Analysis of the robustness and recovery of critical infrastructures by Goal Tree Success Tree – Dynamic Master Logic Diagram, within a multi-state system-of-systems framework, in the presence of epistemic uncertainty

Elisa Ferrario, Nicola Pedroni, Enrico Zio

Analysis of the robustness and recovery of critical infrastructures by Goal Tree Success Tree – Dynamic Master Logic Diagram, within a multi-state system-of-systems framework, in the presence of epistemic uncertainty

E. Ferrario[a], *N. Pedroni*[a] *and E. Zio*[a,b]

[a]*Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy - Electricité de France, at École Centrale Paris - Supelec, France*

enrico.zio@ecp.fr, enrico.zio@supelec.fr

[b]*Department of Energy, Politecnico di Milano, Italy*

enrico.zio@polimi.it

ABSTRACT

In this paper, we look at the robustness and recovery of connected critical infrastructures (CIs) under a system-of-systems (SoS) framework taking into account i) the dependencies among the components of an individual CI and the interdependencies among different CIs; ii) the variability in component performance, by a multi-state model; iii) the epistemic uncertainty in the probabilities of transitions between different components states and in the mean values of the holding times distributions, by means of intervals. We adopt the Goal Tree Success Tree – Dynamic Master Logic Diagram (GTST-DMLD) for system modelling and perform the quantitative assessment by Monte Carlo simulation. We illustrate the approach by way of a simplified case study consisting of two interdependent infrastructures (electric power system and gas network) and a supervisory control and data acquisition (SCADA) system connected to the gas network.

Keywords: critical infrastructures; electric power system, gas distribution network, SCADA, robustness; recovery time; multi-state; Goal Tree Success Tree – Dynamic Master Logic Diagram; Monte Carlo simulation; epistemic uncertainty; imprecise probability; interval analysis

1. INTRODUCTION

Critical infrastructures (CIs), e.g., transportation, electric power, water, gas, communication systems, interact on the basis of complex relationships that cross the single infrastructure boundary. This exposes CIs to the risk that a failure in an infrastructure can have negative impacts on another interconnected one. For example, CIs are getting more and more dependent on information technologies that, on one hand, provide control and support their increasing efficiency, but, on the other hand, create new vulnerabilities [1]. As additional example from the field, the widespread power electric blackout that occurred in the Midwest and Northeast of the United States and Ontario, Canada, on August 2003, affected the serviceability of the water system at Cleveland, OH, due to the lack of power needed to operate the water pumping stations [2]. Analyzing and understanding the interdependences existing among infrastructure systems is fundamental for the safe operation and control of these "systems of systems".

We adopt a system-of-systems (SoS) framework of analysis to evaluate the SoS robustness and recovery properties, considering the dependencies among the components of a critical infrastructure and the interdependencies among different CIs. For a more realistic representation, we utilize a multi-state model for consideration of the different degrees of damage that the individual components may experience [3]. Transitions between different states of damage occur stochastically (aleatory uncertainty) and epistemic uncertainty affects the associated transition probabilities due to insufficient knowledge and information on the components degradation behavior [4-6]. Indeed, safety-CIs are highly reliable and, thus, undergo few degradations to failure, so that it is difficult to estimate damage levels and transition probabilities [7-11].

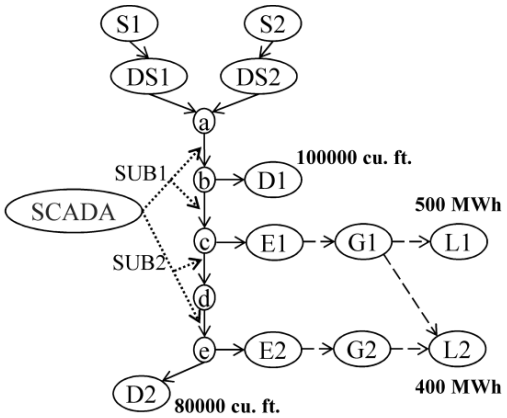
For illustration purpose, we adapt the framework of analysis to a case study proposed in [1], in which the system considered consists of two interdependent infrastructures (gas and electric power networks), and a supervisory control and data acquisition (SCADA) system connected to the gas network. To measure the robustness and recovery capacity of the system, we look at the steady-state probability distributions of the supply of gas and electricity at the demand nodes and the time needed to recover the SoS from the worst scenario to a level in which all the demand nodes are satisfied, respectively.

We propose a hierarchical model description of the system logic and functionality by Goal Tree Success Tree – Dynamic Master Logic Diagram (GTST-DMLD) [12], extending its representation characteristics to evaluate the physical flows of gas and electricity through the interdependent infrastructures. We adopt intervals to describe the epistemic uncertainty in the probabilities of transition between different components states and in the mean values of the holding time distributions [13-21] and we use interval analysis to calculate the (uncertain) probabilities of the states of all the components of the CIs [22-27]. Finally, we employ Monte Carlo simulation [28, 29] for the probabilistic evaluation of the SoS performance.

The paper is organized as follows. In Section 2, the case study is presented; in Section 3, the SoS modelling by GTST-DMLD is illustrated; in Section 4, details of the procedural steps to evaluate the SoS performance under epistemic uncertainty are given; in Section 5, the results of the analysis are shown and discussed; in Section 6, conclusions are provided. Finally, in Appendix A, a brief overview of imprecise probabilities is given and in Appendix B further details of the operative steps to process the epistemic uncertainty by interval analysis are reported.

2. CASE STUDY

The case study is taken from [1] and it deals with two interconnected infrastructures, i.e., a natural gas distribution network and an electricity generation/distribution network (Figure 1 top, solid and dashed lines, respectively). The gas distribution network is supported by a SCADA system (Figure 1 top, dotted lines). The objective of this interconnected system of systems (SoS) is to provide the necessary amount of gas and electricity (hereafter also called "product") to four demand nodes (end-nodes), namely D1 and D2 (gas) and L1 and L2 (electricity).



