Hindawi
Complexity
Volume 2018, Article ID 3687839, 15 pages
https://doi.org/10.1155/2018/3687839

Research Article

An Iterative Information-Theoretic Approach to the Detection of Structures in Complex Systems

Marco Villani[1,2] **Laura Sani**[3] **Riccardo Pecori**[3,4] **Michele Amoretti**[3] **Andrea Roli**[5] **Monica Mordonini**[3] **Roberto Serra**[1,2] **and Stefano Cagnoni**[3]

[1]*Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Modena, Italy*
[2]*European Centre for Living Technology, Venezia, Italy*
[3]*Department of Engineering and Architecture, University of Parma, Parma, Italy*
[4]*SMARTEST Research Centre, eCAMPUS University, Novedrate, Italy*
[5]*Department of Computer Science and Engineering (DISI), University of Bologna, Bologna, Italy*

Correspondence should be addressed to Stefano Cagnoni; stefano.cagnoni@unipr.it

Received 21 March 2018; Revised 20 July 2018; Accepted 30 July 2018; Published 11 November 2018

Academic Editor: Diego R. Amancio

Systems that exhibit complex behaviours often contain inherent dynamical structures which evolve over time in a coordinated way. In this paper, we present a methodology based on the Relevance Index method aimed at revealing the dynamical structures hidden in complex systems. The method iterates two basic steps: detection of relevant variable sets based on the computation of the Relevance Index, and application of a sieving algorithm, which refines the results. This approach is able to highlight the organization of a complex system into sets of variables, which interact with one another at different hierarchical levels, detected, in turn, in the different iterations of the sieve. The method can be applied directly to systems composed of a small number of variables, whereas it requires the help of a custom metaheuristic in case of systems with larger dimensions. We have evaluated the potential of the method by applying it to three case studies: synthetic data generated by a nonlinear stochastic dynamical system, a small-sized and well-known system modelling a catalytic reaction, and a larger one, which describes the interactions within a social community, that requires the use of the metaheuristic. The experiments we made to validate the method produced interesting results, effectively uncovering hidden details of the systems to which it was applied.

1. Introduction

Systems that exhibit complex behaviours are generally made up of elementary constituents interacting in a nonlinear way. These constituents could be organized in a hierarchical structure [1, 2] as, for example, in biological systems (composed of cells that form tissues, which compose organs that compose organisms). However, very often complex systems are characterised by relations among components at different levels, so that the structure of their interactions cannot be represented by a clear tree-like topology but rather by entangled hierarchies [3].

The identification of such structures involves the detection of the parts composing the system and their subcomponents (up to a predefined level of granularity) and, possibly, the interactions among these components. Frequently, these structures need to be inferred by observing a system's dynamics, either in its unperturbed condition or after perturbations.

In previous works [4, 5], we have introduced the Relevance Index (RI), a measure based on information theory that seems suitable for exploring the organization of complex systems. The RI makes it possible to identify, as components of a system, relevant sets of variables that show an integrated behaviour and interact more weakly with the rest of the system. The RI method has been applied with interesting results to several systems: some of them had been artificially designed in order to test the effectiveness of the technique, while others referred to interesting physical, chemical, biological, or socio-economic systems [6, 7]. In addition,

the efficiency of the method has also been improved by using a parallel implementation of the RI computation [8] and some metaheuristics to deal with the "curse of dimensionality" when analysing high-dimensional systems [9, 10]. In general, the method can be applied every time a collection of observations of the values of the system variables at different instants or conditions is available. This may be the case in off-line system analysis, but also in on-line analysis, when observations come from a data stream. In general, we suppose that information on the relations among the system's variables is not available; however, this method can be successfully combined with community detection algorithms in complex networks [11, 12].

In many cases, the components identified by means of the RI have an intricate nested structure, which makes it hard to understand which groups of variables are really important. To overcome this problem, a filtering or sieving algorithm has been introduced [13], whose aim is to filter out any proper subset or superset of a reference variable set which has a larger RI value. By iteratively considering variable sets in descending order of index values and running the sieve, one can obtain an ensemble of disjoint or only partially overlapping variable sets, which can be considered the most relevant building blocks of the system. Since this algorithm is based on a well-defined criterion, user interpretation is not required for detecting the most relevant groups of variables after computing the RI for each possible subset. Nevertheless, at this stage, only the lowest-level components of a possible hierarchy can be uncovered. An iterative application of the method is able to construct a hierarchical view of the system without any prior information about its structure, just by analysing data that describe its dynamics.

Somehow, we could say that our method represents a way of giving a purely "objective" description of the structure of a complex system, based only on the observation of its states. In fact, our analysis anticipates any possible interpretation of its results that can be given a posteriori, as it is able to detect intrinsic dependencies among system variables or variable sets. Moreover, whenever the system can be observed in different operational conditions (or from different initial conditions), the bias depending on specific samples may be drastically reduced. However, one should not forget that any data collection implies that data are already biased (i) by the measuring technique/device or (ii) by the goals/aims with which the data have been collected.

This paper, besides summarising our previous work on the Relevance Index from the point of view of both the development of the method and its implementation, reports the first results we obtained by applying the iterative sieve algorithm to synthetic data and to two real-world systems, which we used as a validation test for the full method. Synthetic data have been generated by considering the steady states of a stochastic nonlinear dynamical system composed of networks with Boolean or random update functions. These data are a typical example of nonlinear multiple-variable interactions and, although generated by an abstract system, are representative of a wide class of real systems. Conversely, the first real-world system represents the results of a catalytic reaction, whose dynamics are well known. The last "system" we have analysed shows how our method can be applied to social sciences: its status over time is represented by the log of a series of meetings, held within a rather large project, in which each potential participant has been marked as absent or present.

The paper is structured as follows: Section 2 introduces the theory based on which the Relevance Index is derived and computed, along with the method for the analysis of complex systems, in which it is employed, whose main feature is a sieving algorithm used to let the actually relevant variable sets emerge among the many sets that may have very similar RI values. Subsequently, in Section 3, we describe the metaheuristic we used to deal with systems whose size makes it impossible to assign a RI to all variable subsets, which relies on a GPU-based implementation of the RI computation. The results of the experimental validation of the method are shown in Section 4. Finally, we outline conclusions and future works in Section 5.

2. Methods

In this section, we succinctly introduce the Relevance Index and outline the method we developed to compute it. We first define the index and then present some improvements that make it applicable to a wide class of complex systems. This metric can be computed just by analysing data recordings of the system state at different instants or in different conditions, without requiring any a priori information on the system structure.

2.1. The Relevance Index. The roots of the RI can be traced back to the work by Tononi et al. [14], who introduced the Functional Cluster Index to detect functional clusters in brain regions. The method that relies on the RI extends the Functional Cluster Index, since it can be applied to dynamical systems, while the Functional Cluster Index had been defined for fluctuations around a steady state. The index was first called *dynamical cluster index* and subsequently *dynamical relevance index* to emphasise the possibility of applying it to actual dynamical systems. However, we preferred to drop the "dynamical" characterisation since the index can be applied to more general scenarios, making it possible to discover multiple relations among system variables even when data do not belong to a time series; in fact, they may even represent the state of different systems.

The RI was originally introduced to understand the actual organization of dynamical systems; to this aim, one needs to *(i)* properly identify meaningful organizational *levels* emerging from the interactions of lower-level entities (and possibly also of higher-level entities, such as groups of interacting chemical species in protocells [15]) and *(ii)* describe the interactions between these meso-levels. In some cases, it is possible to describe such a structure by means of a simple tree-like hierarchy, as happens in several physical systems where the levels can be identified with the space-time scales of the phenomena (microscopic and macroscopic or micro-meso-macro), in inclusion hierarchies (e.g., like an organ made of tissues, which comprise cells, etc.), in social

organizations, and so on. However, one frequently encounters cases where the interactions among the high levels are graph-like, and their organization cannot be satisfactorily described by a simpler hierarchical structure.

The purpose of the RI method is to identify subsets of variables that behave in a coordinated way in a dynamical system. This means that the variables that are members of the subset are *integrated* with the other variables of the subset much more tightly than with the external ones. These subsets are possible candidates as higher-level entities for describing the organization of a system; they will be called *relevant subsets* (RSs, omitting the specification that they are initially just candidates). A quantitative measure, well suited for identifying them, is defined as follows (the presentation below follows the one given in [5]).

Let U be the set of discrete variables describing a system whose status changes in time, and let us suppose the time series of their values is available. According to information theory [16, 17], Shannon's entropy of an element x_i is defined as

$$H(x_i) = -\sum_{v \in V_i} p(v) \log p(v), \tag{1}$$

where V_i is the set of possible values of x_i and $p(v)$ the probability of occurrence of symbol v. Since, in this work, we deal with observational data, probabilities will be estimated through relative frequencies.

