), (20), (21), (22) and (23) shape the flux in the sub-network linking guest and host points. Specifically, (10) requires that each point from the guest receives at most one unit; this is needed when we want to be able to embed a redex where some points (e.g. upward inner names) would not match with an entity of the agent and (those points) would be deleted anyway when composing the resulting agent back. Constraints (11), (20) and (21) disable edges between guest ports and host inner names, between mismatching ports of matching nodes and between ports of mismatching nodes. Constraint (23) ensures that ascending inner names or descending outer names of the redex are not matched with positive ports of the agent. Finally, the flux of ports of the same node has to act compactly, as expressed by (22): if there is flux between the i-th ports of two nodes, then there should be flux between every other ports.

Constraints (12), (13) and (14) relate flux and network variables ensuring that the formers assume a true value if, and only, if there is actual flux between the corresponding guest and host handles. In particular, (13) propagates the information about the absence of flux between handles disabling the sub-network linking handles points and, *vice versa*, (14) propagates the information in the other way disabling flux between handles if there is no flux between their points.

The remaining constraints prevent fluxes from mixing. Constraint (15) requires guest handles to send their output to exactly one destination thus rendering the sub-network between handles a function assigning guest handles to host handles. This mapping is subject to some additional conditions when edges

are involved: (18) and (19) ensure that the edges are injectively mapped to edges only, (17) forbids host outer names to receive flux from an edge and an outer name at the same time. Finally, (16) states that the output of host points cannot bypass the guest if there is flux between its handle and an edge from the guest.

Adequacy. Let N be a solution of $\mathrm{DLGE}[G, H]$. The corresponding link graph embedding $\phi : G \hookrightarrow H$ is defined as follows:

$$\phi^v(v) \triangleq v' \in V_H \ \text{if} \ \exists i : N_{(v,i),(v',i)} = 1 \qquad \phi^e(e) \triangleq e' \in E_H \ \text{if} \ F_{e,e'} = 1$$

$$\phi^i(x) \triangleq \begin{cases} \phi^{i^-}(x) & if \ x \in Y_H^- \uplus P_H^+ \\ \phi^{i^+}(x) & if \ x \in X_H^+ \uplus P_H^+ \end{cases} \qquad \phi^o(y) \triangleq \begin{cases} \phi^{o^-}(y) & if \ y \in X_G^- \\ \phi^{o^+}(y) & if \ y \in Y_G^+ \end{cases}$$

where

$$\phi^{o^-}(y) \triangleq y' \in X_H^- \uplus P_H^- \ \text{if} \ F_{y,y'} = 1 \qquad \phi^{o^+}(y) \triangleq y' \in Y_H^+ \uplus P_H^- \ \text{if} \ F_{y,y'} = 1,$$

$$\phi^{i^-}(x) \triangleq x' \in Y_G^- \uplus P_G^+ \ \text{if} \ N_{x,x'} = 1, \qquad \phi^{i^+}(x) \triangleq x' \in X_G^+ \uplus P_G^+ \ \text{if} \ N_{x,x'} = 1$$

and $dom(\phi^{i^+}) \cap dom(\phi^{i^-}) = \varnothing$.

It is easy to check that these components of ϕ are well-defined and compliant with Definition 10.

On the other way around, let $\phi : G \hookrightarrow H$ be a link graph embedding. The corresponding solution N of $DLGE[G, H]$ is defined as follows:

$$N_{p,p'} \triangleq \begin{cases} 1 & if \ p \in X_H^+ \uplus Y_H^- \land p' = \phi^i(p) \\ 1 & if \ p' = (v, i) \in P_G^+ \land p = (\phi^v(v), i) \\ 0 & otherwise \end{cases}$$

$$N_{p,h'} \triangleq \begin{cases} 1 & if \ h' = link_H(p) \land \nexists p' : N_{p,p'} = 1 \\ 0 & otherwise \end{cases}$$

$$N_{h,h'} \triangleq \begin{cases} 1 & if \ h' \in E_H \land h \in E_G \land h' = \phi^e(h) \\ 1 & if \ h' \in Y_H^+ \uplus X_H^- \land h \in Y_G^+ \uplus X_G^- \land h' = \phi^o(h) \\ 1 & if \ h = (v, i) \in P_G^- \land h' = (\phi^v(v), i) \\ 0 & otherwise \end{cases}$$

$$F_{h,h'} = 1 \overset{\triangle}{\Longleftrightarrow} N_{h,h'} \neq 0$$

Every constraint of $DLGE[G, H]$ is satisfied by the solution just defined.