Arc capacities

Deterministic	**Stochastic**	
DS1_a: 100000 cu. ft.	S1_DS1: [90, 95, 100, 105]*10^3 cu. ft.	E1_G1: [600, 800] MWh
DS2_a: 200000 cu. ft.	S2_DS2: [100, 160, 200]*10^3 cu. ft.	E2_G2: [0, 250, 400] MWh
b_D1: 100000 cu. ft.	SUB1: [0, 1]	SUB2: [0, 1]
c_E1: 80000 cu. ft.	*If SUB1 = 0*	*If SUB2 = 0*
e_E2: 40000 cu. ft	a_b: [0, 40, 90, 130, 170, 210, 250]*10^3 cu. ft.	c_d: [0, 10, 20, 40, 60, 80, 90]*10^3 cu. ft.
e_D2: 110000 cu. ft	b_c: [0, 40, 60, 80, 100, 120, 140]*10^3 cu. ft.	d_e: [0, 10, 20, 40, 60, 80, 90]*10^3 cu. ft.
G1_L1: 500 MWh	*If SUB1 = 1*	*If SUB2 = 1*
G1_L2: 400 MWh	a_b: [0, 50, 100, 150, 200, 250, 300]*10^3 cu. ft.	c_d: [0, 20, 30, 50, 70, 90, 100]*10^3 cu. ft.
G2_L2: 400 MWh	b_c: [0, 70, 90, 110, 130, 150, 170]*10^3 cu. ft.	d_e: [0, 20, 30, 50, 70, 90, 100]*10^3 cu. ft.

Figure 1: Top: Interdependent gas (solid lines) and electric (dashed lines) infrastructures and SCADA system (dotted lines) [1]; the quantities demanded by the end-nodes D1, D2, L1, L2 are reported in bold. Bottom: deterministic and stochastic arc capacities.

The gas distribution network, supplied by two sources of gas (namely, S1 and S2, that are connected to the network by arcs S1_DS1 and S2_DS2, respectively), provides gas to the end-nodes D1 and D2 and to two nodes of the electricity network (E1 and E2). Once the gas enters

into nodes E1 and E2, it is transformed into electrical energy that flows through arcs E1_G1 and E2_G2 (representing the electric power generation stations) to supply the end-nodes of electricity (L1 and L2); notice that the demand L2 can be supplied by both electrical generations E1_G1 and E2_G2. The assumption is made that the gas-electricity transformation occurs with a constant coefficient, i.e., 100 cu. ft. of natural gas produces 1 MWh of electricity [1].

A SCADA system controls the gas flow through arcs a_b, b_c, c_d and d_e. It is assumed that: i) the SCADA has two core subsystems controlling different sets of arcs (in particular, the first one – SUB1 – refers to links a_b and b_c, whereas the second one – SUB2 – controls arcs c_d and d_e); ii) the SCADA is always provided with electric power [1].

The capacities of the arcs of the gas and electricity networks (determining the maximum flows of gas or electricity supported by the arcs) can be deterministic (i.e., fixed constant values) or stochastic (i.e., randomly evolving in time) (Figure 1 bottom). The stochastic capacities give rise to a multi-state model that reflects the possibly different degrees of damage of the arcs. On the contrary, the SCADA system state is defined by a binary random variable, whose values 1 and 0 represent its complete and partial functioning, respectively. For example, when the state of the SCADA subsystem SUB1 (controlling arcs a_b and b_c) is 0, the capacity of these arcs decreases because of the incorrect information provided by the SCADA subsystem (even if the arcs are not subject to a direct damage). On the basis of the two states of the SCADA subsystems, two different vectors of capacities are identified for each arc a_b, b_c, c_d and d_e as illustrated in Figure 1 bottom.

In the following, we generically denote the value of the state of a component (i.e., the capacity of the arcs) as $\zeta^{c,i}$, $i \in \{1,2,\ldots,S^c\}$, where c indicates the component of interest and i the state number (when $i = 1$, the component is in the worst state, whereas when $i = S^c$, it is in

the best state); S^c is the total number of states for that component. For example, component S1_DS1 has S^{S1_DS1} = 4 possible states of gas capacity: $\zeta^{S1\_DS1,1}$ = 90000 cu. ft., $\zeta^{S1\_DS1,2}$ = 95000 cu. ft., $\zeta^{S1\_DS1,3}$ = 100000 cu. ft., $\zeta^{S1\_DS1,4}$ = 105000 cu. ft. The total number of components in the SoS is referred to as NC.

Changes in the arc capacities are due to random failures or recovery actions. The state transitions over time are modeled by Markov and semi-Markov processes as in [1]. Semi-Markov processes are adopted to represent the evolution of the capacities of the gas supply links (S1_DS1 and S2_DS2), whereas Markov processes are used for all the others arcs. Both Markov and semi-Markov processes for a generic component *c*, *c* = 1, 2, ..., NC, are defined by a transition probability matrix $\underline{\underline{P}}^c = \{p_{ij}^c : i, j = 1, 2, ..., S^c\}$, where p_{ij}^c is the one-step probability of transition from state *i* to state *j*. In addition, the semi-Markov processes are characterized by a matrix of continuous probability distribution (e.g. Normal), $\underline{\underline{T}}^c = \{th_{ij}^c \approx N(\mu_{ij}^c, \sigma_{ij}^c) : i, j = 1, 2, ..., S^c\}$, for the holding time, i.e., for the time of residence in state *i* before performing a transition to state *j*. The total number of components in the SoS described by the semi-Markov processes is referred to as NS.

