Natural Computing (2023) 22:3–12
https://doi.org/10.1007/s11047-022-09900-7

Solving subset sum by spiking neural P systems with astrocytes producing calcium

Bogdan Aman[1,2]

Accepted: 14 June 2022 / Published online: 2 July 2022

Abstract
We consider spiking neural P systems with astrocytes producing calcium that differs from the standard spiking neural P systems in several ways: we have a new type of resources called calcium unit alongside the standard spike, we have a new type of place called astrocyte alongside the standard neurons and we also have dedicated synapses for communicating calcium units alongside those used for communicating spikes. In order to illustrate how these systems work we give an example of a spiking neural P system with astrocytes producing calcium that generates all even numbers using only two neurons and one astrocyte and without using either forgetting rules or delay in the evolution rules. We prove that in this framework, we can provide semi-uniform and uniform constructions of spiking neural P system with astrocytes producing calcium working in a non-deterministic way that are able to solve the subset sum problem in a polynomial number of steps without using either forgetting rules or delay in the evolution rules.

Keywords Spiking neural P systems · Astrocyte · Calcium unit · Efficiency · Subset sum problem

1 Introduction

Membrane computing (Paun 2002) is a well-known branch of natural computing (Rozenberg et al. 2012) that aims to obtain formal models (membrane systems - also called P systems) inspired by the way the living cells function, are structured and communicate with other cells. The arrangement of the cells can form different patterns: trees (cell-like Paun 2000) or graphs (tissue-like Martín-Vide et al. 2003 or neural-like Ionescu et al. 2006). Inside each cell can be placed multisets of objects (inspired from the molecular species), and these multisets of objects can be rewritten using evolution rules (inspired from the chemical reactions).

Spiking neural P systems (Ionescu et al. 2006) are inspired by how the human brain works by spiking information, namely describing how spikes are transmitted from one neuron to another through specific connections called synapses (Gerstner and Kistler 2002). A spiking neural P system consists of a set of neurons placed in the nodes of a directed graph; the neurons send signals (spikes) along synapses (edges of the graph). The information stored inside the neurons is given by the number of spikes and their timings. Several variants of spiking neural P systems were proposed by considering additional features: spiking neural P systems with astrocytes (Paun 2007), spiking neural P systems with anti-spikes (Pan and Paun 2009) and spiking neural P systems with weights (Pan et al. 2012), spiking neural P systems with astrocytes producing calcium (Aman and Ciobanu 2020).

It was proven that spiking neural P systems and their variants: (i) have the same computational power as Turing machines (Leporati et al. 2009), (ii) can solve in a linear or polynomial-time **NP**-complete problems (Ishdorj and Leporati 2008; Leporati and Gutiérrez-Naranjo 2008; Leporati et al. 2009a; Pan et al. 2011; Song et al. 2014; iii) can be connected with automata (Aman and Ciobanu 2015, 2016; Cabarle et al. 2016); iv) can be used in applications (Wang et al. 2015; Zhang et al. 2014, 2017).

Regarding the application of spiking neural P systems to solve intractable problems, a frequent approach is to use

✉ Bogdan Aman
bogdan.aman@iit.academiaromana-is.ro

[1] Institute of Computer Science, Romanian Academy, Iaşi, Romania

[2] Faculty of Computer Science, Alexandru Ioan Cuza University, Iaşi, Romania

families of spiking neural P systems such that each member of this family solves only a finite set of instances of a given size. The solutions of the intractable problems are specified either by the fact that the system halts and the output neuron emits a spike during the computation, or by the existence of output neurons signalling true or false.

The first step towards such a specification was taken already in Leporati et al. (2009a) for the case of non-deterministic (and non-confluent) SN P systems.

In this paper we study the efficiency of spiking neural P systems with astrocytes producing calcium. We show that these systems on which we impose additional restriction, namely we do not consider forgetting rules nor delay in the evolution rules, are powerful enough to provide polynomial-time solutions to the subset sum problem. We consider two ways of constructing the system: (i) semi-uniform: we construct a spiking neural P system with astrocytes producing calcium for each instance of the subset problem and embed the parameters into the constructed systems as number of spikes, graph of neurons and astrocytes, and evolution rules; (ii) uniform: we construct a spiking neural P system with astrocytes producing calcium for all instances of the same size of the subset problem and provide the parameters as resources provided to the input places. In both situations if the subset sum problem has a solution, the output neuron outputs a resource; otherwise, the output neuron does not output any resource.

The rest of the paper is structured as follows. In Sect. 2 we present the spiking neural P systems with astrocytes producing calcium, and in order to illustrate how these systems work we provide an example of a spiking neural P system with astrocytes producing calcium that generates all even numbers without using either forgetting rules or delay in the evolution rules. In Sect. 3, we provide semi-uniform and uniform constructions of spiking neural P system with astrocytes producing calcium working in a non-deterministic way that are able to provide solutions to the subset sum problem in a polynomial time without using either forgetting rules or delay in the evolution rules. Conclusion and references end the paper.

2 Spiking neural P systems with astrocytes producing calcium

Before presenting the spiking neural P systems with astrocytes producing calcium (Aman and Ciobanu 2020), we give several definitions and notations needed in what follows. The set of non-negative integers is denoted by $\mathbb{N}$. The free monoid obtained from a given finite alphabet $V = \{a_1, \ldots, a_n\}$ and the operation of concatenation operation is denoted by V^*; its elements are called strings, while λ denotes the empty string. $V^+ = V^* \setminus \{\lambda\}$ denotes the set of all non-empty strings over V. If $V = \{a\}$, then we use a^* and a^+ to stand for $\{a\}^*$ and $\{a\}^+$, respectively.

A regular expression E over an alphabet V is recursively defined as: (i) $E = \lambda$ is a regular expression and if $a \in V$ then $E = a$ is also a regular expression; (ii) if the expressions E_1 and E_2 are regular, then also the expressions $E = (E_1)(E_2)$, $E = (E_1) \cup (E_2)$ and $E = (E_1)^+$ are regular. For each regular expression E we provide a language $L(E)$ recursively defined as:

$$L(E) = \begin{cases} \{E\} & \text{if } E = \lambda \text{ or } E = a; \\ L(E_1)L(E_2) & \text{if } E = (E_1)(E_2); \\ L(E_1) \cup L(E_2) & \text{if } E = (E_1) \cup (E_2); \\ L((E_1)^+) & \text{if } E = (E_1)^+. \end{cases}$$

Some parentheses can be omitted when writing a regular expression, $(E_1)(E_2)$ denotes concatenation, and also $E^* = (E)^+ \cup \{\lambda\}$. More details about formal languages can be found in Rozenberg and Salomaa (1997).