The constraint satisfaction problem in Fig. 3 is sound and complete with respect to the directed link graph embedding problem given in Definition 10.

Proposition 1 (Adequacy of DLGE). *For any two concrete directed link graphs G and H, there is a bijective correspondence between the directed link graph embeddings of G into H and the solutions of* $\mathrm{DLGE}[G, H]$.

4.2   Place Graphs

Let us fix the guest and host place graphs: $G : n_G \to m_G$ and $H : n_H \to m_H$. We characterize the embeddings of G into H as the solutions of the constraint satisfaction problem in Fig. 4. The problem is a direct encoding of Definition 11 as a matching problem presented, as usual, as a bipartite graph. Sites, nodes and roots of the two place graphs are represented as nodes and partitioned into the guest and the host ones. For convenience of exposition, the graph is complete.

$$M_{h,g} \in \{0,1\} \qquad \begin{aligned} &g \in n_G \uplus V_G \uplus m_G, \\ &h \in n_H \uplus V_H \uplus m_H \end{aligned} \tag{24}$$

$$M_{h,g} = 0 \qquad g \in n_G \uplus V_G, \ h \in m_H \tag{25}$$

$$M_{h,g} = 0 \qquad g \in V_G \uplus m_G, \ h \in n_H \tag{26}$$

$$M_{h,g} = 0 \qquad \begin{aligned} &g \in V_G, \ h \in V_H, \\ &\mathit{ctrl}_G(g) \neq \mathit{ctrl}_H(h) \end{aligned} \tag{27}$$

$$M_{h,g} = 0 \qquad \begin{aligned} &g \in m_G, \ h \notin m_H, \\ &v \in \mathit{prnt}_H^*(h) \cap V_G, \\ &\mathit{ctrl}_G(v) \notin \Sigma_a \end{aligned} \tag{28}$$

$$M_{h,g} \leq M_{h',g'} \qquad \begin{aligned} &g \notin m_G, \ g' \in \mathit{prnt}_G(g), \\ &h \notin m_H, \ h' \in \mathit{prnt}_H(h) \end{aligned} \tag{29}$$

$$\sum_{h \in V_H \uplus m_H} M_{h,g} = 1 \qquad g \in m_G \tag{30}$$

$$\sum_{h \in n_H \uplus V_H} M_{h,g} = 1 \qquad g \in V_G \tag{31}$$

$$m_G \cdot \sum_{g \in n_G \uplus V_G} M_{h,g} + \sum_{g \in m_G} M_{h,g} \leq m_G \qquad h \in V_H \tag{32}$$

$$|\mathit{prnt}_H^{-1}(h)| \cdot M_{h,g} \leq \sum_{\substack{h' \in \mathit{prnt}_H^{-1}(h), \\ g' \in \mathit{prnt}_G^{-1}(g)}} M_{h',g'} \qquad g \in V_G, \ h \in V_H \tag{33}$$

$$|\mathit{prnt}_G^{-1}(g) \setminus n_G| \cdot M_{h,g} \leq \sum_{\substack{h' \in \mathit{prnt}_H^{-1}(h) \setminus n_h, \\ g' \in \mathit{prnt}_G^{-1}(g) \setminus n_g}} M_{h',g'} \qquad g \in m_G, \ h \in V_H \tag{34}$$

$$M_{h,g} + \sum_{h' \in \mathit{prnt}_H^*(h), g' \in m_G} M_{h',g'} \leq 1 \qquad g \in V_G, \ h \in V_H \tag{35}$$

Fig. 4. Constraints of PGE$[G, H]$.

Edges are modelled by the boolean variables defined in (24); these are the only variables used by the problem. So far, a solution is nothing more than a relation between the components of guest and host containing only those pairs connected by an edge assigned a non-zero value. To capture exactly those assignments that are actual place graph embeddings some conditions have to be imposed.