Differently from [1], we take into account the epistemic uncertainty affecting the transition probabilities and the holding time distributions of the Markov and semi-Markov processes, respectively. In particular, intervals $[\underline{p}_{ij}^c, \overline{p}_{ij}^c]$, *c* = 1, 2, ..., NC, *i*,*j* = 1, …, S^c, (instead of fixed constant values) are used to describe the state transition probabilities for both Markov and semi-Markov processes (matrices $\underline{\underline{P}}^c$, *c* = S1_DS1, S2_DS2, a_b, b_c, c_d d_e, SCADA, E1_G1 and E2_G2, in Figure 2 with respect to the states defined in Figure 1 bottom) [30-35]. The holding time distributions for the components modeled by the semi-Markov processes are considered normal with epistemically-uncertain mean (described by an interval $[\underline{\mu}_{ij}^c, \overline{\mu}_{ij}^c]$) and

fixed standard deviation, σ_{ij}^{c}, (matrices $\underline{\underline{\mathrm{T}}}^{c}$, c = S1_DS1, S2_DS2, in Figure 2); this level-2 hierarchical representation produces a family of Normal probability distributions characterized by the same standard deviation, but different mean values: such a bundle of distributions is often referred to as distributional probability-box (p-box) [36-41]. Notice that we have considered a single value instead of an interval of values for the standard deviation just to reduce the computational time of the simulation, but this does not represent a limitation of the approach.

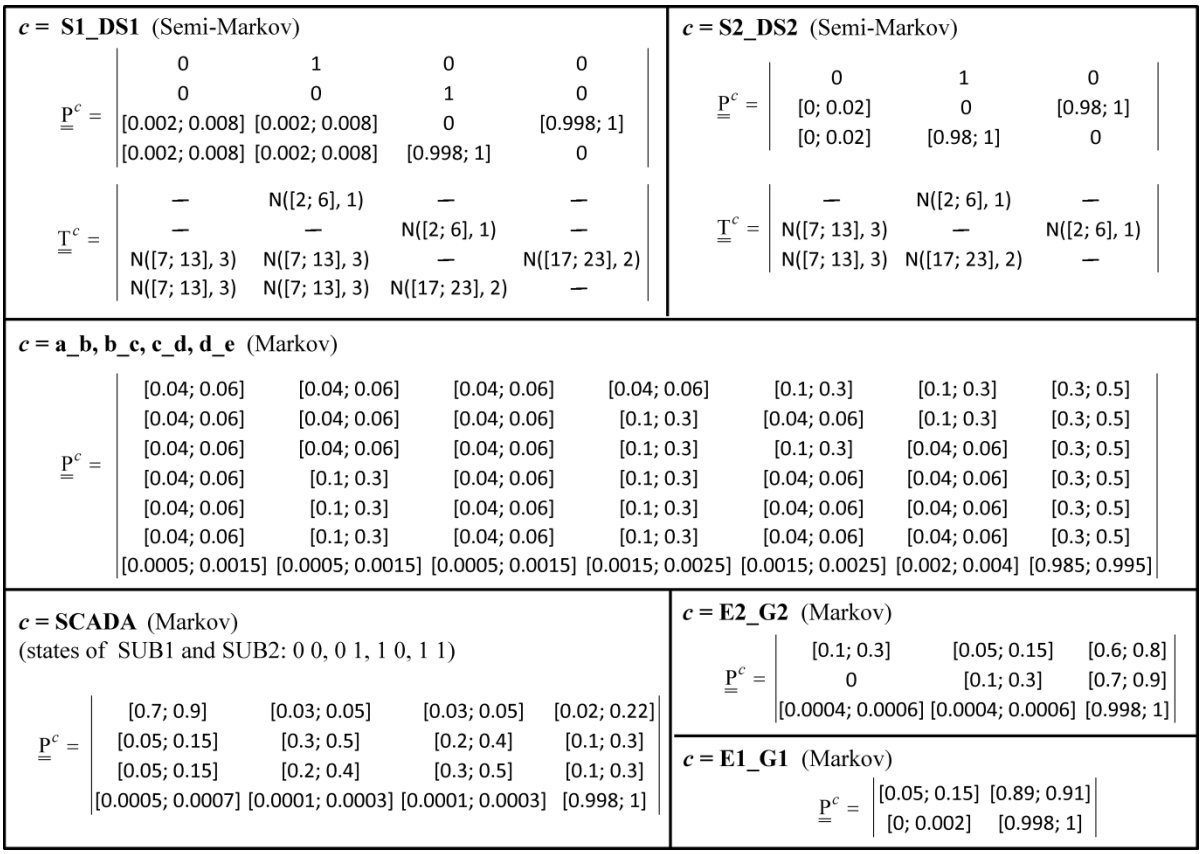


Figure 2: Holding time distributions (matrices $\underline{\underline{\mathrm{T}}}^{c}$) for the arcs described by semi-Markov processes: each element of the matrix represents a Normal distribution with uncertain (interval) mean and fixed standard deviation. State transition probability matrices ($\underline{\underline{\mathrm{P}}}^{c}$) for the arcs described by Markov and semi-Markov processes: each element of the matrix represents an interval for the corresponding transition probability.

In the present work, the demand nodes are not given the same importance: in particular, D1 is more important than L1; on its turn, L1 is more important than both D2 and L2 (which instead

are equally important). These assumptions are made to illustrate and motivate the logical repartition of electricity and gas flows in the network and its representation in the GTST-DMLD given in the next Section 3.