In Definition 1 we present the spiking neural P systems with astrocytes producing calcium introduced in Aman and Ciobanu (2020), and used in this paper to provide solutions to the subset sum problem in a polynomial time without using either forgetting rules or delay in the evolution rules.

Definition 1 A *spiking neural P system with astrocytes producing calcium* of degree $m + n \geq 1$ is a construct $\Pi = (O, \sigma_1, \ldots, \sigma_m, \tau_1, \ldots, \tau_n, syn_a, syn_c, in, out)$, where:

- $O = \{a, c\}$ is an alphabet (a is called spike, and c is called calcium unit);
- $\sigma_1, \ldots, \sigma_m$ are neurons of the form $\sigma_i = (a_i, c_i, R_i)$, for $1 \leq i \leq m$, where: (a) $a_i, c_i \geq 0$ represent the initial number of spikes and calcium units contained in neuron σ_i; (b) R_i represents a finite set containing rules of the form: (1) $E_a E_c / a^e c^{e'} \rightarrow a^p c^{p'}; d$, where E_a and E_c are regular expressions over a and c, respectively, and $e + e' \geq 1$, $p + p' \geq 1$, $d \geq 0$, with the restrictions $e \geq p$ and $e' \geq p'$; (1′) $a^e c^{e'} \rightarrow \lambda$, where $e, e' \geq 1$, with the restriction that $a^e c^{e'} \notin L(E_a E_c)$ for any rule of type (1) from R_i;
- $\tau, \ldots, \tau_n$ are astrocytes of the form $\tau_i = (c_i, A_i)$, for $1 \leq i \leq n$, where: (a) $c_i \geq 0$ represents the initial number of calcium units contained in astrocyte τ_i; (b) A_i represents a finite set containing rules of the form: (2) $E_c / c^e \rightarrow c^p; d$, where E_c is a regular expression over c, and $e \geq 1$, $p \geq 0$, $d \geq 0$, with the restriction $e \geq p$; (2′) $c^e \rightarrow \lambda$, where $e \geq 1$, with the restriction that $c^e \notin L(E_c)$ for any rule of type (2) from A_i;
- $syn_a \subseteq \{(\sigma_i, \sigma_j) \mid 1 \leq i \neq j \leq m\}$ indicates the synapses used to send spikes from neurons to neurons;

- $syn_c \subseteq \{(\tau_i, \tau_j) \mid 1 \leq i \neq j \leq n\} \cup \{(\sigma_j, \tau_i) \mid 1 \leq i \leq n,$- indicates the synapses used to send calcium units from astrocytes to neurons and astrocytes, and from neurons to astrocytes;
- $in, out \in \{1, 2, \ldots, m\}$ indicate the input and output neurons.

Note that unlike in spiking neural P systems (Ionescu et al. 2006) where only synapses from the set syn_a are needed, in spiking neural P system with astrocytes producing calcium we use two types of synapses, defined by the relations syn_a and syn_c, for communicating either spikes or calcium units; thus these synapses work as filters on the communicated resources. Also, the synapses from syn_a, those communicating spikes, can connect only neurons, while the synapses from syn_c, those communicating calcium units, can connect neurons to astrocytes and also can connect astrocytes to neurons and other astrocytes.

Next, we detail how the rules of the spiking neural P system with astrocytes producing calcium work.

A spiking rule of the form $E_a E_c / a^e c^{e'} \to a^p c^{p'}; d$ is applicable in neuron σ_i containing k spikes and k' calcium units if $k \geq e$, $k' \geq e'$ and $a^k c^{k'} \in L(E_a E_c)$. Applying the rule $E_a E_c / a^e c^{e'} \to a^p c^{p'}; d$ leads to the removal of e spikes and e' calcium units from neuron σ_i, and the creation after d time units of p spikes and p' calcium units to be sent on the synapses connected to the neuron σ_i. Note that if there are no synapses on which some of the created resources can be sent, then those resources are lost. If a rule has the form $a^e c^{e'} / a^e c^{e'} \to a^p c^{p'}; d$ is simply written as $a^e c^{e'} \to a^p c^{p'}; d$.

A forgetting rule of the form $a^e c^{e'} \to \lambda$ is applicable in neuron σ_i containing only e spikes and e' calcium units. Applying the rule $a^e c^{e'} \to \lambda$ leads to the removal of e spikes and e' calcium units from neuron σ_i without any delay.

A calcium rule of the form $E_c / c^e \to c^p; d$ is applicable in astrocyte τ_i containing k calcium units if $k \geq e$ and $c^k \in L(E_c)$. Applying the rule $E_c / c^e \to c^p; d$ leads to the removal of e calcium units from astrocyte τ_i, and the creation after d time units of p calcium units to be sent on the synapses connected to astrocyte τ_i. If a rule has the form $c^e / c^e \to c^p; d$ is simply written as $c^e \to c^p; d$.

A forgetting rule of the form $c^e \to \lambda$ is applicable in astrocyte τ_i containing only e calcium units. Applying the rule $c^e \to \lambda$ leads to the removal of e calcium units from astrocyte τ_i without any delay.

The evolution of the spiking neural P systems with astrocytes producing calcium implies the existence of a global clock. If the delay d appearing in the rules described above is 0, then the production of the spikes and/or calcium units is done instantaneously and these are sent using the existing synapses connected to the neuron or the astrocyte where the rule was executed. On the other hand, if the delay is greater than or equal to 1, a rule applied at step t leads to the closure of the neuron or the astrocyte for the steps $t, \ldots, t + d - 1$; this means that during these steps new spikes and/or calcium units cannot be sent nor received by the astrocyte or neuron where the applied rule resides. Due to the fact that a closed neuron or astrocyte cannot exchange resources, note that all spikes and/or calcium units being communicated towards closed neurons and astrocytes are lost. In the step $t + d$, after firing the created spikes and/or calcium units along the available synapses in syn_a and syn_c, respectively, the neuron or astrocyte becomes open again and is available for receiving resources starting from the step $t + d + 1$.