The entropy of a pair of elements x_i and x_j is defined by means of their joint probabilities:

$$H(x_i, x_j) = -\sum_{v \in V_i} \sum_{w \in V_j} p(v, w) \log p(v, w). \tag{2}$$

Equation (2) can be obviously extended to sets composed of more than two elements.

Let us now consider a subset S of U composed of k elements. Its integration $I(S)$, also known as intrinsic information or multi-information, is defined as

$$I(S) = \sum_{x \in S} H(x) - H(S). \tag{3}$$

$I(S)$ represents the deviation from statistical independence of the k elements in S. The integration alone could be used to try and identify the relevant subsets. However, finding relevant sets can be seen as a two-objective optimization problem: on the one hand, we want to find sets whose variables are strongly correlated with one another while, on the other hand, we also want such sets to be as independent as possible from the rest of the system, a property that can be measured by the *mutual information* between the set under consideration and the rest of the system (the lower the mutual information, the less dependent the sets of variables).

The two objectives can be unified by trying to find the $N_{\max}$ subsets S_j, with $j = 1, \ldots, N_{\max}$, of U that exhibit the highest values of the *Relevance Index*, defined as the ratio between the integration $I(S_j)$ and the mutual information $M(S_j ; U \backslash S_j)$ between S_j and the rest of the system $U \backslash S_j$.

The mutual information between a variable set S_j and $U \backslash S_j$ is defined as

$$\begin{aligned} M(S_j ; U \backslash S_j) &\equiv H(S_j) + H(S_j | U \backslash S_j) \\ &= H(S_j) + H(U \backslash S_j) - H(S_j, U \backslash S_j), \end{aligned} \tag{4}$$

where $H(A \mid B)$ is the conditional entropy, and $H(A, B)$ denotes the joint entropy.

Finally, the *Relevance Index* of a set S_j is defined as

$$\mathrm{RI}(S_j) = \frac{I(S_j)}{M(S_j ; U \backslash S_j)}. \tag{5}$$

Note that, being a ratio, the RI is undefined in all those cases where $M(S_j ; U \backslash S_j)$ vanishes.

In these cases, however, the subset S_j is statistically independent from the rest of the system and, therefore, should be analysed separately. These situations must obviously be screened out in advance.

It is also worth noting that the RI increases with subset size k. In [4], we have introduced a possible way to normalize it; however, a better approach consists in assessing the statistical significance of the RI of S_k, where k, in this case, is the subset size, by means of a statistical significance index. In this paper, we will consider the z-score $T_c(S_k)$ defined as

$$T_c(S_k) = \frac{\nu r(S_k) - \nu \langle r_h \rangle}{\nu \sigma(r_h)} = \frac{r(S_k) - \langle r_h \rangle}{\sigma(r_h)}, \tag{6}$$

where $\langle r_h \rangle$ and $\sigma(r_h)$ are, respectively, the average and the standard deviation of the RI of a sample of subsets of size k extracted from a reference system U_h, randomly generated, which preserves the priors of each single variable in U, and $\nu = \langle MI_h \rangle / \langle I_h \rangle$ is a normalization constant. It is worth noting that the aim of the reference system is to quantify the finite-size effects affecting the information-theoretic measures of a random instance of a system with finite dimensions.

The search for the RSs of a dynamical system by means of this method requires a collection of observations of the variables at different instants. Since the RI (and its statistical significance) depends only on symbol frequencies, in principle, a time series is not strictly required; a collection of snapshots of the system variables is enough to compute it.

Of course, the comparison with the reference system will be more significant, the less noisy and the more such snapshots are. In [5], however, we have verified that the analysis of complex systems based on the RI works correctly also in the presence of noisy data.

A list of relevant sets can be obtained, in principle, by enumerating all possible subsets of U and ranking them according to their RI values (or any other associated statistical significance index, such as the T_c index used in this work). The highest T_c subsets are most likely to play a relevant role in determining the system dynamics.

Input: The array C of all the CRSs, ranked by their T_c value in descending order.
Output: RS, a subset of C
$RS \leftarrow \varnothing$
$n \leftarrow |C|$
Initialize auxiliary array $Del[k] \leftarrow 0$ for k in 1… n
for $i = 1$ to $n - 1$ **do**
  for $j = i + 1$ to n **do**
    if $Del[i] \neq 1 \wedge Del[j] \neq 1$ **then**
      if $C[i] \subset C[j] \vee C[j] \subset C[i]$ **then**
        $Del[j] \leftarrow 1$
      end if
    end if
  end for
end for
for $i = 1$ to n **do**
  if $Del[i] = 0$ **then**
    $RS \leftarrow RS \cup \{C[i]\}$
  end if
end for

Algorithm 1: Sieving algorithm.

For large-sized systems, an exhaustive enumeration is obviously impractical because the number of possible subsets of set U is $2^{|U|}$, where $|U|$ denotes the cardinality of U. When the computation load needed for an exhaustive enumeration exceeds the available computing resources, one needs to resort either to random sampling or to metaheuristic techniques, like the genetic algorithms hybridized with a local search we use in this work [9]. The main general idea of this approach consists in performing a sampling that is biased towards sets of variables characterised by high T_c values. Indeed, a genetic algorithm performs a sampling [18] where parameters are iteratively modified so that subsequent samplings are much denser in those regions where the objective function (the T_c, in this case) is likely to be higher.

Whatever the method used to compute the index, the collection of RSs returned is likely to contain RSs included in others or partially overlapping, which requires further analyses to assess their actual relevance. Indeed, having a high T_c value is not sufficient to characterise a RS because such a value might result from the inclusion of a smaller set characterised by a higher T_c (i.e., the set under consideration is a superset of a more relevant one), in which case the only relevant set would be the latter. Conversely, a set having a high T_c value might reach an even higher value, if some other relevant variables are added to it (i.e., the set under consideration is a subset of a more relevant one), in which case we would consider only the larger set as relevant.

To tackle this problem, in [13], we have proposed a postprocessing sieving algorithm to reduce the overall number of subsets. The main assumption of the procedure is that if set A is a proper subset of B, that is, $A \subset B$, then only the higher T_c subset is taken into consideration.

Therefore, only disjoint or partially overlapping subsets are kept. After this postprocessing procedure, the remaining subsets cannot be decomposed any further, and thus, they represent the building blocks of the dynamical organization of the system. The pseudocode of the sieve is presented in Algorithm 1.

2.2. The Iterative Sieving Method. The method previously described allows one to identify a plausible organization of the system in terms of its lowest-level, possibly overlapping, subsets of variables. Nevertheless, since complex systems have often a hierarchical structure, one may want to be able to make hypotheses on aggregated relations among the RSs thus identified, so as to derive a hierarchy of RSs. To this aim, we devised an iterative version of the sieving method, which acts on the data by iteratively grouping one or more RSs into a single entity. In fact, there are several ways to do so; the simplest one, yet quite effective, consists in iteratively running the sieving algorithm on the same data, each time using a new representation, in which the top-ranked RS of the previous iteration, in terms of T_c values, is considered as atomic and substituted by a single variable (henceforth called a *group variable*). In this way, each run produces a new atomic group of variables composed of both single variables and group variables introduced in previous iterations.

Suppose we have four variables denoted as A, B, C, and D in the system and that the group $(A\,B)$ is the most relevant set detected by the first iteration of the algorithm. Then, the second iteration will analyse the dynamics of a system comprising the three variables $(A\,B)$, C, and D, and so on until the T_c value of the most relevant set detected falls below a preset threshold, which we usually set equal to 3.0.

As will be shown in Section 4, this version of the iterative sieve is quite effective. However, other variants may be implemented, which may produce more than one group variable per iteration. In this latter case, instead of considering just the top-ranked RS as a new group variable, one may possibly transform the first q sets into group variables, with q chosen according to some empirical criterion.

The iterations of the sieving algorithm come to an end when the T_c of the top-ranked RS falls below a preset threshold (usually equal to 3.0), which means it is no longer possible to find new RSs that deviate significantly from the behaviour of the reference system.

3. Implementation

The problem of finding the RSs of a complex system is a combinatorial optimization problem, whose size increases exponentially with the system dimension, soon making it infeasible to follow an exhaustive approach, in which T_c is computed for each possible subset of system variables. The computation of T_c itself is a rather lengthy procedure. Therefore, we tried to limit the computation time needed to run our method, on the one hand, by implementing the T_c computation algorithm as massively parallel GPU code [8], while on the other hand, by designing a metaheuristic, in which a genetic algorithm is hybridized with a local search. The latter is described in detail in the following subsections.