The objectives of the analysis are to determine the cumulative distribution functions of i) the product delivered to the demand nodes (i.e., D1, D2, L1, L2) at the steady state and ii) the time needed to recover the SoS from the worst scenario. Since the state transition probabilities of the network components are affected by epistemic uncertainty and are described by intervals, $[\underline{p}_{ij}^{c}, \overline{p}_{ij}^{c}]$, c = 1, 2, ..., NC, i,j = 1, …, S^c, the corresponding component steady-state probabilities are also affected by epistemic uncertainty and represented by intervals of possible values, $[\Pi_{\min}^{c,i}, \Pi_{\max}^{c,i}]$, c = 1, 2, ..., NC, i = 1, 2, ..., S^c. As a consequence, a set of cumulative distribution functions corresponding to the set of possible steady-state probabilities within the intervals $[\Pi_{\min}^{c,i}, \Pi_{\max}^{c,i}]$, c = 1, 2, ..., NC, i = 1, …, S^c, is obtained for each demand node. For the same reason (i.e., for the presence of the epistemic uncertainty in the state transition probabilities and in the mean of the components holding time distributions) a set of cumulative distribution functions for the recovery time of the system is obtained in correspondence of the set of possible state transition probabilities.

3. SYSTEM-OF-SYSTEMS MODELLING

3.1. Goal Tree Success Tree – Dynamic Master Logic Diagram: basic concepts

The Goal Tree Success Tree – Dynamic Master Logic Diagram (GTST-DMLD) is a goal-oriented method based on a hierarchical framework [12]. It gives a comprehensive description of the systems in terms of functions (qualities), objects (parts) and their relationships (interactions). The first description is provided by the Goal Tree (GT), the second by the

Success Tree (ST) and the third by the DMLD [12].

The GT identifies the hierarchy of the qualities of the system composing the objective of the analysis, i.e., the goal, organizing them in functions that are in turn subdivided into other functions and so on. The hierarchy is built by answering questions on "how" the subfunctions can attain the parent functions (looking at the hierarchy from top to bottom) and on "why" the functions are needed (looking at the hierarchy from bottom to top). Two types of qualities, i.e., main and support functions, are considered: the former directly contribute to achieving the goal, whereas, the latter support the realization of the former [42].

The ST represents the hierarchy of the objects of the system, from the entire system to the parts necessary to attain the last levels of the GT. This hierarchy is built identifying the elements that are "part of" the parent objects. As for the GT, two types of objects are distinguished also in the ST: main and support. The former are directly contributing to achievement of the main functions, whereas the latter are needed for the operation of the former [42].

The DMLD is an extension of the Master Logic Diagram (MLD) [12] introduced to model the dynamic behavior of a physical system. It describes the interactions between parts, functions and parts and functions, in the form of a dependency matrix, and it includes the dynamics by means of time-dependent fuzzy logic rules [12].

A conceptual sketch of GTST-DMLD is given in Figure 3.

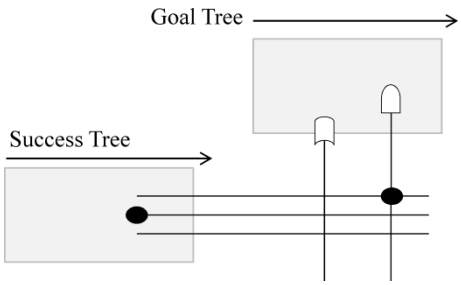


Figure 3: Conceptual sketch of GTST-DMLD: the filled dots indicate the possible dependencies between the objects (filled dot on the left) and between the objects and functions (filled dot on the right), the logic gates indicate how a given function depends on the input values.

The GT is drawn at the top, the ST tree on the left and the DMLD is represented by filled dots at the intersections between vertical and horizontal lines, to indicate the possible dependencies between the elements on the left and on the top. Several types of logic gates can be used to represent the time-dependent fuzzy logic rules, and different dependency-matrix nodes to describe the probabilities and degrees of truth in the relationships [12]. Figure 4 gives an example of dependency of an element C on two elements A and B by the "AND" gate in a DMLD [12]. In this case, the output value of the element C is the minimum value between the inputs A and B. Replacing the "AND" gate with an "OR" gate, the output value will be the maximum between the input values.

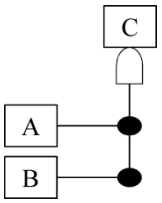


Figure 4: Example of an element C that depends on two elements A and B by an "AND" gate.

Further details on the *construction* of the GTST-DMLD modeling and its *applications* are not given here for brevity sake: the interested reader is referred to the cited literature [12, 42]. In the next Section 3.2, the adaptation of the GTST-DMLD for modeling interconnected networked infrastructures is illustrated.

3.2. Goal Tree Success Tree – Dynamic Master Logic Diagram for interconnected networked infrastructures

In this Section, we adapt the GTST-DMLD presented in Section 3.1, in general terms, for an adequate representation of interconnected networked infrastructures, and in particular of the ones making the SoS of our case study of Section 2. Specifically, we introduce new concepts in order to model in the diagram not only the dependency relations between the components, but also the ways in which the flows of gas and electricity are partitioned into the network on the basis of i) the importance of the demand nodes, ii) the amount of product necessary to satisfy each demand, iii) the constraints of the arc capacities and iv) the information provided by the SCADA system. In the following, first we explain the notation adopted in the GTST-DMLD and, then, we apply it to the case study of interest.

In the present work, we distinguish between three main types of dependency: *direct*, *indirect* and *constraint-based* dependencies, as illustrated in Figures 5 and 6. The first ones, pictorially represented by a dot, express the fact that the product of the element on the bottom passes straight into the element on the top. Indirect dependencies, represented by a hexagon, capture the relations between arcs that share the same input flow, but whose outputs are not related. This type of dependencies is important for the optimal allocation of the product in the network: for example, it is used to describe those cases where the flow exceedance in an arc can be better partitioned into another arc that is not directly connected to it, but that shares one of the inputs (see the example of Figure 5 b). Finally, constraint-based dependencies, depicted by a triangle, are employed to take into account those relations that do not involve an exchange of physical product, but rather a transfer of information which may impact the state of the connected element. Finally, it is worth noting that in the model, we adopt the symbol of triangle also to represent some physical constraints posed by the problem, like the maximum

flow required by a demand node.