Using the application of rules as previously described, the number of resources (spikes and calcium units) of neuron σ_i at time $t + 1$ can be deduced using its number of resources at time t:

$$a_i(t+1) = \begin{cases} a_i(t) - r_a + n_a & \text{if a rule r of } \mathrm{R_i} \text{ is used;} \\ a_i(t) + n_a & \text{if no rule of } \mathrm{R_i} \text{ is used,} \end{cases}$$

$$c_i(t+1) = \begin{cases} c_i(t) - r_c + n_c & \text{if a rule r of } \mathrm{R_i} \text{ is used;} \\ c_i(t) + n_c & \text{if no rule of } \mathrm{R_i} \text{ is used,} \end{cases}$$

where r_a and r_c are the number of spikes and calcium units consumed by the used rule r, while n_a and n_c are the number of spikes and calcium units received on the synapses from syn_a and syn_c, respectively, connected to neuron σ_i. The number of calcium units of astrocyte τ_i at time $t + 1$ can be deduced using its number of calcium units at time t:

$$c_i(t+1) = \begin{cases} c_i(t) - r'_c + n'_c & \text{if a rule r}' \text{ of } \mathrm{A_i} \text{ is used;} \\ c_i(t) + n'_c & \text{if no rule of } \mathrm{A_i} \text{ is used,} \end{cases}$$

where r'_c is the number of calcium units consumed by the used rule r', while n'_c is the number of calcium units received on the synapses from syn_c connected to astrocyte τ_i.

At every step of the computation if there exists an applicable rule in a neuron or in an astrocyte, then such a rule must be used. If more than one rule is applicable in a neuron or an astrocyte, then exactly one of the rule is selected in a non-deterministic manner and applied. This leads to sequential execution of rules inside each neuron and astrocyte, and parallel execution of rules in the neurons and astrocytes of the system.

For any computation there exists an output spike *train* that is a sequence of digits 0 and 1, in which 1 marks the steps at which the output neuron fires sending resources to the environment, while 0 marks the steps at which the output neuron does not send resources to the environment. Although there exist various ways of extracting the result of a computation in spiking neural P systems (Ionescu et al. 2006), in what follows we consider the standard one that counts the distance between the first two 1 digits appearing in the output spike train. We denote by $N_2(\Pi)$ the set of numbers computed in the standard mode by a spiking neural P system with astrocytes producing calcium Π, while we denote by $N_2SNP^k_{m,n}$ the family of all such sets computed by such a system, where m, n and k represent the maximum number of neurons, astrocytes or rules used in every neuron or astrocyte, respectively.

Example 1 We illustrate how spiking neural P systems with astrocytes producing calcium work by constructing a system Π that generates all even numbers without using either forgetting rules or delay in the evolution rules. The system in its initial state is depicted in Fig. 1.

A spiking neural P system with seven neurons that generates all even natural numbers is given in Ionescu et al. (2006), while our system requires two neurons and one astrocyte for the same purpose. In the initial configuration, all neurons and astrocytes contain resources, namely a spike and a calcium unit are placed in the neuron σ_1, a spike is placed in the neuron σ_2, while two calcium units are placed in the astrocyte τ_1. However in the first step of the computation only the neuron σ_1 (the output neuron) and the astrocyte τ_1 have the necessary resources to fire; as the delay of the applied rules is 0, the resources (spikes and calcium units) are created and communicated to the connected neurons and astrocytes immediately.

This means that neuron σ_1 is able to apply the rule $ac \rightarrow a$ that sends a spike to the environment, thus marking the fact that we have to count till the second spike is sent to the environment to get the result of the computation. In parallel, the astrocyte τ_1 can choose in a non-deterministic manner to execute any of its rules $c^2 \rightarrow c^2$ and $c^2 \rightarrow c$; assume that only rule $c^2 \rightarrow c^2$ is applied in the first $s \geq 0$ steps. This implies that astrocyte τ_1 produces and sends instantly two calcium units to each of the neurons σ_1 and σ_2. In the next step, the astrocyte τ_1 will contain no resources and thus will be unable to fire. On the other hand, the neurons σ_1 and σ_2 have enough spikes and calcium units to fire one of their rules. This means that two calcium units are sent from neuron σ_1 to the astrocyte τ_1 by executing the rule $c^2 \rightarrow c^2$, while the neuron σ_2 removes the calcium units by executing the rule $ac^2/c^c \rightarrow c$. Note that even if a calcium unit is produced by the rule $ac^2/c^c \rightarrow c$ applied inside neuron σ_2, this calcium unit is discarded as there are no outgoing synapses from syn_c connecting the neuron σ_2 with other neurons or astrocytes. Once the astrocyte τ_1 has two calcium units is ready to fire again and to send calcium units to both neurons. After two computational steps we reached the same configuration of the system; these two steps form a loop that can be applied for an arbitrary number of times.

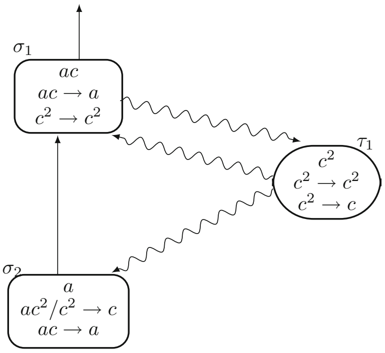

Fig. 1 A spiking neural P system with astrocytes producing calcium generating all even numbers without using either forgetting rules or delay in the evolution rules. We use rectangles and ellipses to depict the neurons and astrocytes, respectively. We also use straight and snake like arrows to depict the synapses from syn_a and syn_c, respectively. The straight arrow without a target leaving neuron σ_1 indicates that this is an output neuron that can send spikes into the environment

The neuron σ_1 can send again a spike into the environment only if the rule $ac \rightarrow a$ is applicable; this implies that the astrocyte τ_1 needs to execute the rule $c^2 \rightarrow c$ at some point during the computation. Assume that the rule $c^2 \rightarrow c$ is applied at step t; notice that $t = 2 \times s + 1$. This implies that immediately, the astrocyte τ_1 sends a calcium unit to each of the neurons σ_1 and σ_2. At step $t + 1$, only the neuron σ_2 can apply the rule $ac \rightarrow a$ sending a spike to neuron σ_1. Thus at step $t + 2$ only the neuron σ_1 contains one spike and one calcium unit, and can apply the rule $ac \rightarrow a$ to output a second spike. As the second spike is fired into the environment at step $t + 2 = 2 \times s + 1 + 2$, it means that $2 \times s + 2$ is the generated number, where $s \geq 0$. After the second spike is sent to the environment the computation halts as there are no more applicable rules in any neurons or in the astrocyte.