3.1. Metaheuristic-Based Search of the Relevant Sets. In the metaheuristic we developed, named HyReSS (Hybrid Relevant Set Search) [9], a genetic algorithm is first executed to address the search towards the basins of attraction of the main local maxima. Then, the results are refined through a series of local searches, driven by the statistics, computed at runtime, on the results that the algorithm is providing. These local searches explore the aforementioned basins of attraction more finely and extensively.

The overall metaheuristic can be partitioned into five main steps, which are performed according to the following sequence:

(1) Genetic algorithm

(2) Relevance-based local search

(3) Frequency-based local search

(4) Cardinality-based local search

(5) Merging

3.1.1. Genetic Algorithm. The first phase of the overall metaheuristic is a genetic algorithm, similar to Deterministic Crowding [19], aimed at boosting the niching characteristics of the overall method. As a matter of fact, HyReSS does not look for a single RS, but for the N_{best} RSs with the highest T_c.

Each individual corresponds to one RS and is represented by a binary string of size N, where each bit set to 1 denotes the inclusion in the RS of the corresponding variable out of the N variables that describe the system. A list ("best-RS memory," of size N_{best}, in the following) stores the best individuals and their corresponding fitness values. At the end of the run, it contains the N_{best} RSs which have been found.

The initial population has a dimension equal to p and is obtained by generating random individuals according to a certain preset distribution of cardinality (pairs, triplets, etc.). This type of generation aims at creating a sample as diversified as possible, which avoids repetitions and is a good representative of the whole search space.

The fitness function that has to be maximized is represented by the T_c itself and is implemented through a CUDA (https://developer.nvidia.com) kernel that can compute in parallel the fitness values of large blocks of individuals.

Evolution proceeds as follows:

(1) Selection of $p/2$ random pairs of individuals

(2) Creation of p children through a single-point crossover

(3) Replacement of the most similar parent having lower fitness with a child, as long as the child is not already a member of the population

Mutation is implemented as bit flips after each mating and is applied with a low probability denoted as P_{mut}.

As regards the third step of the above list, the algorithm is elitist, that is, a child is inserted in the new population only if its fitness is better than the fitness of the replaced parent. This implies that the overall fitness of the population increases monotonically over the generations.

The steps described above are iterated until the population cannot evolve anymore. If, after completing an iteration, the new generation is exactly the same as the previous one, new parents are randomly generated.

The evolutionary phase has two possible stopping conditions:

(i) The number of evaluations of the fitness function is above a certain threshold α_f.

(ii) New parents have been generated at least α_p times.

At the end of the evolutionary algorithm, the N_{best} fittest individuals are selected as input for the following phases.

3.1.2. Search Based on the Relevance of Variables. During the execution of the genetic algorithm, a presence coefficient (PC_i) and an absence coefficient (AC_i) are calculated for each variable i of the system to be analysed. The more frequently variable i is included in high-fitness RSs, the higher PC_i. The more frequently variable i is not included in high-fitness RSs, the higher AC_i.

To compute PC_i and AC_i, whenever iteration t of the genetic algorithm has terminated, a fitness threshold (τ), which separates high-fitness RSs from low-fitness ones, is set, corresponding to a certain percentile β of the whole fitness range. The value of the threshold is given by

$$\tau(t) = \text{minFitness} + (\text{maxFitness} - \text{minFitness}) * \beta. \quad (7)$$

PC_i and AC_i correspond to the sum of the fitness values of the RSs, having fitness greater than τ, in which the variable was present or absent, respectively. These

sums are cumulated over the generations and normalized with respect to the number of generations in which the corresponding RSs belonged to the population.

Considering these two coefficients, a further ratio, defined as $R_{ap,i} = \mathrm{AC}_i/\mathrm{PC}_i$, is calculated.

Finally, the variable is labelled as relevant if

(i) PC_i is greater than a threshold γ (i.e., it belongs to the γth percentile of the full range of PC_i values);

(ii) $R_{ap,i}$ is lower than a certain threshold δ.

The most relevant variables are recombined, during the local search procedure, with other, randomly chosen, variables. This recombination takes place according to the following steps:

(1) All possible subsets (made of simple combinations) of the most relevant variables having cardinality greater than 1 are calculated.

(2) For each cardinality of the subsets, the individual with the highest fitness is selected. These individuals are used to create new RSs by forcing the presence (absence) of relevant (irrelevant) variables and by randomly adding other variables into the RSs themselves.

(3) Each newly generated individual is evaluated and, if its fitness is higher than the fitness of the lowest-fitness individual in the best-RS memory, it substitutes the latter.

At the end of this search phase, a local search is performed again, this time within the neighbourhood of the best individual of the best-RS memory. The latter is updated if new higher-fitness individuals have been found.

3.1.3. Search Based on the Frequency of Variables. In this third phase, we replicate once more the same procedure used to generate new individuals and to explore the neighbourhood of the best one. However, in this case, we follow a different criterion, which considers the frequency with which each variable has been included in the RSs evaluated in the previous phases.

Such a frequency value is used to identify two classes of variables and to assign to each variable belonging to them a higher probability of being included in the newly generated RSs. These two classes are the following:

(1) Variables having a frequency much lower than the average

(2) Variables having a frequency much higher than the average

On the one hand, variables of the first type could have been previously "neglected"; thus, it is worth checking whether they are able to generate good individuals. On the other hand, variables of the second kind are very likely to actually have high relevance.

3.1.4. Search Based on the Group Cardinality. This is the last search phase of the metaheuristic, which exploits some indices computed during all previous phases. In particular, such indices are the $N - 2$ frequencies of occurrence in the previous steps of groups of each possible cardinality from 2 to $N - 1$. These indices are normalized with respect to the a priori probability of occurrence of groups of corresponding size, given by the corresponding binomial coefficient

$$\binom{N}{c}, \tag{8}$$

where N is the total number of variables and c the cardinality of the group. Thus, new RSs are randomly created, with probability inversely proportional to the normalized index corresponding to their size, and are possibly stored into the best-RS memory if their fitness is high enough.

3.1.5. Merging. This is the final phase of the metaheuristic. In this phase, a limited pool of variables is selected by considering all variables that are included in the highest-fitness RSs in the best-RS memory. This is done according to the following steps:

(1) A size θ for the pool is chosen.

(2) The best individuals are progressively OR-ed bitwise in decreasing order of fitness. The procedure starts from the best two RSs until the result of the bitwise OR contains θ bits set to 1, or all the RSs have been processed.

(3) An exhaustive search over all the possible RSs containing the selected variables is performed, and the best-RS memory is updated accordingly.

4. Results and Discussion

In this section, we present three case studies analysed by means of the proposed iterative RI method. In all the analyses performed on our test cases, we retain the highest-ranked RS of each sieve iteration, merge its variables into a new entity, treated from then on as a single variable replacing its component variables, and iterate the procedure until the algorithm reaches a stopping condition based on the T_c of the group detected in the last iteration.

The first two case studies have been chosen to validate the intrinsic properties of the method on systems for which a "ground truth" is available. In fact, the relationships between the variables of the first system have been hand-coded, and the dynamical behaviour of the second system is also very well known.

The first test case consists of two sets of data generated synthetically to assess the effectiveness of our method on a system which has clear and well-established dynamics and compare its performance with classical pair correlation techniques.

The second example is a deterministic simulation of a chemical system (a Catalytic Reaction Network, CRN)

described by 22 variables. Given its limited dimension, this system has been analysed using an exhaustive search over all possible variable subsets.

The third case study, denoted as Green Community (GC), features 136 variables, which represent the participation (presence or absence) of 136 people in a series of meetings, held during a project (the so-called Green Community project [20]). Given the size of the system, for which an exhaustive search is obviously infeasible independently of the available computational power, this case study has been analysed using the metaheuristic described in Section 3.1. For each iteration of the sieving algorithm, ten independent runs of the metaheuristic were performed to take its stochastic nature properly into account and assess the repeatability of its results. For each iteration, the results provided by this algorithm were almost identical in all ten runs (only occasionally, in one of the runs, one of the 50 highest-T_c RSs was different from the ones detected in the other nine runs). For the smaller-size systems for which the comparison was possible (iterations of the sieving algorithm on systems described by fewer than 30 variables), the results were the same provided by an exhaustive search based on the same parallel code.

The parameters regulating the behaviour of the metaheuristic were set as reported in Table 1.

For all case studies taken into consideration, tests have been performed on a Linux server equipped with two Intel Xeon E5-2620v4 (2×8 cores, 2.1 GHz) CPUs, 64 GB RAM, two NVIDIA GeForce GTX 1070 GPUs.

Moreover, given the stochastic nature of the generation of the reference system, 100 independent runs with different random seeds were executed to assess the performance of the algorithm.