It is worth mentioning that since in the present case we are interested in analyzing the flows passing through the network (and not just the dependency relations), the inputs of an arc are flows and the output is (generally) the sum of the flow inputs. For this reason, in this context the "AND" gate assumes a different meaning than that in [12] (see the previous Section 3.1): in particular, the output value is the sum of the input values and it is represented by a "+" in the middle of the gate, as shown in the following examples (Figures 5 and 6). We can then distinguish between the "logical" gates studied by Hu and Modarres [12] and the "physical" gates proposed in the present work: the first ones are needed to highlight the logical connections between the elements that take part/role in a given structure or function of interest; the second ones are used to evaluate the physical flow distribution in the system. Examples of the types of dependencies (direct, indirect and constraint-based) associated to the physical gates are shown in the following.

For clarity of illustration, in Figure 5, examples of two types of direct and indirect dependencies are given, with respect to different graph representations. Notice that nodes are neglected and just the relations between arcs are considered. Figure 5 a. shows the dependence of arc C on two input arcs A and B: arc C receives all the input products from A and B (e.g., if the flows in arcs A and B are 50 and 70 units, respectively, the flow in arc C is 120 units); this complete direct dependence is depicted by a black dot. Figures 5 b. and c. describe the same "physical" situation (i.e., an input arc A and two output arcs B and C), but with different relative importance of the arcs. Two different cases are illustrated. In the first case (Figure 5 b.) arc B is more important than C: thus, in this situation, the flow from A supplies first arc B until its demand is satisfied, and then arc C: e.g., if the flow in arc A is 100 units and both arcs B and C need 80 units, arc B will receive 80 units – demand fully satisfied

– and arc C the rest, i.e., 20 units, – demand partially satisfied. Arc C is dependent on arc B since the flow that can reach C depends on the quantity given to B. In the second case (Figure 5 c.), arcs B and C are equally important: thus, the input flow (A) is divided into equal parts on the basis of the number of output arcs (i.e., two in this example); with respect to the numeric example above, both arcs B and C will receive 50 units – demands partially satisfied. Arcs B and C are reciprocally dependent since the product distributed to one of them depends on that delivered to the other one. The dependency between arcs B and C is "indirect" for both cases since the output of an arc is not the input of the other one and vice versa. In the case of Figure 5 b., the flow that enters in C is given by the difference between the entire flow from A (direct dependency) and the flow given to B (indirect dependency); this concept is illustrated in the GTST-DMLD by the symbol of direct dependency from A to C (dot) and the symbol of indirect dependency from B to C (hexagon). In particular, for the quantitative evaluation of the model, a white hexagon is introduced to reduce the input flow from arc A by the quantity of product given to arc B: in this view, the white hexagon assumes the value of the flow in B with a negative sign. The flow given to B can be the entire flow of A or a lower value depending on the constraints and arc capacity (see the following example in Figure 6). In the case of Figure 5 c., the flow from A is divided into equal parts: this condition is represented by a grey dot. However, this equal partition of the flow may not represent the optimal one, since some output arcs may require less flow than the one allocated according to this criterion, e.g., if the flows in arc A is 100 units and arcs B and C need 80 and 20 units, respectively, giving 50 units to both arcs is not a good allocation of the resource since B is partially satisfied and some product (i.e., 30 units) given to arc C is wasted. Thus, to optimize the repartition of the flow, indirect dependencies are adopted: they are directed from an output arc to all the other output arcs that share the same input. In this case, the "surplus flow" is a positive quantity and it is represented by a grey hexagon (to distinguish it from the "negative"

white hexagon of the example in Figure 5 b).

Notice that the graph representation of Figures 5 b. and 5 c. are identical; however, the partition of the flux from A is completely different in the two cases: this means that the graph representation alone cannot be used to describe the repartition of the flows in the network according to different criteria. On the contrary, the DMLD can capture and represent this aspect, which is useful in the quantitative evaluation of system performance.

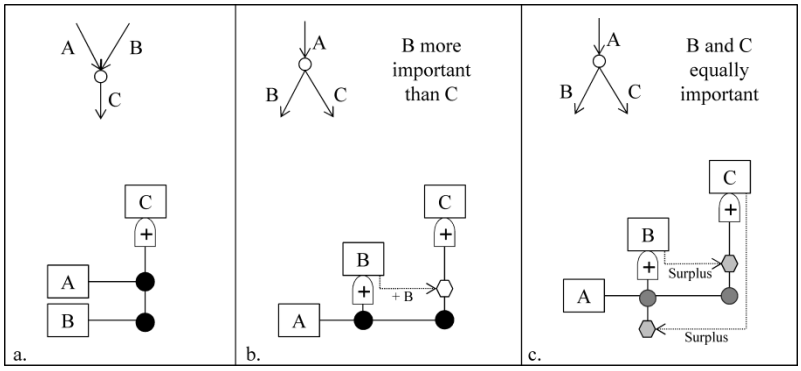


Figure 5: Examples of direct and indirect dependencies with respect to possible graph representations.

In Figure 6, examples of two types of constraint-based dependencies are given, with respect to different possible graph representations. Figure 6 a. depicts the same situation as Figure 5 a., with an additional arc D whose behavior impacts on the state of arc C (however, notice that D is not an input to C). This dependency is represented by a grey triangle and it means that the output of C can be modified on the basis of the state of arc D. In the present case study, this constraint-based dependency is used to model the SCADA system that can decrease the actual flow of the controlled arc if it is in a damage state. Figure 6 b. represents the same situation of Figure 5 c. with the addition of another arc (D) sequential to arc C. In this case, there is not a "real dependency" from arc D to arc C, but we adopt the symbol of constraint-based dependency (triangle) as a partitioning constraint to represent the fact that the capacity (or the demand) of arc D can limit the amount of flow in input to arc C, e.g., if the flows in arc A is 100 units, the capacity of arc C is 50 units and arcs B and D need 80 and 20 units,

respectively, the repartition of the flow is as follows: first 100 units from A are equally divided into arcs B and C (50 units each) and the surplus (if there is) is partitioned into arcs B and C, then the constraint-based dependency is considered (i.e., arc D needs 20 units) and the new surplus is given to arc B (i.e., the exceedance of 30 units from arc C is directed to arc B). This partitioning constraint is represented in the DMLD by a black triangle and it is needed to control the input flow partitioned in different arcs and guarantee that it is not higher than necessary.