Thus $N_2(\Pi) = \{2n \mid n \geq 1\}$ and $N_2(\Pi) \in N_2SNP^2_{2,1}$.

3 Solving the subset sum problem

Spiking neural P systems with astrocytes producing calcium can be used to solve a decision problem I_X, Θ_X, where I_X is a language over a finite alphabet (its elements are called instances) and Θ_X is a total Boolean function over I_X. We consider two ways of constructing the system that will solve the decision problem: (i) semi-uniform: we construct a spiking neural P system with astrocytes producing calcium for each instance of the subset sum problem and embed the parameters into the constructed systems as number of resources (spikes and calcium units), graph of neurons and astrocytes, and evolution rules; (ii) uniform: we construct a spiking neural P system with astrocytes producing calcium for all instances of the same size of the subset problem and provide the parameters as resources sent to the input places. In both situations if the subset sum problem has a solution, the output neuron sends a resource to the environment; otherwise, the output neuron does not output any resource to the environment. Usually the uniform solutions are preferred to the semi-uniform solutions as they relate only to the structure of a given decision problem.

The NP-complete subset sum problem (Garey and Johnson 1979) is as follows: given a finite (multi)set of positive integer numbers, $V = \{v_1, \ldots, v_n\}$ and a positive integer number S, determine whether or not there exists a non-empty (multi)set B, where $B \subseteq V$, such that $\sum_{b \in B} b = S$.

3.1 Semi-uniform solution to the subset sum problem

If we allow rules from a neuron or an astrocyte to be chosen in a non-deterministic manner, then the spiking neural P systems with astrocytes producing calcium can solve any given instance of the subset sum problem in a polynomial number of steps. Depending on how we encode the instance of the problem in spiking neural P systems with astrocytes producing calcium, there exist several possible solutions. We emphasize the fact that all these solutions occur without using either forgetting rules or delay in the evolution rules.

If we allow the instance of the problem to be encoded in the system in the rules and in the initial number of resources (spikes and calcium units), then the spiking neural P systems with astrocytes producing calcium depicted in Fig. 2 solves any given instance of the subset sum problem in two computational steps.

The non-deterministic manner of applying the rules in the system of Fig. 2 results from using the rules $a^{v_i}c \rightarrow a^{v_i}$ and $a^{v_i}c \rightarrow c$, inside the neurons σ_i, for $1 \leq i \leq n$. If the rule $a^{v_i}c \rightarrow a^{v_i}$ is chosen in neuron σ_i, then neuron σ_{out} receives immediately v_i spikes meaning that the number v_i was chosen to be included in the sum. On the other hand, if the rule $a^{v_i}c \rightarrow c$ is chosen in neuron σ_i, then the calcium unit is lost and neuron σ_{out} does not receive any spikes meaning that the number v_i was not chosen to be included in the sum. The output neuron σ_{out} sends in step two of the computation a spike to the environment only if the number of spikes residing in it equals S. Note that the system consists of $n+1$ neurons, and initially contains $n + \sum_i v_i$ spikes and calcium units. The construction represents an improvement with respect to the one presented in Leporati et al. (2009a, 2009) as we do not use forgetting rules nor delays in the evolution rules.

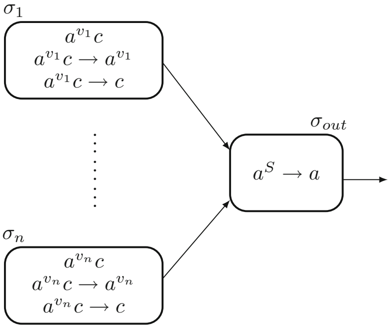

Fig. 2 A semi-uniform spiking neural P system with astrocytes producing calcium solving the subset sum problem without using either forgetting rules or delay in the evolution rules. The instance of the problem is encoded in the system in the rules and in number of the initial resources (spikes and calcium units)

If we allow the instance of the problem to be encoded in the system in the rules and in the number of neurons and astrocytes, then the spiking neural P systems with astrocytes producing calcium depicted in Fig. 3 solves any given instance of the subset sum problem in three computational steps.

The non-deterministic manner of applying the rules results from using the rules $ac \rightarrow a$ and $ac \rightarrow c$ inside the neurons σ_i, for $1 \leq i \leq n$. If the rule $ac \rightarrow c$ is chosen in

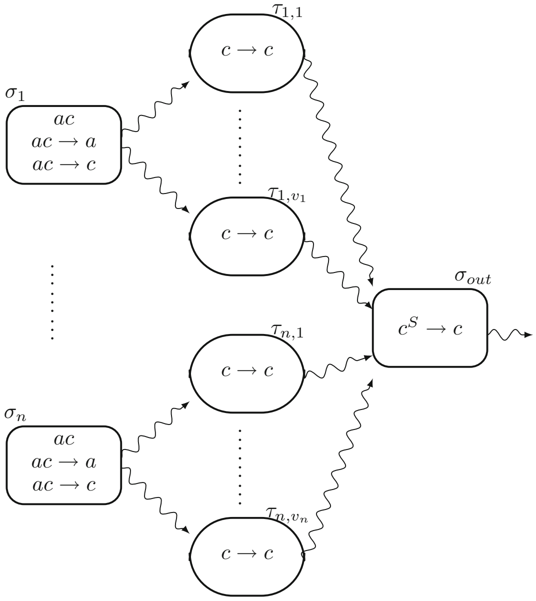

Fig. 3 A semi-uniform spiking neural P system with astrocytes producing calcium solving the subset sum problem without using either forgetting rules or delay in the evolution rules. The instance of the problem is encoded in the system in the rules and in the number of used places (neurons and astrocytes)

neuron σ_i, then each of the astrocytes $\tau_{i,1},\ldots,\tau_{i,v_i}$ receive immediately a calcium unit meaning that the number v_i was chosen to be included in the sum. On the other hand, if the rule $ac \rightarrow a$ is chosen in neuron σ_i, then the spike is lost and astrocytes $\tau_{i,1},\ldots,\tau_{i,v_i}$ do not receive any calcium units meaning that the number v_i was not chosen to be included in the sum. In the next step the neuron σ_{out} receives immediately calcium units from all the astrocytes that had one calcium unit. The output neuron σ_{out} sends in step three of the computation a calcium unit to the environment only if the number of calcium units residing in it equals S. Note that the system consists of $n+1$ neurons and $\sum_i v_i$ astrocytes, and initially contains $2 \times n$ resources (spikes and calcium units in equal amounts). The construction represents an improvement with respect to the one presented in Leporati et al. (2009a) as we do not use forgetting rules nor delays in the evolution rules, and also the number of used places (neurons and astrocytes) is smaller.