Execution times are summarised in Table 2, and results are discussed in the following subsections. The execution time on the Green Community case study has been computed considering only one run of the metaheuristic for each iteration of the sieving algorithm. For the synthetic case, we only report the execution times of the tests with $p = 0.5$, since different settings produce only slightly different results.

In the following sections, we describe the three systems and provide an interpretation of the results we obtained in our experiments.

4.1. Synthetic Data. We designed two synthetic datasets generated by simple systems, whose dynamics are known, with the aim of assessing the effectiveness of the method, similarly to machine learning contexts in which the *ground truth* is known. The systems we consider are composed of networks whose nodes are updated in terms of nonlinear, possibly probabilistic, functions at discrete time steps. These systems can be considered as archetypal examples of more complex systems because their interactions are examples of typical interactions occurring in real systems. At the same time, as we will see in the following, the more-than-binary nature of the relationships makes it impossible to detect the dynamically relevant parts of the systems by using classical pair correlation techniques: on the contrary, given also its ability to directly deal with large-size groups, our methodology is able to correctly identify the hidden structures.

Table 1: Settings of the parameters of the metaheuristic. The parameters are defined in Section 3.1.

P_{mut}	p	α_f	α_p	β	γ	δ	θ
0.1	50,176	501,760	3	0.75	0.75	0.3	15

4.1.1. Synthetic Data 1: Test Case. The first system has been designed as a test case and is composed of three subsystems: $S = S_1 \cup S_2 \cup S_3$. Each of the first two subsystems is composed of a clique of three nodes, in which the state of each node at time $t + 1$ depends on the state of the other two nodes at time t. The update function is an *exclusive OR* (XOR) of the two input nodes. This function is a prototypical example of a function that depends on both inputs. The third system, S_3, is composed of six independent nodes, each assuming value 0 or 1 according to a Bernoulli distribution with probability 0.5. In summary, the system is composed of two Boolean networks (BNs) immersed in a noisy environment. The dynamics of these BNs is synchronous and deterministic; therefore, after a short transient, the BNs settle into an attractor. In this case, each clique has four fixed points ({(000),(011),(101),(110)}), each with a basin of attraction of size two. As previously done in the case of BNs, we consider the attractors as representative of the dynamics of the whole system; the frequency of occurrence of each attractor in the data to be analysed is proportional to its basin of attraction [5]. If S_1 and S_2 are independent, then we expect the method to be able to identify the two cliques, distinguishing them from the random nodes. Conversely, if a dependence between S_1 and S_2 exists, then the system should still detect the two cliques, but it should also identify, in a further step, a superset including both BNs. This happens because their state space is strongly constrained with respect to the possible states assumed by the six random nodes, which, as such, are expected to have a negligible T_c. To perform our test, we produced data representing a collection of state values of S according to the following procedure:

(1) The first three values $(S_1 = \{x_1, x_2, x_3\})$ are set by choosing at random one among the fixed points of the BN.

(2) The fourth to the sixth values $(S_2 = \{x_4, x_5, x_6\})$ are either an ordered copy of the previous values $(\{x_1, x_2, x_3\})$ with probability p, or they are set independently by choosing at random one fixed point with probability $1 - p$.

(3) The remaining nodes $(S_3 = \{x_7, x_8, \ldots, x_{12}\})$ are independently set to 0/1 with probability 0.5.

Note that if $p = 0$, then S_1 and S_2 are independent, otherwise, S_2 depends on S_1. The system is depicted in Figure 1.

A selection of representative results of the analysis of this first system is summarised in Table 3 for the case of

Table 2: Summary of the parameters of the analysed systems and execution times of the sieving algorithm.

System	Variables	Samples	Time (s)
Artificial data 1 ($p = 0.5$)	12	1000	7.81 ± 1.39
Artificial data 2 ($p = 0.5$)	18	1000	18.37 ± 0.14
Catalytic reactions network	22	312	32.1 ± 0.4
Green Community	136	101	3904 ± 463

independent cliques and Table 4 for the case with probabilistic dependence between S_1 and S_2 with $p = 0.75$. We can observe that in the case with $p = 0$, the method first detects S_1 and then S_2; in the third iteration, it groups S_1 and S_2 and adds an extra node, but the T_c value of this subset is very low. Indeed, in this case, after the second iteration, we cannot expect any further meaningful clustering of relevant variables. It is worth observing that, since S_1 and S_2 are independent, in the first iterations, the two sets of variables have a very similar T_c value. The case where S_1 and S_2 are dependent is quite trivial: the method already groups the variables composing the two systems in the very first iteration. We also generated data with $p = 0.25$, $p = 0.5$, and $p = 1$. In the cases with $p > 0.25$, results were qualitatively the same as in the case with $p = 0.75$, while for $p = 0.25$ which represents a mild dependence, results were comparable to independence.

4.1.2. Synthetic Data 2: System with Chain of Dependencies. We also produced a more elaborated variant of the previous system by introducing a further BN of the same kind as the previous ones. However, in this case, we imposed a gradual dependence among these three BNs according to the following procedure:

(1) The first three values ($S_1 = \{x_1, x_2, x_3\}$) are set by choosing at random one among the fixed points of the BN.

(2) The fourth to the sixth values ($S_2 = \{x_4, x_5, x_6\}$) are set by choosing at random one among the fixed points of the BN with probability $1 - p$, while they are computed by XOR-ing two randomly chosen nodes in S_1 with probability p.

(3) The same as for point 2 holds for $S_3 = \{x_7, x_8, x_9\}$, except that it depends upon the nodes in S_2.

(4) The remaining nodes ($S_4 = \{x_{10}, x_{11}, \ldots, x_{18}\}$) are independently set to 0/1 with probability 0.5.

The results of the application of the sieving algorithm to these artificially generated data deserve a detailed discussion. Table 5 summarises the results of the case where variables are independent ($p = 0$). We can observe that the method first detects S_1, but the T_c of S_2 and S_3 is comparable. In the second iteration, it identifies S_2 and, subsequently, S_3. The algorithm is then able to identify the three subsystems and to distinguish them from the noisy environment because in the fourth iteration it groups all three systems together.

When a mild dependence is introduced ($p = 0.25$), we should observe a consequent dependence of S_2 upon S_1 and of S_3 upon S_2. This is indeed what the sieving algorithm returns, as shown in Table 6. It is important to remark that, with this low level of dependence, systems S_2 and S_3 are still detected as single entities but are then grouped according to the chain of dependencies. This tendency starts to dilute for higher values of p up to $p = 1$, where there is a complete dependence of nodes in S_i from nodes in S_{i-1} (with $i = 2, 3$). Results for $p = 1$ are summarised in Table 7, where we can observe that S_1 is identified first; then, single nodes in S_2 are iteratively added until they form the set $S_1 + S_2$. Subsequently, the nodes in S_3 are added so as to group all nine variables in the BNs. We emphasise that the sieving algorithm correctly identifies and combines these groups according to the chained dependence among BNs we have introduced.

Finally, to assess the effectiveness of the method, we also analysed these two sets of data by performing a hierarchical clustering based on paired Pearson correlations between variables, as typically done when networks are analysed. As expected, this approach did not discover any relevant set because the main relations involve more than two variables. We are not claiming that our method outperforms any other method based on correlations, but just that, by its nature, it is able to capture relations involving multiple variables.

The successful outcome of the application of the sieving algorithm to these artificially built datasets provides evidence to the effectiveness of the method. Nevertheless, its potential should be expressed on more complex cases, which are the subject of the following two sections in which data from a Catalytic Reaction Network and the Green Community data are analysed.

4.2. Catalytic Reaction Network. The formation of groups of molecules able to collectively self-replicate is thought to be fundamental for the origin of life [21–27] and is likely to play an important role also in future bio-technological systems [28]. Indeed, currently living beings are based on self-replicating chemical structures, where the presence of enzymes (biological catalysers) plays an essential role.

Many attempts have been made to determine the chemical arrangements that allow sustainable self-maintaining behaviours, one of the currently most sophisticated being probably Reflexive Autocatalytic Food-generated (RAF) sets [29, 30], recently utilized also in biochemical contexts [29, 31–33] or in protocell architectures [34].

On the other hand, efforts have been made to identify the relevant chemical species involved in real or artificial complex chemical reaction schemes [35, 36].