Constraints (25) and (26) prevent roots and sites from the host to be matched with nodes or sites and nodes or roots respectively. (27) disables matching between nodes decorated with different controls. Constraint (28) prevents any matching for host nodes under a passive context (i.e. have an ancestor labelled with a passive control). (29) propagates the matching along the parent map from children to parents. Constraints (30) and (31) ensure that the matching is a function when restricted to guest nodes and roots (the codomain restriction follows by (25) and (26)). (32) says that if a node from the host cannot be matched with a root or a node/site from the guest at the same time; moreover, if the host node is matched with a node then it cannot be matched to anything else.

The remaining constraints are the counterpart of (29) and propagate matchings from parents to children. (33) applies to matchings between nodes and says that if parents are matched, then children from the host node are covered by children from the guest node. In particular, the matching is a perfect assignment when restricted to guest children that are nodes (because of (32)) and is a surjection on those that are sites. (34) imposes a similar condition on matchings between guest roots and host nodes. Specifically, it says that the matching has to cover child nodes from the guest (moreover, it is injective on them) leaving child sites to match whatever remains ranging from nothing to all unmatched children. Finally, (35) prevents matching from happening inside a parameter.

Adequacy. Let $\boldsymbol{M}$ be a solution of $\mathrm{PGE}[G, H]$. The corresponding place graph embedding $\phi : G \hookrightarrow H$ is defined as follows:

$$\phi^{\mathsf{v}}(g) \triangleq h \in V_H \text{ if } \exists i : M_{h,g} = 1 \qquad \phi^{\mathsf{s}}(g) \triangleq \{h \in n_h \uplus V_H \mid M_{h,g} = 1\}$$

$$\phi^{\mathsf{r}}(g) \triangleq h \in m_H \uplus V_H \text{ if } M_{h,g} = 1$$

These components of ϕ are well-defined and compliant with Definition 11.
On the opposite direction, let $\phi : G \hookrightarrow H$ be a place graph embedding. The corresponding solution $\boldsymbol{M}$ of $\mathrm{PGE}[G, H]$ is defined as aside. It is easy to check that every constraint of $\mathrm{PGE}[G, H]$ is satisfied by this solution. Hence, the constraint satisfaction problem in Fig. 4 is sound and complete with respect to the place graph embedding problem (Definition 11).

$$M_{h,g} \triangleq \begin{cases} 1 & \text{if } g \in V_G \wedge h = \phi^{\mathsf{v}}(g) \\ 1 & \text{if } g \in m_G \wedge h = \phi^{\mathsf{r}}(g) \\ 1 & \text{if } g \in n_G \wedge h \in \phi^{\mathsf{s}}(g) \\ 0 & \text{otherwise} \end{cases}$$

Proposition 2 (Adequacy of PGE). *For any two concrete place graphs G and H, there is a bijective correspondence between the place graph embeddings of G into H and the solutions of $\mathrm{PGE}[G, H]$.*

4.3   Bigraphs

Let $G : \langle n_G, X_G \rangle \to \langle m_G, Y_G \rangle$ and $H : \langle n_H, X_H \rangle \to \langle m_H, Y_H \rangle$ be two bigraphs. By taking advantage of the orthogonality of the link and place structures we

$$M_{v,v'} = N_{p,p'} \qquad \begin{aligned} & v \in V_H, \ v' \in V_G, \\ & p = (v,k) \in P_H^+, \ p' = (v',k) \in P_G^+ \end{aligned} \qquad (36)$$

$$M_{v,v'} = F_{h,h'} \qquad v \in V_H, \ v' \in V_G, \ h \in P_G^-, \ h' \in P_H^- \qquad (37)$$

$$\sum_{p' \in X_G^+} N_{p,p'} \leq \sum_{h \in prnt_H^*(v), g \in n_G} M_{h,g} \qquad v \in V_H, \ p = (v,k) \in P_H^+ \qquad (38)$$

$$\sum_{h \in X_G^-} F_{h,h'} \leq \sum_{h \in prnt_H^*(v), g \in n_G} M_{h,g} \qquad v \in V_H, \ h' = (v,k) \in P_H^- \qquad (39)$$