If we allow the instance of the problem to be encoded in the system in the initial number of resources (spikes and calcium units) and in the number of neurons and astrocytes, then the spiking neural P systems with astrocytes producing calcium depicted in Fig. 4 solves any given instance of the subset sum problem in a polynomial number of computational steps.

The non-deterministic manner of applying the rules results from using the rules $ac \rightarrow a$ and $ac \rightarrow c$ inside the neurons σ'_i, for $1 \le i \le n$. If the rule $ac \rightarrow c$ is chosen in neuron σ'_i, then each of the astrocytes $\tau_{i,1},\ldots,\tau_{i,v_i}$ receive immediately a calcium unit meaning that the number v_i was chosen to be included in the sum. On the other hand, if the rule $ac \rightarrow a$ is chosen in neuron σ'_i, then the spike is lost and astrocytes $\tau_{i,1},\ldots,\tau_{i,v_i}$ do not receive any calcium units meaning that the number v_i was not chosen to be included in the sum. In the next step the neurons σ_3 and σ_4 receive immediately calcium units from all the astrocytes that had one calcium unit; let us denote this value by S_B. In parallel in the first two steps in neuron σ_3 the rule $a^+c^+/ac \rightarrow c$ is applied twice reducing the multiset of σ_3 from $a^{S+3}c^3$ to $a^{S+1}c$ and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3; also in neuron σ_4 the rule $a^+c^+/ac \rightarrow a$ is applied twice reducing the multiset of σ_4 from $a^{S+2}c^2$ to a^S and the produced spikes are lost as there are no synapses from syn_a leaving from σ_4. At the end of the first two steps, the neuron σ_3 contains the multiset $a^{S+1}c^{S_B+1}$, while the neuron σ_4 contains the multiset $a^Sc^{S_B}$. Depending on the values of S_B and S there exist three possible evolutions:

- $S_B < S$. In this case in the next S_B steps in neuron σ_3 the rule $a^+c^+/ac \rightarrow c$ is applied reducing the multiset of σ_3 from $a^{S+1}c^{S_B+1}$ to $a^{S-S_B+1}c$ and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3; also in neuron σ_4 the rule $a^+c^+/ac \rightarrow a$ is

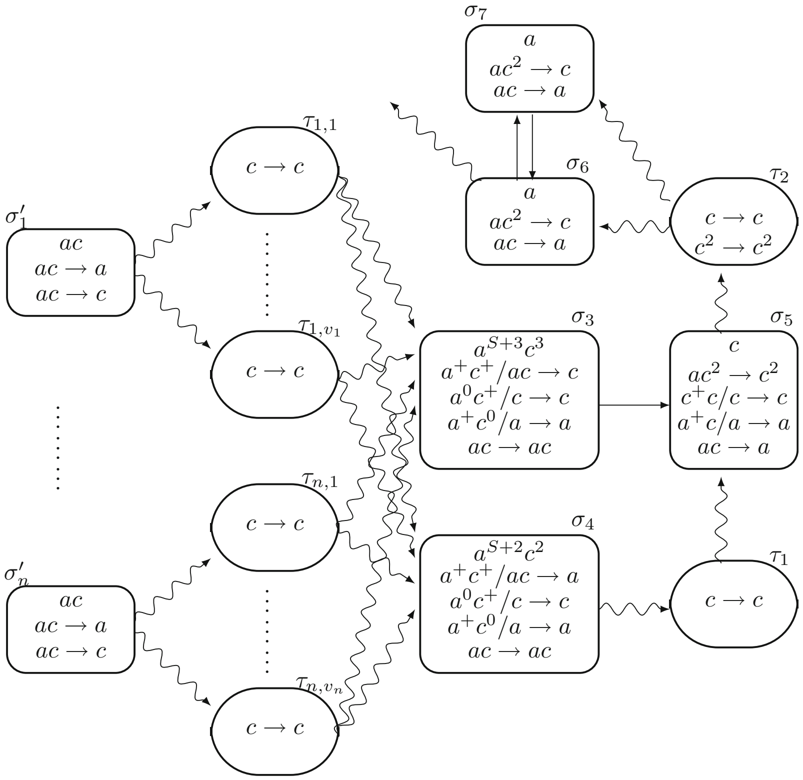

Fig. 4 A semi-uniform spiking neural P system with astrocytes producing calcium solving the subset sum problem without using either forgetting rules or delay in the evolution rules. The instance of the problem is encoded in the system in the initial number of resources (spikes and calcium units) and in the number of used places (neurons and astrocytes)

applied reducing the multiset of σ_4 from $a^S c^{S_B}$ to a^{S-S_B} and the produced spikes are lost as there are no synapses from syn_a leaving from σ_4. In the next $S - S_B + 1$ steps in neuron σ_3 the rule $a^+c^+/ac \rightarrow c$ is applied once and the produced calcium unit is lost as there are no synapses from syn_c leaving from σ_3, while the rule $a^+c^0/a \rightarrow a$ is applied $S - S_B$ times sending $S - S_B$ spikes to neuron σ_5. Also in parallel in the next $S - S_B$ steps in neuron σ_4 the rule $a^+c^0/a \rightarrow a$ is applied and all $S - S_B$ spikes are lost as there are no synapses from syn_a leaving from σ_4. Starting from step $S + 4$ in neuron σ_5 the rule $a^+c/a \rightarrow a$ is applied for $S - S_B$ steps and the produced spikes are lost as there are no synapses from syn_a leaving from σ_5. Thus no calcium unit is emitted into the environment and the computation stops after $2 \times S - S_B + 3$ steps.