Typically, the systems we have analysed are immersed in Continuous-flow Stirred-Tank Reactors (CSTRs) [37], featuring a constant influx of feed molecules (constantly present in CSTRs and therefore playing the role of the "food" species constituting the base of RAF arrangements) and a continuous outgoing flux of all the molecular species proportional to their concentration. In this case, as the typical attractors are fixed points, which do not provide any useful information for computing the RI, a different

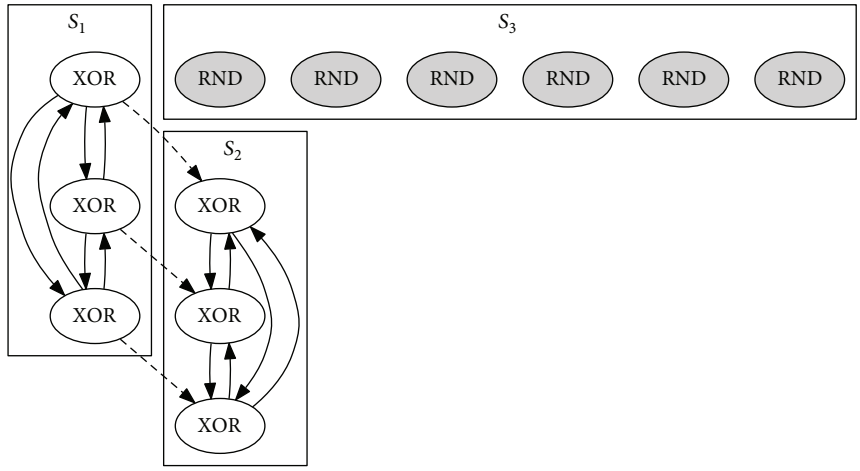

Figure 1: A system composed of three subsystems: S_1 and S_2 are two cliques with XOR functional dependencies. The nodes of S_2 depend on the nodes of S_1 with probability p. The third subsystem S_3 is composed of independent random Boolean nodes.

Table 3: *Test case:* RSs found in the main iterations of the sieving algorithm and corresponding T_c values with $p = 0$. At the end of each iteration, the RS with the highest T_c is grouped into a single variable for the next iteration. In iteration 1, we show also the second RS ranked. Note that, in iteration 3, the T_c of group $(S_1\ S_2\ x_{11})$ (lower than the chosen threshold of 3.0) makes it not relevant.

Synthetic data 1: independent cliques ($p = 0$)		
Iteration	Relevant set(s)	T_c
1	$x_1\ x_2\ x_3 \longrightarrow (S_1)$	1517.01
	$x_4\ x_5\ x_6$	1454.52
2	$x_4\ x_5\ x_6 \longrightarrow (S_2)$	684.923
3	$S_1\ S_2\ x_{11}$	1.644

Table 4: *Test case:* RSs found in the main iterations of the sieving algorithm and corresponding T_c values with $p = 0.75$. At the end of each iteration, the RS with the highest T_c is grouped into a single variable for the next iteration. Note that, in iteration 2, the T_c of the group $(S_1 + S_2\ x_9)$ (lower than the chosen threshold of 3.0) makes it not relevant.

Synthetic data 1: dependent cliques ($p = 0.75$)		
Iteration	Relevant set	T_c
1	$x_1\ x_2\ x_3\ x_4\ x_5\ x_6 \longrightarrow (S_1 + S_2)$	1524.96
2	$S_1 + S_2\ x_9$	2.769

approach has been followed, which consists in perturbing the fixed points and recording the transient.

By taking advantage of the aforementioned perturbative approach, we apply the iterative sieving to the data series coming from the perturbations imposed on the simulation of chemical arrangements. This allows us to investigate the effectiveness of the RI method in identifying RSs in systems where several reactions take place simultaneously, using only the concentrations of the various chemical species as input and without any prior knowledge about the reaction graph. The simulations are based on a relatively simple system inspired by a model used in [38–40] and originally proposed by Kauffman [26, 41].

Table 5: *Chain of dependencies case:* RSs found in the main iterations of the sieving algorithm and corresponding T_c values with $p = 0$. In iteration 1, we also show the RSs ranked second and third. At the end of each iteration, the RS with the highest T_c is grouped into a single variable for the next iteration.

Synthetic data 2: independent cliques ($p = 0$)		
Iteration	Relevant set(s)	T_c
1	$x_1\ x_2\ x_3 \longrightarrow (S_1)$	838.024
	$x_4\ x_5\ x_6$	836.677
	$x_7\ x_8\ x_9$	831.635
2	$x_4\ x_5\ x_6 \longrightarrow (S_2)$	664.485
3	$x_7\ x_8\ x_9 \longrightarrow (S_3)$	576.078
4	$S_1\ S_2\ S_3$	7.013

Table 6: *Chain of dependencies case*: RSs found in the main iterations of the sieving algorithm and corresponding T_c values with $p = 0.25$. At the end of each iteration, the RS with the highest T_c is grouped into a single variable for the next iteration. Note that, in iteration 6, the T_c of the group detected (lower than the chosen threshold of 3.0) makes it not relevant.

Synthetic data 2: $p = 0.25$		
Iteration	Relevant set	T_c
1	$x_1\ x_2\ x_3 \longrightarrow (S_1)$	815.723
2	$x_4\ x_5\ x_6 \longrightarrow (S_2)$	325.067
3	$x_7\ x_8\ x_9 \longrightarrow (S_3)$	152.263
4	$S_1\ S_2 \longrightarrow (S_1 + S_2)$	17.209
5	$S_1 + S_2\ S_3 \longrightarrow (S_1 + S_2 + S_3)$	9.330
6	$S_1 + S_2 + S_3\ x_{16}$	2.786

Table 7: *Chain of dependencies case*: RSs found in the main iterations of the sieving algorithm and corresponding T_c values with $p = 1$. At the end of each iteration, the RS with the highest T_c is grouped into a single variable for the next iteration.

Iteration	Synthetic data 2: $p = 1$ Relevant set	T_c
1	$x_1\ x_2\ x_3 \longrightarrow (S_1)$	900.645
2	$S_1\ x_4\ x_6$	267.988
3	$S_1 + x_4 + x_6\ x_5 \longrightarrow (S_1 + S_2)$	122.419
4	$S_1 + S_2\ x_8$	50.414
5	$S_1 + S_2 + x_8\ x_7$	23.408
6	$S_1 + S_2 + x_8 + x_7\ x_9 \longrightarrow (S_1 + S_2 + S_3)$	12.222

The analysed scheme involves enzymatic condensations, whose process is considered as being composed of three steps: the first two creates (reversibly) a temporary complex (composed by one of the two substrates and the catalyst) that can be used by a third reaction, which combines the complex and a second substrate to finally release the catalyst and the final product. The aforementioned three steps are summarised as follows:

(1) Complex formation: $\mathrm{A} + \mathrm{C} \xrightarrow{C_{\mathrm{comp}}} \mathrm{A} : \mathrm{C}$.

(2) Complex dissociation: $\mathrm{A} : \mathrm{C} \xrightarrow{C_{\mathrm{diss}}} \mathrm{A} + \mathrm{C}$.

(3) Final condensation: $\mathrm{A} : \mathrm{C} + \mathrm{B} \xrightarrow{C_{\mathrm{cond}}} \mathrm{AB} + \mathrm{C}$.

C_{comp}, C_{diss}, and C_{cond} are respectively the reaction kinetic constants of complex formation, complex dissociation, and final condensation. The dynamic of the systems is described adopting a deterministic approach, whereby the reaction scheme is translated into a set of Ordinary Differential Equations ruled by the mass action law (see [34] for further details) and integrated by means of a custom Euler method with step-size control.

The main entities of the model are molecular species ("polymers"), represented by linear strings of letters (A, B, C, and D). In the example of Figure 2, they form a catalytic reaction system composed of seven distinct condensation reactions divided into two distinct RAF pathways: a chain of linear reactions (RAF1), the presence of whose root is guaranteed from the outside, and a RAF where two reciprocally catalysing reactions are the roots of another linear reaction chain (RAF2).

As mentioned before, the asymptotic behaviour of this kind of systems is a single fixed point [35] due to the system feedback structure. In order to apply our analysis, we need to observe the feedbacks in action. Therefore, we perturbed the concentration of some molecules in order to trigger a response in the concentration of (some) other species. Therefore, we temporarily lowered, one by one, by two orders of magnitude, the input concentrations of the food species (coloured ellipses in the example of Figure 2) after the system reached its stationary state. Note that we could also simulate the temporarily disappearance of the chemical species inside the CSTR vessel: in this case (i) the grouping process would be different (a consideration that highlights the fact that the perturbation itself is a dynamical process with a significant influence on the final observations) and (ii) the identification of the chemical structures would clearly be easier. However, this procedure is not feasible in real experiments. In order to analyse the system response to perturbations, we used a three-level coding where, for each species, the digits 0–1–2 stand for "concentration decreasing," "no change," and "concentration increasing," respectively. In this experiment, we consider the concentration of a chemical species as being constant if it has not changed by more than 1% in a time period of 10 seconds. In practice, the time series is obtained by computing (and then properly coding) the sign of the difference between two consecutive samples of the original data. Note that, in order to better observe the dependencies of the system, we set an observation frequency high enough to allow several observations during the transient situations. In other words, the transients are "smooth."