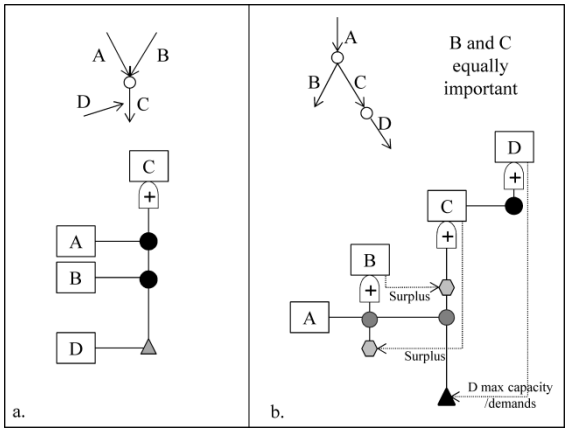


Figure 6: Examples of constraint-based dependencies with respect to possible graph representations.

Finally, another type of constraint is taken into account, i.e., the one related to the capacity of the arcs: when the flow in input to an arc is higher than the capacity of the arc itself, the output flow will be equal to the capacity of the arc. The arc capacity can be deterministic or stochastic and in the GTST-DMLD it is represented by a grey or dot-filled rectangular, respectively (see Figure 7).

In Figure 7, the GTST-DMLD of the case study of Section 2 is shown.

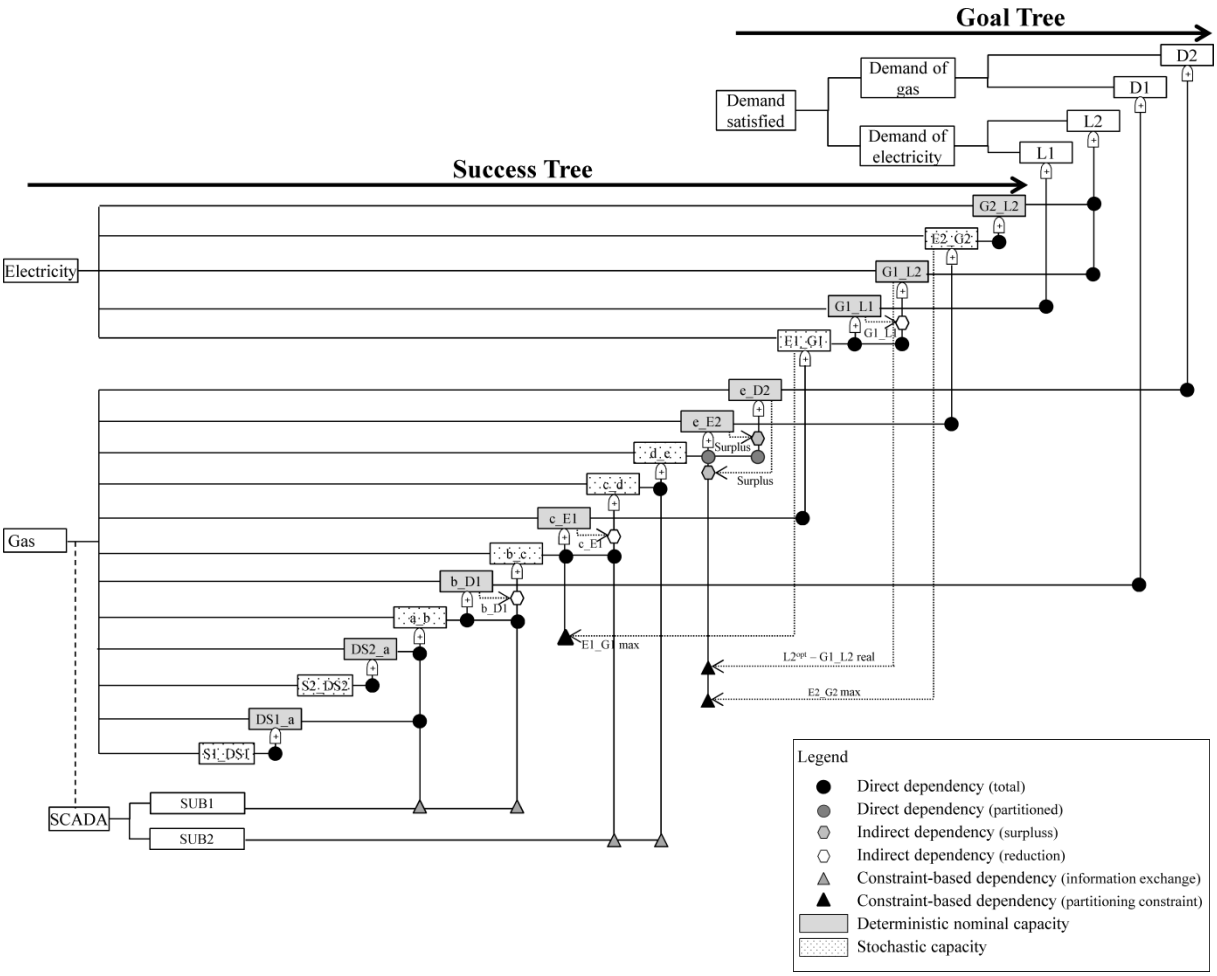


Figure 7: GTST-DMLD of the case study of Section 2 corresponding to the graph of Figure 1.