Fig. 5. Constraints of $\mathrm{DBGE}[G,H]$.

can define the constraint satisfaction problem capturing bigraph embeddings by simply composing the constraints given above for the link and place graph embeddings and by adding four consistency constraints to relate the solutions of the two problems. These additional constraint families are reported in Fig. 5. The families (36) and (37) ensure that solutions for $\mathrm{DLGE}[G,H]$ and $\mathrm{PGE}[G,H]$ agree on nodes since the map $\phi^\vee$ has to be shared by the corresponding link and place embeddings. The families (38) and (39) respectively, ensure that positive ports (negative ports resp.) are in the same image as upward inner names (downward inner names resp.) only if their node is part of the parameter i.e. only if it is matched to a site from the guest or it descends from a node that is so.

Conditions (38) and (39) correspond exactly to (B1) and (B2). It thus follows from Propositions 1 and 2 that the CSP defined by Figs. 3 to 5 is sound and complete with respect to the bigraph embedding problem given in Definition 12.

Theorem 1 (Adequacy of BGE). *For any two concrete bigraphs G and H, there is a bijective correspondence between the bigraph embeddings of G into H and the solutions of $\mathrm{DBGE}[G,H]$.*

5   Experimental Results

The reduction algorithm presented in the previous section has been successfully integrated into jLibBig, an extensible Java library for manipulating bigraphs and bigraphical reactive systems which can be used for implementing a wide range of tools and it can be adapted to support several extensions of bigraphs [24].

The proposed algorithm is implemented by extending the data structures and the models for pure bigraphs to suit our definition of directed bigraphs.

In this section we test our implementation by simulating a system in which we want to track the position and the movements of a fleet of vehicles inside a territory divided in "zones", which are accessible via "roads". The rewriting rule in question and an example agent can be found in Fig. 6.

We evaluate the running time of the different components of our algorithm: model construction, CSP resolution, building of the actual embedding and execution of the rewriting rule. Moreover, we want to evaluate how these performances scale while increasing the size of the agent. The parameters used to build the
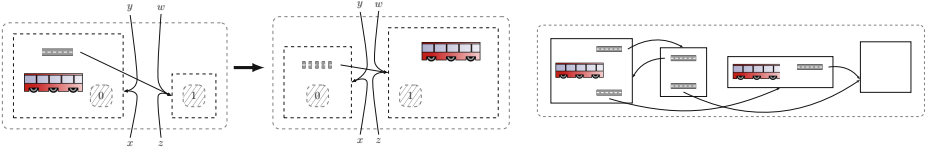
Fig. 6. Rewriting rule (left) and example of an agent (right) for the test cases.

tests are: number of zones, number of cars and "connectivity degree". The last parameter is a number between 1 and 100 representing the probability of the existence of a connection between two nodes; a value of 100 means that every node is connected to all its neighbours.
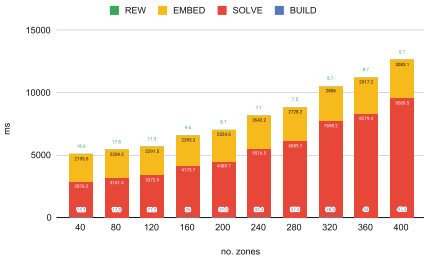
Fig. 7. Execution times vs. number of zones, 70 cars and 100% connectivity.
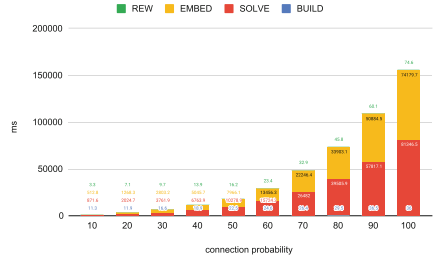
Fig. 8. Execution times vs. connectivity, 11×11 grid and 70 cars.

We consider the following kinds of tests:

1. varying number of zones, with fixed number of cars and connectivity degree;
2. varying connectivity degree, with fixed numbers of zones and cars.