In Figure 2, we report the most salient steps of the analysis, while the T_c values computed for the main groups are reported in Table 8.

The entire linear reaction chain (RAF1) is immediately detected whereas, within the more complex RAF2 organization, the other groups highlight the strict relations among the reagents and the catalyser of the same reaction (Figures 2(b) and 2(c)). The time series is too short to permit a complete detection of the whole RAF2 group (Figure 2(c)) with sufficiently high significance; however, in the subsequent iterations of the method, although groups with significance below the chosen threshold (corresponding to a critical value of T_c equal to 3.0) are found, we can actually detect the correct configurations. Indeed, we noticed that, if the time series length is artificially duplicated, an increased T_c value can also be computed for these groups, which confirms the correctness of their detection. All data are represented in Table 8, which displays the T_c values of the RS detected in each iteration of the sieving algorithm.

Note that the relatively long chain in RAF2 (composed by reactions R2, R3, R4, and R5) is "discovered" by our approach starting from the final part of the tail and subsequently "going upward" towards its head (Figures 2(c) and 2(d)). This effect is due to the perturbations on the "last" food species of the chain (e.g., CA, CB, or AC), which heavily affect the final part of this chain. However, their effects cannot propagate towards the species (e.g., BAB or AAB) that are located "upward" along the chain. On the contrary, perturbations on BA, B, or AA heavily affect the initial part of this chain, as well as its final part—the higher the distance from its initial source, the weaker the effect. This attenuation process (observed also in [36]) induces a dynamical hierarchy on the chain system, which permits the fine subdivision highlighted in Figure 2. The same phenomenon is not observable in RAF1, on the one hand, because of the small size of the chain and, on the other hand, because the perturbations hit the root of the chain directly, causing strong and evident effects along the whole (short) structure. We remind that the "root" of RAF2 is composed of two reciprocally catalysing reactions: indeed,

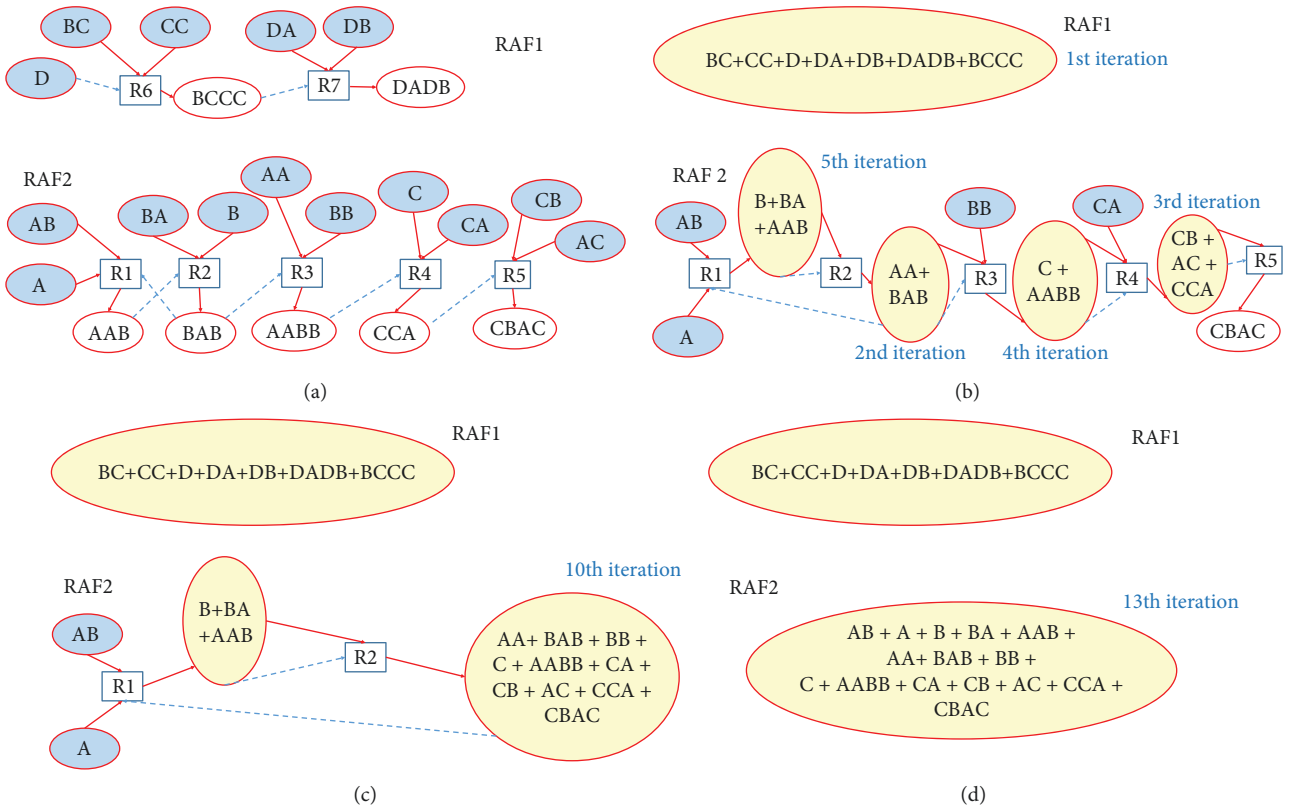

FIGURE 2: (a) The chemical system under analysis in the CRN case study. Elliptic nodes represent chemical species: the ones filled in blue stand for those injected into the CSTR (food species), while the empty white ones are the more complex species built by specific condensation of the food species. Rectangular shapes represent reactions, where incoming arrows are oriented from substrates to reactions and outgoing arrows from reactions to products. Dashed lines indicate the catalytic role of a particular molecular species within the specific reaction context. The kinetic constants of all reactions which take place have the same value ($C_{comp} = 5000\ mol^{-1}\ s^{-1}$; $C_{diss} = 250\ s^{-1}$; and $C_{cond} = 500\ mol^{-1}\ s^{-1}$); the incoming concentration of each food species is 0.01 mol, whereas, every second, 5% of the CSTR volume is renewed. (b) The five RSs found after the first five iterations of the iterated sieving algorithm. (c) The situation at the end of the iterated sieving algorithm. Note, however, that, if the method is further iterated (d), despite finding groups with significance below the chosen threshold (corresponding to a critical value of T_c close to 3.0), the iterative sieve detects the correct configuration anyway. For the sake of completeness, the results of all sieving steps are reported in the attached Supplementary Materials (available here).

this strong dynamical union permits interesting resilience effects [30, 33, 34].

4.3. Green Community. In this subsection, we examine a set of data extracted from a very large and complex corpus collected during the monitoring of the Green Communities (GC) project. The project started in Italy in 2012, within a call supported by the EU Interregional Operational Program; it initially involved only four mountain communities, dealing with a core topic about energy efficiency and renewable energy production-related issues. Later, it was extended to other social and organizational themes, gradually involving many heterogeneous stakeholders, including specialists, engineers, researchers, local administrations, and representatives. In order to manage this increasing complexity, the project decided to take advantage of an evaluation approach, the Dynamic Evaluation, developed within the "Emergence by Design" European project (http://cordis.europa.eu/project/rcn/102441_en.html), aimed at supporting the monitoring, management, and development of projects and programs [20].

The data are extensive and rather complex; on the one hand, the overall data structure was not designed with the RI application in mind; on the other hand, the data do contain information that social scientists are not used to analyse and that could be analysed successfully through our techniques.

Therefore, in order to observe the presence of (formal or informal) coalitions within the four mountain communities, we decided to extract from this database only the data about the involved stakeholders' attendance (or absence) at some significant points-of-control of the GC project that are very heterogeneous and include official meetings, global conferences, other relevant panels, and e-mail discussions. Moreover, they are distributed over time without a predefined frequency. Following an indication presented also in [9], our idea was that the simple presence or absence at important meetings (or even the mere permission to participate in them) could carry significant information about the real project profiles. Therefore, we obtained a very sparse matrix composed of 136 variables and 101 points-of-control (observations) (see Figure 3(a)).

Table 8: RSs found in each iteration of the sieving algorithm and corresponding T_c values for the CRN case study. The last three iterations are separated from the others because the T_c values of the detected RSs are lower than the threshold of 3.0.