- $S_B = S$. In this case in the next $S - 1$ steps in neuron σ_3 the rule $a^+c^+/ac \rightarrow c$ is applied reducing the multiset of σ_3 from $a^{S+1}c^{S_B+1}$ to a^2c^2 and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3; also in neuron σ_4 the rule $a^+c^+/ac \rightarrow a$ is applied $S_B - 1$ times reducing the multiset of σ_4 from $a^S c^{S_B}$ to ac and the produced spikes are lost as there are no synapses from syn_a leaving from σ_4. In the next step in neuron σ_3 the rule $a^+c^+/ac \rightarrow c$ is applied once reducing the multiset of σ_3 from a^2c^2 to ac and the produced calcium unit is lost as there are no synapses from syn_c leaving from σ_3. In parallel in neuron σ_4 the rule $ac \rightarrow ac$ is applied sending a calcium unit to astrocyte τ_1, while the spike is lost as there are no synapses from syn_a leaving from σ_4. In the next step in neuron σ_3 the rule $ac \rightarrow ac$ sends a spike to neuron σ_5, while the calcium unit is lost as there are no synapses from syn_c leaving from σ_3. In parallel the astrocyte τ_1 applies the rule sending the calcium unit to the neuron σ_5. Thus after $S + 3$ steps the neuron σ_5 contains a spike and a calcium unit being able to apply the rule $c^2 \rightarrow c^2$ sending two calcium units to astrocyte τ_2, that will send in next step to both neurons σ_6 and σ_7 by applying the rule $c^2 \rightarrow c^2$. Next neuron σ_7 applies rule $ac^2 \rightarrow c$ removing all its resources as there are no synapses from syn_c leaving from σ_7, while in parallel the neuron σ_6 applies the rule $ac^2 \rightarrow c$; thus a calcium unit is emitted into the environment and the computation stops after $S + 6$ steps.
- $S_B > S$. In this case in the next S steps in neuron σ_3 the rule $a^+c^+/ac \rightarrow c$ is applied reducing the multiset of σ_3 from $a^{S+1}c^{S_B+1}$ to ac^{S_B-S+1} and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3; also in neuron σ_4 the rule $a^+c^+/ac \rightarrow a$ is applied reducing the multiset of σ_4 from $a^S c^{S_B}$ to c^{S_B-S} and the produced spikes are lost as there are no synapses from syn_a leaving from σ_4. In the next $S_B - S + 1$ steps in neuron σ_3 the rule $a^+c^+/ac \rightarrow c$ is applied once and afterwards the rule $a^0c^+/c \rightarrow c$ is applied $S_B - S$ times and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3. Also in parallel in the next $S_B - S$ steps in neuron σ_4 the rule $a^0c^+/c \rightarrow c$ is applied sending $S_B - S$ calcium units to astrocyte τ_1. Each calcium unit reaching astrocyte τ_1 will be eliminated after four steps: (i) in astrocyte τ_1 the rule $c \rightarrow c$ is applied sending the calcium unit to neuron σ_5; (ii) in neuron σ_5 the rule $c^+c/c \rightarrow c$ is applied sending the calcium unit to astrocyte τ_2; (iii) in astrocyte τ_2 the rule $c \rightarrow c$ is applied sending a calcium unit to both neurons σ_6 and σ_7; (iv) in neurons σ_6 and σ_7 the rule $ac \rightarrow a$ is applied removing the calcium unit. Thus no calcium unit is emitted into the environment and the computation stops after $S_B + 8$ steps.

Note that the system consists of $n + 5$ neurons and $2 + \sum_i v_i$ astrocytes, initially contains $2 \times n + 2 \times S + 13$ resources (spikes and calcium units), and we do not use forgetting rules nor delays in the evolution rules.

3.2 Uniform solution to the subset sum problem

Constructing a spiking neural P system with astrocytes producing calcium which solves the subset sum problem in a uniform way implies that only the number n of variables is known by the system, while the values from V and the number S should be available only as resources in some input places (neurons and astrocytes) when the computation starts. We consider that the instance of the problem is provided to the system using $n + 2$ input neurons; we introduce v_i, $1 \leq i \leq n$, spikes in n input neurons, $2 \times S + 5$ spikes and $S + 5$ calcium units in an input neuron and $2 \times S + 4$ spikes and $S + 4$ calcium units in another input neuron.

The spiking neural P systems with astrocytes producing calcium depicted in Fig. 5 solves any given instance of the subset sum problem in a polynomial number of computational steps.

The non-deterministic manner of applying the rules results from using the rules $ac \rightarrow a$ and $ac \rightarrow c$ inside the neurons σ_i'', for $1 \leq i \leq n$. If the rule $ac \rightarrow a$ is chosen in neuron σ_i'', then the spike is lost as there are no synapses from syn_a leaving from neuron σ_i''. On the other hand, if the rule $ac \rightarrow c$ is chosen in neuron σ_i'', then the astrocyte τ_i'' receives immediately a calcium unit meaning that the number v_i was chosen to be included in the sum. In the next step the astrocyte τ_i'' sends a calcium unit to both the neuron σ_i' and the astrocyte τ_i'''. This means that in the next v_i steps the rule $c \rightarrow c$ from the astrocyte τ_1''' and the rule

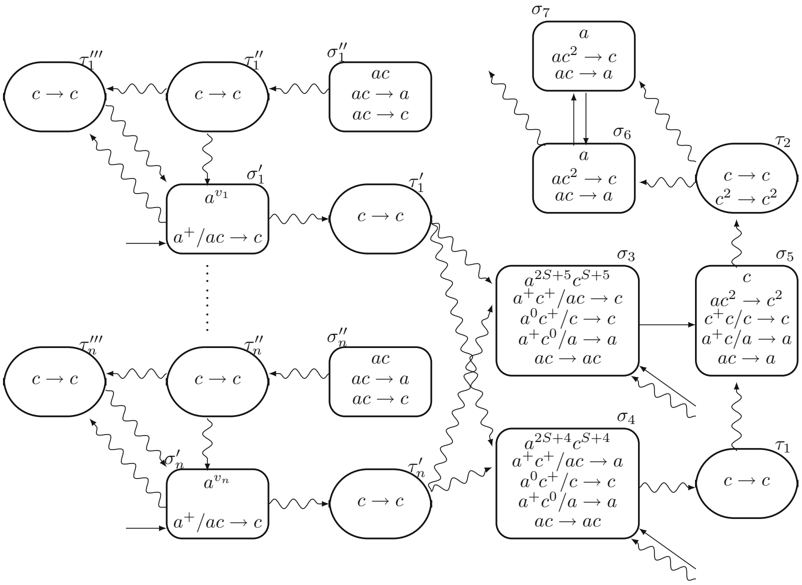

Fig. 5 An uniform spiking neural P system with astrocytes producing calcium solving the subset sum problem without using either forgetting rules or delay in the evolution rules. The instance of the problem is encoded in the system in the initial number of resources (spikes and calcium units)