Each test case is made up of four groups of instances, where for each group we choose an increasing value for their fixed parameters. For each group we choose ten values for its variable parameter. The instances generation works as follows: for each test case and for each group of that particular test case we generate ten random instances for each combination of the values of the fixed parameters and the variable one. We then take the average of the running times of those ten random instances. At the end of the process, for each group we have tested 100 instances, 10 for each value of the variable parameter, so 400 instances for each test case and 1200 in total.

All tests have been performed on an Intel Core i7-4710HQ (4 cores at 3.5 GHz), 8 GB of RAM running on ArchLinux with kernel 5.5.2 and using OpenJDK 12.

We briefly review the results obtained from these tests and refer to the companion technical report [6] for more details.

Time vs. Number of Zones. In this case we evaluate how our implementation scales with an increasing number of zones; see Fig. 7. We can see that the running time grows exponentially, especially the resolution time. Similarly to the previous test case, the time spent building the CSP and applying the reaction rule is negligible even though we can see that the time necessary to build the CSP increases linearly with the grid size. We can also observe that there is no correlation between the rewriting time and the number of zones.

Time vs. Connectivity Degree. In this case we evaluate how our implementation scales with an increasing connectivity degree; see Fig. 8. We can see that the running time scales exponentially, no matter the grid size or the number of cars. Once again, we see that although increasing, the time spent building the model and applying the rewriting rule is negligible.

6   Conclusions and Future Works

In this paper, we have presented a new version of *directed* bigraphs and bigraphic reactive systems, which subsume previous versions (such as Milner's bigraphs). For this kind of bigraphs we have provided a sound and complete algorithm for solving the embedding problem, based on a constraint satisfaction problem. The resulting model is compact and the a number of variables and linear constraints are polynomially bounded by the size of the guest and host bigraphs. Differently from existing solutions, this algorithm applies also to non-ground hosts.

The algorithm has been successfully integrated into jLibBig, an extensible library for manipulating bigraphical reactive systems. The empirical evaluation of the implementation of our algorithm in jLibBig looks promising. It cannot be considered a rigorous experimental validation yet, mainly because performance depends on the implementation and the solver and the model is not optimized for any specific solver. Moreover, up to now there are no "official" (or "widely recognized") benchmarks, nor any other algorithms or available tools that solve the directed bigraph embedding problem, to compare with.

The proposed approach offers great flexibility: it can be easily applied also to other extensions of bigraphs and directed bigraphs. An interesting direction for future work would be to extend the algorithm also to stochastic and probabilistic bigraphs [18]; this would offer useful modelling and verification tools for quantitative aspects, e.g. for systems biology [2,8]. Approximated and weighted embeddings are supported in jLibBig, but still as experimental feature. In fact, the theoretical foundations of these extensions have not been fully investigated yet, suggesting another line of research.

References

1. Bacci, G., Grohmann, D., Miculan, M.: DBtk: a toolkit for directed bigraphs. In: Kurz, A., Lenisa, M., Tarlecki, A. (eds.) CALCO 2009. LNCS, vol. 5728, pp. 413–422. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03741-2_28