Iteration	Detected relevant set	T_c
1	$[BC][CC][D][DA][DB][DADB][BCCC]$	305.711
2	$[AA][BAB]$	33.682
3	$[CB][AC][CCA]$	28.616
4	$[C][AABB]$	26.048
5	$[B][BA][AAB]$	24.293
6	$[C + AABB][CA]$	14.236
7	$[CB + AC + CCA][CBAC]$	10.762
8	$[C + AABB + CA][CB + AC + CCA + CBAC]$	11.535
9	$[AA + BAB][BB]$	5.213
10	$[AA + BAB + BB]$ $[C + AABB + CA + CB + AC + CCA + CBAC]$	4.484
11	$[B + BA + AAB][AA + BAB + BB + C +$ $AABB + CA + CB + AC + CCA + CBAC]$	2.953
12	$[A][B + BA + AAB + AA + BAB + BB + C +$ $AABB + CA + CB + AC + CCA + CBAC]$	0.957
13	$[AB][A + B + BA + AAB + AA + BAB + BB +$ $C + AABB + CA + CB + AC + CCA + CBAC]$	0.261

Analysing this matrix, we were able to infer some insightful indications about the formation of coalitions during the GC project.

However, considering the subject of the present paper, we simply report the following observations:

(i) The iterated sieving procedure automatically stopped after 26 iterations when a T_c lower than the threshold of 3.0 characterised the last detected RS, resulting in the final organization shown in Figure 3(b).

(ii) The groups exhibiting the simplest behaviours are composed of stakeholders (the system "variables" or "agents" in the following) present at only one event of the GC project. In fact, some events may have a restricted list of stakeholders that are allowed to participate in it: this piece of information is indeed a significant part of the process itself and forces particular forms of coordination among agents (it excludes the agents that wish to participate in a particular event but do not have the correct permission and could also force the presence of agents that would not participate). On the other hand, the detection of these simple groups is a confirmation of the correctness of the RI procedure when dealing with real-world data.

(iii) Groups D and B, despite their illusory simplicity, are not so obvious: in particular, the explicit recognition of group D (and group B) indicates the existence of a distance between the variables belonging to these groups and other variables that, in spite of exhibiting similar behaviours, belong to other groups.

(iv) The large group named G is composed of several large subgroups which, in turn, may be composed of other smaller RSs. Some of these groups include very active variables (in particular group G7, which includes the head of the GC project and the two social researchers involved in the observation of the whole project), whereas other groups, despite the relatively low activity of their members, are dynamically very heterogeneous (e.g., group G2).

Therefore, the detection of the most obvious groups, whose correctness was confirmed by the social researchers that collected the data, shows that the iterative RI procedure correctly works in analysing the GC case. Moreover, the less obvious groups have been considered very "interesting" and sometimes "enlightening" by these specialists.

In any case, the final comment of the social specialists involved in the project was that "the RI methodology could constitute a very interesting *dashboard* potentially able to effectively support the fieldwork of the observers" (personal communication).

As a final observation, we can notice how even a measurement as simple as the recording of the presence/absence in (formal or informal) project meetings can result in the detection of very sophisticated groupings or hierarchies.

5. Conclusions

In this paper, we have formally introduced, for the first time, a methodology able to support a wide application of the RI method. The technique we propose realizes a sieving action that is performed iteratively until a certain threshold in the T_c value is reached and permits to group together variables (or sets of variables) of a complex system, which are detected as the most relevant by the RI method.

The iterated sieving algorithm introduced into the method aims to reduce the overall number of subsets found by such a method. This is done by keeping only disjoint or partially overlapping subsets of variables, which means that only the subsets having the highest RI are taken into consideration in defining the architecture of the whole complex system. In the end, this appears to be built upon variable subsets that cannot be decomposed any further and represent the actual building blocks of the system.

The proposed approach, based on information-theoretic measures, has proven to be able to extract hidden information about the organization of the three complex systems we have analysed.

Regarding future work, we plan to apply the RI method, enriched with the sieving capability, to several other complex systems: social networks, biological networks, or sociotechnological systems. This can be done quite easily because it can be applied to systems characterised by both continuous and discrete (Boolean or multivalued) variables. The ultimate objective is twofold and encompasses both finding new insights about those systems and further refining the method.

In particular, from a methodological point of view, considering that the RI is a ratio and that the same RI values can be obtained by different pairs of values of Integration and

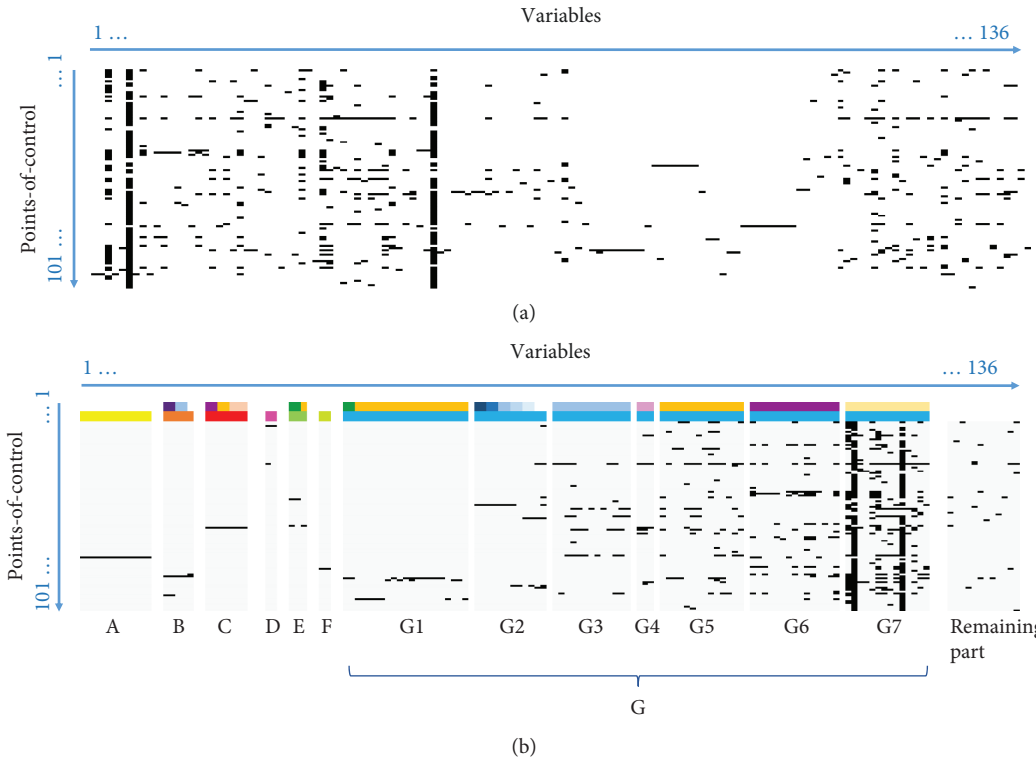

Figure 3: (a) The temporal behaviour of the 136 variables over the 101 points-of-control, which are distributed over time without a regular frequency. The involved stakeholders' attendance at the points-of-control of the GC project is marked in black (the absence is marked in white). (b) The same variables resorted in order to highlight the RSs found by the RI algorithm. The two coloured bars highlight (i) the different "final" groups (lower coloured bar) and (ii) the internal subdivisions of the final groups (the smallest RSs detected during the 26 iterations of the sieving procedure). Group G was obtained by grouping smaller RSs: for easier graphical representation, the exact multistep sequence of the assembly has not been represented. The same observations hold for the assembly processes in general: for example, the steps leading to grouping RSs B, C, E, G1, G2, and G3 are intertwined, while groups G4 and G3 are formed "before" the final expression of group G1. The other variables (termed as the "remaining part") could not be assigned to any group with a sufficiently high degree of significance.

Mutual Information, we will investigate how the statistical distribution of the values of the two terms of the ratio affects the actual relevance of the index computed from such pairs.

From the point of view of the applications of the method, we are interested in studying more systems with a large number of variables, whose RSs cannot be computed exhaustively. For such systems, the use of the metaheuristic described in this paper, as well as its further development to fit different types of input data, will be necessary to find the relevant sets in a reasonable time.

Data Availability

The data used in the experiments described in this paper have been generated or gathered during the EU-funded project "Emergence by Design (MD)" and are currently available only to the members of the research consortium involved in that project.

Conflicts of Interest

All authors declare no conflict of interest.

Acknowledgments

Marco Villani, Roberto Serra, and Andrea Roli are grateful to partners and staff involved in the "Emergence by Design (MD)" European project [FP7/2007–2013] (in particular to David Lane, Paolo Gurisatti, Margherita Russo, Stefania Sardo, and Valentina Anzoise) for their support and the precious discussions. The authors are grateful to the four mountain communities involved in the Green Communities project. Part of this research has been carried out at the High Performance Computing (HPC) facility of the University of Parma. The work of Michele Amoretti was supported by the University of Parma Research Fund (FIL 2016) Project "NEXTALGO: Efficient Algorithms for Next-Generation Distributed Systems."

Supplementary Materials

A PowerPoint file is provided as a more detailed description of the experiments summarised in Figure 2. Each slide of the presentation describes the results of one step of the sieving algorithm applied to the catalytic reactions network. *(Supplementary Materials)*

References

[1] R. Badii and A. Politi, *Complexity–Hierarchical Structures and Scaling in Physics. Cambridge Nonlinear Science Series*, Cambridge University Press, 1997.