The GT on the top represents the main goal of the system of systems (SoS), related to the supply of the demands of gas and electricity: the objective is achieved if the corresponding nodes D1, D2, L1 and L2 receive the required amount of gas and electricity, respectively. In the present case study, we limit the analysis to the last level of the GT, i.e., we analyze the performance of each demand, without investigating a global indicator of the SoS.

The ST is composed by the main hierarchies of the gas and electricity networks (that directly provide the demand nodes with gas and electricity to achieve the goal function) and by the support hierarchy of the SCADA system (that is needed for the control of the gas network and, therefore, it is not directly involved in the achievement of the goal function); given its support role, it is represented in a parallel dashed branch connected to the gas hierarchy.

The DMLD is represented by the relationships between objects of the ST or between objects

of the ST and functions of the GT. It allows determining the goal function by the evaluation of all the dependencies from the bottom to the top of the diagram, following the rules explained above for the direct, indirect and constraint-based dependencies. For example, arc a_b depends on two arcs, DS1_a and DS2_b, connected by direct dependencies (Figure 7). Thus, the output of a_b is given by the sum of the corresponding input values, i.e., DS1_a + DS2_b. This value may, then, be modified by the constraint-based dependency of the SCADA system and by the (stochastic) capacity of arc a_b itself.

4. EVALUATION OF THE SYSTEM-OF-SYSTEMS PERFORMANCE

In this Section, we illustrate the evaluation of the performance of the system of systems (SoS), described in Section 2, in the presence of epistemic uncertainties (represented by intervals) affecting the components' state transition probabilities and the mean values of the holding time distributions. As already mentioned in Section 2, the system performance is quantified in terms of i) robustness, measured by the steady-state probability distributions of the product delivered at the demand nodes (see Section 4.1) and ii) recovery capacity, measured by the time needed to recover the SoS from the worst scenario (see Section 4.2). The reader is referred to Appendix A for a brief overview of imprecise (interval) probabilities.

4.1. Robustness

To compute the steady-state probability distributions of the product delivered at the demand nodes the following three main steps are carried out:

1. Processing the epistemic uncertainties by interval analysis: this step leads to the evaluation of the intervals of the steady-state probabilities, $[\Pi_{\min}^{c,i}, \Pi_{\max}^{c,i}]$, $i = 1, 2, ..., S^c$, for the states of each component ($c = 1, 2, ..., NC$) of the SoS.
2. Evaluation of the SoS performance (i.e, robustness) by Monte Carlo simulation: this step

leads to the determination of a set of cumulative distribution functions (CDFs) of the product delivered at each demand node at steady state, one for each possible combination of steady-state probabilities ranging within the intervals $[\Pi_{\min}^{c,i}, \Pi_{\max}^{c,i}]$, i = 1, 2, ..., S^c, (found at step 1. above).

3. Post-processing the results obtained at the previous step 2: this step leads to the identification of two extreme upper and lower CDFs that bound the set of CDFs produced at step 2. above.

In more details:

1. Solve the following optimization problems for the lower (resp., upper) bounds $\Pi_{\min}^{c,i}$ (resp., $\Pi_{\max}^{c,i}$), c = 1, 2, ..., NC, for each row i, i = 1, 2, ..., S^c, of the transition probability matrix $\underline{\underline{P}}^c$ (that is composed by probability intervals $[\underline{p}_{ij}^c, \overline{p}_{ij}^c]$, i,j = 1, 2, ..., S^c):

$$\Pi_{\min}^{c,i} = \min_{p_{ij}^c,\ j=1,2,...,S^c} \{\Pi^{c,i}\},\ \forall i = 1,2,...,S^c,\ c = 1, 2, ..., NC$$
$$\Pi_{\max}^{c,i} = \max_{p_{ij}^c,\ j=1,2,...,S^c} \{\Pi^{c,i}\},\ \forall i = 1,2,...,S^c,\ c = 1, 2, ..., NC \quad (1)$$

such that:

$$p_{ij}^c \in [\underline{p}_{ij}^c, \overline{p}_{ij}^c] \quad (2)$$

$$\sum_{j=1}^{S^c} p_{ij}^c = 1 \quad (3)$$

$$\underline{\underline{\Pi}}^c = \underline{\underline{\Pi}}^c \cdot \underline{\underline{P}}^c \quad (4)$$

The constraint of eq. (2) means that the transition probability from state i to state j is not known precisely and can take values in the interval $[\underline{p}_{ij}^c, \overline{p}_{ij}^c]$ [27]; the constraint of eq. (3) refers to a fundamental property of Markov and semi-Markov processes, i.e., the states for each component are exhaustive [43]; finally, eq. (4) reports the definition of steady-state

probability for a Markov process [43]. Notice that the sum of the elements of the vector $\underline{\Pi}^c$ is equal to 1. In the case of a semi-Markov process, the output of eq. (4), i.e., $\underline{\Pi}^c$, is weighted by the expected time of residence, τ^i, in a given state, *i*, before performing a transition [44]: $\xi^{c,i} = \Pi^{c,i} \cdot \tau^i / \sum_{j=1}^{S^c} \Pi^{c,j} \cdot \tau^j$ for i = 1, …, S^c. Notice that the optimization problems (1) can be solved by performing an exhaustive greedy search within the probability intervals $[\underline{p}_{ij}^c, \overline{p}_{ij}^c]$, if the dimensions of the corresponding transition probability matrices are relatively small (e.g., below 4 x 4), otherwise, alternative intelligent techniques should be sought, e.g., meta-heuristic methods like Genetic Algorithms (GAs) [27]. In this work, we resort to GAs for arcs a_b, b_c, c_d, d_e (whose transition probability matrices are 7 x 7), whereas we perform an exhaustive search for all the other arcs. In Appendix B, the operative steps to obtain the lower and upper bounds of the steady state probabilities (i.e., $[\underline{\Pi}_{\min}^c, \underline{\Pi}_{\max}^c]$) by performing an exhaustive search are detailed and the need to resort to alternative intelligent techniques when the dimension of the transition probability matrix increases is discussed.