2. Bacci, G., Grohmann, D., Miculan, M.: A framework for protein and membrane interactions. arXiv preprint arXiv:0911.4513 (2009)
3. Bacci, G., Miculan, M., Rizzi, R.: Finding a forest in a tree. In: Maffei, M., Tuosto, E. (eds.) TGC 2014. LNCS, vol. 8902, pp. 17–33. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45917-1_2
4. Bundgaard, M., Glenstrup, A.J., Hildebrandt, T., Højsgaard, E., Niss, H.: Formalizing higher-order mobile embedded business processes with binding bigraphs. In: Lea, D., Zavattaro, G. (eds.) COORDINATION 2008. LNCS, vol. 5052, pp. 83–99. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68265-3_6
5. Burco, F., Miculan, M., Peressotti, M.: Towards a formal model for composable container systems. In: Proceedings of the SAC, pp. 173–175. ACM (2020)
6. Chiapperini, A., Miculan, M., Peressotti, M.: A CSP implementation of the directed bigraph embedding problem. CoRR abs/2003.10209 (2020)
7. Damgaard, T.C., Glenstrup, A.J., Birkedal, L., Milner, R.: An inductive characterization of matching in binding bigraphs. Form. Asp. Comp. **25**(2), 257–288 (2013)
8. Damgaard, T.C., Højsgaard, E., Krivine, J.: Formal cellular machinery. Electr. Not. Theoret. Comput. Sci. **284**, 55–74 (2012)
9. Faithfull, A.J., Perrone, G., Hildebrandt, T.T.: Big red: a development environment for bigraphs. Electr. Commun. EASST **61**, 1–10 (2013)
10. Gassara, A., Rodriguez, I.B., Jmaiel, M., Drira, K.: Executing bigraphical reactive systems. Discr. Appl. Math. **253**, 73–92 (2019)
11. Glenstrup, A.J., Damgaard, T.C., Birkedal, L., Højsgaard, E.: An implementation of bigraph matching. IT University of Copenhagen p. 22 (2007)
12. Grohmann, D.: Security, cryptography and directed bigraphs. In: Ehrig, H., Heckel, R., Rozenberg, G., Taentzer, G. (eds.) ICGT 2008. LNCS, vol. 5214, pp. 487–489. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87405-8_41
13. Grohmann, D., Miculan, M.: Directed bigraphs. ENTCS **173**, 121–137 (2007)
14. Grohmann, D., Miculan, M.: Reactive systems over directed bigraphs. In: Caires, L., Vasconcelos, V.T. (eds.) CONCUR 2007. LNCS, vol. 4703, pp. 380–394. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74407-8_26
15. Grohmann, D., Miculan, M.: Controlling resource access in directed bigraphs. ECE-ASST **10**, 1–21 (2008)
16. Højsgaard, E., et al.: Bigraphical languages and their simulation. IT University of Copenhagen, Programming, Logic, and Semantics (2012)
17. Jensen, O.H., Milner, R.: Bigraphs and transitions. In: ACM SIGPLAN Notices, vol. 38, pp. 38–49. ACM (2003)
18. Krivine, J., Milner, R., Troina, A.: Stochastic bigraphs. Electr. Not. Theor. Comput. Sci. **218**, 73–96 (2008)
19. Mansutti, A., Miculan, M., Peressotti, M.: Distributed execution of bigraphical reactive systems. ECEASST **71**, 1–21 (2014)
20. Mansutti, A., Miculan, M., Peressotti, M.: Multi-agent systems design and prototyping with bigraphical reactive systems. In: Magoutis, K., Pietzuch, P. (eds.) DAIS 2014. LNCS, vol. 8460, pp. 201–208. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43352-2_16
21. Mansutti, A., Miculan, M., Peressotti, M.: Towards distributed bigraphical reactive systems. In: Echahed, R., Habel, A., Mosbah, M. (eds.) Proceedings of the GCM 2014, p. 45 (2014). workshop version
22. Miculan, M., Peressotti, M.: Bigraphs reloaded: a presheaf presentation. Technical report UDMI/01/2013, Dept. of Mathematics and Computer Science, Univ. of Udine (2013). http://www.dimi.uniud.it/miculan/Papers/UDMI012013.pdf

23. Miculan, M., Peressotti, M.: A CSP implementation of the bigraph embedding problem. CoRR abs/1412.1042 (2014)
24. Miculan, M., Peressotti, M.: jLibBig: a library for bigraphical reactive systems, November 2015. https://github.com/bigraphs/jlibbig
25. Milner, R.: The Space and Motion of Communicating Agents. Cambridge University Press, Cambridge (2009)
26. Perrone, G., Debois, S., Hildebrandt, T.T.: Bigraphical refinement. In: Refine@FM. EPTCS, vol. 55, pp. 20–36 (2011)
27. Perrone, G., Debois, S., Hildebrandt, T.T.: A model checker for bigraphs. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 1320–1325 (2012)
28. Sahli, H., Ledoux, T., Rutten, É.: Modeling self-adaptive fog systems using bigraphs. In: Proceedings of the FOCLASA, Oslo, Norway, pp. 1–16, September 2019
29. Sevegnani, M., Calder, M.: Bigraphs with sharing. Theoretical Computer Science **577**, 43–73 (2015)
30. Sevegnani, M., Unsworth, C., Calder, M.: A SAT based algorithm for the matching problem in bigraphs with sharing. University of Glasgow, Technical report (2010)