[2] H. Simon, "The architecture of complexity," *Proceedings of the American Philosophical Society*, vol. 106, no. 6, 1962.

[3] Y. Bar-Yam, *Dynamics of Complex Systems*, Studies in nonlinearity., Addison–Wesley, Reading, MA, 1997.

[4] M. Villani, P. Carra, A. Roli, A. Filisetti, and R. Serra, "On the robustness of the detection of relevant sets in complex dynamical systems," in *Advances in Artificial Life, Evolutionary Computation and Systems Chemistry*, F. Rossi, F. Mavelli, P. Stano, and D. Caivano, Eds., pp. 15–28, Springer International Publishing, Cham, 2016.

[5] M. Villani, A. Roli, A. Filisetti, M. Fiorucci, I. Poli, and R. Serra, "The search for candidate relevant subsets of variables in complex systems," *Artificial Life*, vol. 21, no. 4, pp. 412–431, 2015.

[6] R. Righi, A. Roli, M. Russo, R. Serra, and M. Villani, "New paths for the application of dci in social sciences: theoretical issues regarding an empirical analysis," in *Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry*, F. Rossi, S. Piotto, and S. Concilio, Eds., pp. 42–52, Springer International Publishing, Cham, 2017.

[7] A. Roli, M. Villani, R. Caprari, and R. Serra, "Identifying critical states through the relevance index," *Entropy*, vol. 19, no. 73, pp. 1–15, 2017.

[8] E. Vicari, M. Amoretti, L. Sani et al., "GPU-based parallel search of relevant variable sets in complex systems," in *Advances in Artificial Life, Evolutionary Computation, and Systems Chemistry*, F. Rossi, S. Piotto, and S. Concilio, Eds., pp. 14–25, Springer International Publishing, Cham, 2017.

[9] L. Sani, M. Amoretti, E. Vicari et al., "Efficient search of relevant structures in complex systems," in *Conference of the Italian Association for Artificial Intelligence*, pp. 35–48, Springer, 2016.

[10] G. Silvestri, L. Sani, M. Amoretti et al., "Searching relevant variable subsets in complex systems using k-means PSO," in *Artificial Life and Evolutionary Computation*, M. Pelillo, I. Poli, A. Roli, R. Serra, D. Slanzi, and M. Villani, Eds., pp. 308–321, Springer International Publishing, Cham, 2018.

[11] U.-U. Narantsatsralt and S. Kang, "Social network community detection using agglomerative spectral clustering," *Complexity*, vol. 2017, Article ID 3719428, 10 pages, 2017.

[12] L. Sani, G. Lombardo, R. Pecori, P. Fornacciari, M. Mordonini, and S. Cagnoni, "Social relevance index for studying communities in a Facebook group of patients," in *Applications of Evolutionary Computation*, K. Sim and P. Kaufmann, Eds., pp. 125–140, Springer International Publishing, Cham, 2018.

[13] A. Filisetti, M. Villani, A. Roli, M. Fiorucci, and R. Serra, "Exploring the organisation of complex systems through the dynamical interactions among their relevant subsets," *ECAL 2015: The Thirteenth European Conference on Artificial Life*, 2015, pp. 286–293, The MIT Press, York, UK, July 2015.

[14] G. Tononi, A. R. McIntosh, D. Patrick Russell, and G. M. Edelman, "Functional clustering: identifying strongly interactive brain regions in neuroimaging data," *NeuroImage*, vol. 7, no. 2, pp. 133–149, 1998.

[15] M. Villani, A. Filisetti, A. Graudenzi, C. Damiani, T. Carletti, and R. Serra, "Growth and division in a dynamic protocell model," *Life*, vol. 4, no. 4, pp. 837–864, 2014.

[16] T. Cover and A. Thomas, *Elements of Information Theory*, WileyInterscience, New York, 2nd edition, 2006.

[17] M. Prokopenko, F. Boschetti, and A. J. Ryan, "An information-theoretic primer on complexity, self-organization, and emergence," *Complexity*, vol. 15, no. 1, pp. 11–28, 2009.

[18] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo, "Model-based search for combinatorial optimization: a critical survey," *Annals of Operations Research*, vol. 131, no. 1–4, pp. 373–395, 2004.

[19] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Natural Computing Series, Springer-Verlag, Berlin Heidelberg, 2nd edition, 2015.

[20] V. Anzoise and S. Sardo, "Dynamic systems and the role of evaluation: the case of the green communities project," *Evaluation and Program Planning*, vol. 54, pp. 162–172, 2016.

[21] F. J. Dyson, *Origins of Life*, Cambridge University Press, Cambridge, UK, 1985.

[22] M. Eigen and P. Schuster, "A principle of natural self-organization. Part A: emergence of the hypercycle," *Naturwissenschaften*, vol. 64, no. 11, pp. 541–565, 1977.

[23] M. Eigen and P. Schuster, "The hypercycle: a principle of natural self-organization, Part B: the abstract hypercycle," *Naturwissenschaften*, vol. 65, no. 1, pp. 7–41, 1978.

[24] A. Filisetti, R. Serra, T. Carletti, M. Villani, and I. Poli, "Non-linear protocell models: synchronization and chaos," *The European Physical Journal B*, vol. 77, no. 2, pp. 249–256, 2010.

[25] S. Jain and S. Krishna, "Autocatalytic sets and the growth of complexity in an evolutionary model," *Physical Review Letters*, vol. 81, no. 25, pp. 5684–5687, 1998.

[26] S. A. Kauffman, *The Origins of Order*, Oxford University Press, Oxford, UK, 1993.

[27] K. Ruiz-Mirazo, C. Briones, and A. de la Escosura, "Prebiotic systems chemistry: new perspectives for the origins of life," *Chemical Reviews*, vol. 114, no. 1, pp. 285–366, 2014.

[28] R. V. Sole, A. Munteanu, C. Rodriguez-Caso, and J. Macıa, "Synthetic protocell biology: from reproduction to computation," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 362, no. 1486, pp. 1727–1739, 2007.

[29] W. Hordijk, J. Hein, and M. Steel, "Autocatalytic sets and the origin of life," *Entropy*, vol. 12, no. 7, pp. 1733–1742, 2010.

[30] W. Hordijk and M. Steel, "Detecting autocatalytic, self-sustaining sets in chemical reaction systems," *Journal of Theoretical Biology*, vol. 227, no. 4, pp. 451–461, 2004.

[31] A. Filisetti, M. Villani, C. Damiani et al., "On RAF sets and autocatalytic cycles in random reaction networks," in *Advances in Artificial Life and Evolutionary Computation*, C. Pizzuti and G. Spezzano, Eds., pp. 113–126, Springer International Publishing, Cham, 2014.

[32] W. Hordijk and M. Steel, "A formal model of autocatalytic sets emerging in an RNA replicator system," *Journal of Systems Chemistry*, vol. 4, no. 1, p. 3, 2013.

[33] V. Vasas, C. Fernando, M. Santos, S. Kauffman, and E. Szathmary, "Evolution before genes," *Biology Direct*, vol. 7, no. 1, p. 1, 2012.

[34] R. Serra and M. Villani, *Modelling Protocells: The Emergent Synchronization of Reproduction and Molecular Replication*, Springer Netherlands, Dordrecht, 2017.

[35] A. Arkin, P. Shen, and J. Ross, "A test case of correlation metric construction of a reaction pathway from measurements," *Science*, vol. 277, no. 5330, pp. 1275–1279, 1997.

[36] W. Vance, A. Arkin, and J. Ross, "Determination of causal connectivities of species in reaction networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5816–5821, 2002.

[37] R. Perry and D. Green, *Perry's Chemical Engineers' Handbook*, McGraw-Hill, 8th edition, 2007.

[38] J. Doyne Farmer, S. A. Kauffman, and N. H. Packard, "Autocatalytic replication of polymers," *Physica D: Nonlinear Phenomena*, vol. 22, no. 1–3, pp. 50–67, 1986, Proceedings of the Fifth Annual International Conference.

[39] A. Filisetti, A. Graudenzi, R. Serra et al., "A stochastic model of the emergence of autocatalytic cycles," *Journal of Systems Chemistry*, vol. 2, no. 1, p. 2, 2011.

[40] A. Filisetti, A. Graudenzi, R. Serra et al., "A stochastic model of autocatalytic reaction networks," *Theory in Biosciences*, vol. 131, no. 2, pp. 85–93, 2012.

[41] S. A. Kauffman, *At Home in the Universe: The Search for Laws of Self-Organization and Complexity*, Oxford University Press, Oxford, UK, 1993.