$a^+/ac \to c$ from the neuron σ_i' are used to send the input placed inside the neuron σ_i' towards the astrocyte τ_i'. In the next v_i steps the astrocyte τ_i' will use the rule $c \to c$ to send a calcium unit to the neurons σ_3 and σ_4. By assuming that for all $v_i \leq S$, for $1 \leq i \leq n$, we expect that all input spikes from the neurons σ_i'', for $1 \leq i \leq n$, will reach the neurons σ_3 and σ_4 in $S+4$ steps. In what follows we denote by S_B the amount of calcium units that reached the neurons σ_3 and σ_4 in the first $S+4$ steps of the computation.

In parallel in the first $S+4$ steps in neuron σ_3 the rule $a^+c^+/ac \to c$ is applied reducing the multiset of σ_3 from $a^{2S+5}c^{S+5}$ to $a^{S+1}c$ and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3; also in neuron σ_4 the rule $a^+c^+/ac \to a$ is applied reducing the multiset of σ_4 from $a^{2S+4}c^{S+4}$ to a^S and the produced spikes are lost as there are no synapses from syn_a leaving from σ_4. At the end of the first $S+4$ steps, the neuron σ_3 contains the multiset $a^{S+1}c^{S_B+1}$, while the neuron σ_4 contains the multiset $a^Sc^{S_B}$. Depending on the values of S_B and S there exist three possible evolutions:

- $S_B < S$. In this case in the next S_B steps in neuron σ_3 the rule $a^+c^+/ac \to c$ is applied reducing the multiset of σ_3 from $a^{S+1}c^{S_B+1}$ to $a^{S-S_B+1}c$ and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3; also in neuron σ_4 the rule $a^+c^+/ac \to a$ is applied reducing the multiset of σ_4 from $a^Sc^{S_B}$ to a^{S-S_B} and the produced spikes are lost as there are no synapses from syn_a leaving from σ_4. In the next $S-S_B+1$ steps in neuron σ_3 the rule $a^+c^+/ac \to c$ is applied once and the produced calcium unit is lost as there are no synapses from syn_c leaving from σ_3, while the rule $a^+c^0/a \to a$ is applied $S-S_B$ times sending $S-S_B$ spikes to neuron σ_5. Also in parallel in the next $S-S_B$ steps in neuron σ_4 the rule $a^+c^0/a \to a$ is applied and all $S-S_B$ spikes are lost as there are no synapses from syn_a leaving from σ_4. Starting from step $S+4$ in neuron σ_5 the rule $a^+c/a \to a$ is applied for $S-S_B$ steps and the produced spikes are lost as there are no synapses from syn_a leaving from σ_5. Thus no calcium unit is emitted into the environment and the computation stops after $3 \times S - S_B + 5$ steps.
- $S_B = S$. In this case in the next $S-1$ steps in neuron σ_3 the rule $a^+c^+/ac \to c$ is applied reducing the multiset of σ_3 from $a^{S+1}c^{S_B+1}$ to a^2c^2 and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3; also in neuron σ_4 the rule $a^+c^+/ac \to a$ is applied $S_B - 1$ times reducing the multiset of σ_4 from $a^Sc^{S_B}$ to ac and the produced spikes are lost as there are no synapses from syn_a leaving from σ_4. In the next step in neuron σ_3 the rule $a^+c^+/ac \to c$ is applied once reducing the multiset of σ_3 from a^2c^2 to ac and the produced calcium unit is lost as there are no synapses from syn_c leaving from σ_3. In parallel in neuron σ_4 the rule $ac \to ac$ is applied sending a calcium unit to astrocyte τ_1, while the spike is lost as there are no synapses from syn_a leaving from σ_4. In the next step in neuron σ_3 the rule $ac \to ac$ sends a spike to neuron σ_5, while the calcium unit is lost as there are no synapses from syn_c leaving from σ_3. In parallel the astrocyte τ_1 applies the rule sending the calcium to the neuron σ_5. Thus after $S+3$ steps the neuron σ_5 contains a spike and a calcium unit being able to apply the rule $c^2 \to c^2$ sending two calcium units to astrocyte τ_2, that will send in next step to both neurons σ_6 and σ_7 by applying the rule $c^2 \to c^2$. Next neuron σ_7 applies rule $ac^2 \to c$

removing all its resources as there are no synapses from syn_c leaving from σ_7, while in parallel the neuron σ_6 applies the rule $ac^2 \rightarrow c$; thus a calcium unit is emitted into the environment and the computation stops after $2 \times S + 8$ steps.

- $S_B > S$. In this case in the next S steps in neuron σ_3 the rule $a^+c^+/ac \rightarrow c$ is applied reducing the multiset of σ_3 from $a^{S+1}c^{S_B+1}$ to ac^{S_B-S+1} and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3; also in neuron σ_4 the rule $a^+c^+/ac \rightarrow a$ is applied reducing the multiset of σ_4 from $a^Sc^{S_B}$ to c^{S_B-S} and the produced spikes are lost as there are no synapses from syn_a leaving from σ_4. In the next $S_B - S + 1$ steps in neuron σ_3 the rule $a^+c^+/ac \rightarrow c$ is applied once and afterwards the rule $a^0c^+/c \rightarrow c$ is applied $S_B - S$ times and the produced calcium units are lost as there are no synapses from syn_c leaving from σ_3. Also in parallel in the next $S_B - S$ steps in neuron σ_4 the rule $a^0c^+/c \rightarrow c$ is applied sending $S_B - S$ calcium units to astrocyte τ_1. Each calcium unit reaching astrocyte τ_1 will be eliminated after four steps: (i) in astrocyte τ_1 the rule $c \rightarrow c$ is applied sending the calcium unit to neuron σ_5; (ii) in neuron σ_5 the rule $c^+c/c \rightarrow c$ is applied sending the calcium unit to astrocyte τ_2; (iii) in astrocyte τ_2 the rule $c \rightarrow c$ is applied sending a calcium unit to both neurons σ_6 and σ_7; (iv) in neurons σ_6 and σ_7 the rule $ac \rightarrow a$ is applied removing the calcium unit. Thus no calcium unit is emitted into the environment and the computation stops after $S_B + S + 10$ steps.

Note that the system consists of $2n + 5$ neurons and $3n + 2$ astrocytes, initially contains $2 \times n + 2 \times S + 13$ resources (spikes and calcium units), and the length of the computation depends on the values of S and S_B. The construction represents an improvement with respect to the one presented in Leporati et al. (2009a) as we do not use forgetting rules nor delays in the evolution rules, and also the number of used places (neurons and astrocytes) is smaller.

4 Conclusion

This paper represents a first step towards using spiking neural P systems with astrocytes producing calcium to solve, in a non-deterministic manner, computationally hard problems. We showed that non-deterministic spiking neural P systems with astrocytes producing calcium that do not consider forgetting rules nor delay in the evolution rules, are powerful enough to provide polynomial-time solutions to the subset sum problem. We provided four ways of constructing such a system: (i) semi-uniform: we constructed a spiking neural P system with astrocytes producing calcium for each instance of the subset problem and embedded the parameters into the constructed systems: in the rules and number of initial resources (Fig. 2), in the rules and in the number of used neurons and astrocytes (Fig. 3), and as number of resources (spikes and calcium units) and in the number of used neurons and astrocytes (Fig. 4) ; (ii) uniform: we constructed a spiking neural P system with astrocytes producing calcium for all instances of the same size of the subset problem and provided the parameters as number of spikes and calcium units (Fig. 5).

In all approaches if the subset sum problem has a solution, the output neuron outputs a resource to the environment; otherwise, the output neuron does not output any resource to the environment.

References

Aman, B., Ciobanu, G. (2015). Automated verification of stochastic spiking neural P systems. In Rozenberg, G., Salomaa, A., Sempere, J. M., and Zandron, C., editors, *16th International conference on membrane computing, CMC 2015, Revised Selected Papers*, volume 9504 of *Lecture Notes in Computer Science*, pages 77–91. Springer. https://doi.org/10.1007/978-3-319-28475-0_6

Aman B, Ciobanu G (2016) Modelling and verification of weighted spiking neural systems. Theoret Comput Sci. https://doi.org/10.1016/j.tcs.2015.11.005

Aman B, Ciobanu G (2020) Spiking neural P systems with astrocytes producing calcium. Int J Neur Sys. https://doi.org/10.1142/S0129065720500665

Cabarle FGC, Adorna HN, Pérez-Jiménez MJ (2016) Notes on spiking neural P systems and finite automata. Natural Comput 15(4):533–539

Garey MR, Johnson DS (1979) Computers and Intractability: a guide to the theory of NP-completeness. Freeman, W. H

Gerstner W, Kistler WM (2002) Spiking neuron models: single neurons, populations, plasticity. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511815706

Ionescu M, Paun G, Yokomori T (2006) Spiking neural P systems. Fundamenta Informaticae, 71(23):279–308

Ishdorj T, Leporati A (2008) Uniform solutions to SAT and 3-SAT by spiking neural P systems with pre-computed resources. Natural Comput 7(4):519–534

Leporati A, Gutiérrez-Naranjo MA (2008) Solving Subset Sum by spiking neural P systems with pre-computed resources. Fundamenta Informaticae 87(1):61–77

Leporati A, Mauri G, Zandron C, Paun G, Pérez-Jiménez MJ (2009) Uniform solutions to SAT and Subset Sum by spiking neural P systems. Natur Comput 8(4):681–702. https://doi.org/10.1007/s11047-008-9091-y

Leporati A, Zandron C, Ferretti C, Mauri G (2009) On the computational power of spiking neural P systems. Int J Unconvent Comput 5(5):459–473

Martín-Vide, C., Paun, G., Pazos, J., and Rodríguez-Patón, A. (2003). Tissue P systems. *Theoretical Computer Science*, 296(2):295–326. https://doi.org/10.1016/S0304-3975(02)00659-X

Pan L, Paun G (2009) Spiking neural P systems with anti-spikes. Int J Comput Commun & Cont 4(3):273–282

Pan L, Paun G, Pérez-Jiménez MJ (2011) Spiking neural P systems with neuron division and budding. Science China Info Scien 54(8):1596–1607

Pan L, Zeng X, Zhang X, Jiang Y (2012) Spiking neural P systems with weighted synapses. Neur Process Lett 35(1):13–27

Paun G (2000) Computing with membranes. J Comput Sys Sci 61(1):108–143

Paun G (2002) Membrane computing: an introduction. Natur Comput Ser. https://doi.org/10.1007/978-3-642-56196-2

Paun G (2007) Spiking neural P systems with astrocyte-like control. J Univer Comput Sci 13(11):1707–1721

Rozenberg, G., Bäck, T., and Kok, J. N., editors (2012). Handbook of Natural Computing. Springer, Berlin https://doi.org/10.1007/978-3-540-92910-9

Rozenberg G, Salomaa A (eds) (1997) Handbook of formal languages. Springer, Berlin

Song T, Luo L, He J, Chen Z, Zhang K (2014) Solving Subset Sum problems by time-free spiking neural P systems. Appl Math Info Sci 8(1):327–332

Wang T, Zhang G, Zhao J, He Z, Wang J, Pérez-Jiménez MJ (2015) Fault diagnosis of electric power systems based on fuzzy reasoning spiking neural P systems. IEEE Trans Power Sys 30(3):1182–1194

Zhang, G., Pérez-Jiménez, M. J., and Gheorghe, M. (2017). Real-life applications with membrane computing. Emergence, Complexity, Computation. Springer, Berlin https://doi.org/10.1007/978-3-319-55989-6

Zhang G, Rong H, Neri F, Pérez-Jiménez MJ (2014) An optimization spiking neural P system for approximately solving combinatorial optimization problems. Int J Neur Sys. https://doi.org/10.1142/S0129065714400061

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.