2. Identify the CDFs of the product delivered at each demand node at steady state for all the possible combinations of components steady-state probabilities found at step 1. above:
   a. For each component *c*, c = 1, 2, ..., NC, let the steady-state probabilities, $\Pi^{c,i}$, i = 1, 2, ..., S^c, range within the corresponding interval $[\Pi_{\min}^{c,i}, \Pi_{\max}^{c,i}]$, i = 1, 2, ..., S^c, to obtain a set of Q^c vectors of steady-state probabilities, $\{\underline{\Pi}^{c,1}, \underline{\Pi}^{c,2}, ..., \underline{\Pi}^{c,q}, ..., \underline{\Pi}^{c,Q^c}\} : q = 1, ..., Q^c\}$, such that $\sum_{i=1}^{S^c} \Pi^{c,q,i} = 1$, q = 1, …, Q^c. Notice that this gives rise to $Q^1 * Q^2 * \ldots * Q^{NC} = N_{tot}$ possible combinations of steady-state probability vectors of the system components, i.e., to N_{tot} steady-state probability vectors for the entire system.

b. For all the NC components, select one steady-state probability vector among the set $\underline{\Pi}^{c,q}$, c = 1, 2, ..., NC, $q \in \{1, ..., Q^c\}$ (generated at step a. above); in other words, this amounts to selecting one of the $Q^1 * Q^2 * \ldots * Q^{NC} = N_{tot}$ steady-state probability vectors for the entire SoS.

c. Fixing the SoS steady-state probability vector selected in b., randomly sample the states $\zeta^{c,i}$ (i.e., the capacities), $i \in \{1, \ldots, S^c\}$, of all the components of the system (i.e., arcs). Then, compute the product delivered at the demand nodes propagating the flow in each component of the SoS through the GTST-DMLD (see Section 3.2).

d. Repeat step c. a large number of times (e.g., 1000 in this work) and obtain the CDF for the product delivered at each demand node.

e. Repeat steps c.-d. for another combination of the steady-state probability vectors, $\underline{\Pi}^{c,q}$, c = 1, 2, ..., NC, $q \in \{1, ..., Q^c\}$, of all the NC components, until all the N_{tot} possible combinations of the steady-state probability vectors of the SoS are explored.

At the end of steps a.-e., an ensemble of CDFs for each demand nodes is obtained, one for each of the N_{tot} possible combinations of steady-state probabilities of the entire SoS.

3. Identify the extreme minimum and maximum CDFs (i.e., the enveloping p-box of the CDFs) of the product delivered at the demand nodes that bound the set of CDFs produced at step 2. above.

4.2. Recovery time

The time needed to recover the SoS from the worst scenario (i.e., the one characterized by components in the worst state) to a level in which all the demand nodes are satisfied, is carried out by three main steps:

1. Processing the epistemic uncertainties by interval analysis: this step leads to the

identification of K^c transition probability matrices $\underline{\underline{\mathcal{P}}}^{c,k}$, c = 1, 2, ..., NC, k = 1, 2, ..., K^c, composed by single values; in addition, for the NS components described by semi-Markov process, this step leads to the identification of H^c matrices $\underline{\underline{\underline{Mu}}}^{c,h}$, $c \in \{1, 2, ..., NC\}$, $h \in \{1, 2, ..., H^c\}$, composed by single values of the mean of the holding time distributions.

2. Evaluation of the SoS performance (i.e., recovery capacity) by Monte Carlo simulation: this step leads to the determination of a set of cumulative distribution functions (CDFs) of the time needed to recover the SoS, one for each possible combination of state probability matrices sampled.
3. Post-processing the results obtained at the previous step 2: this step leads to the identification of two extreme upper and lower CDFs that bound the set of CDFs produced at step 2. above.

In more details, step 1. is described in Appendix B (steps B1.-B3.); step 2 instead is performed as follows:

a. Randomly select NC matrices $\underline{\underline{\mathcal{P}}}^{c,k}$, c = 1, 2, ..., NC, $k \in \{1, 2, ..., K^c\}$, for all the NC components of the SoS and NS matrices $\underline{\underline{\underline{Mu}}}^{c,h}$, $h \in \{1, 2, ..., H^c\}$, for the NS components c described by a semi-Markov process.
b. Set u = 1 (counter of the number of simulations).
c. Initialize the state of the components at the worst state ($\zeta^{c,i}$, i = 1, c = 1, 2, ..., NC): in this state configuration of the SoS, the product delivered to the demand nodes is lower than the optimum required.
d. Initialize the following time variables:

- system simulation time $t = 0$, starting time of the simulation: this variable represents the current simulation time and is needed to compute the recovery time of the SoS;
- components' state transition time $ts^c = \Delta t$, $c = 1, 2, ..., NC$, where Δt is the time step of the simulation ($\Delta t = 1$ in arbitrary units, in this work): these time variables (ts^c, $c = 1, 2, ..., NC$) are needed to determine if the component c can perform a state transition at a given time step t, as illustrated in the next step e.; they are set to 1 since at this time step all the components perform the first state transition.

e. Set $t = t + \Delta t$: if $t = ts^c$, then the component c, $c \in \{1, ..., NC\}$, performs a state transition: then, randomly sample its new state from the matrix $\underline{\underline{\mathcal{P}}}^{c,k}$ ($k \in \{1, ..., K^c\}$) selected at step a. and update the variable ts^c as follows:
   - ✓ If c is described by a Markov process, $ts^c = ts^c + \Delta t$, since a state transition occurs at each time step.
   - ✓ If c is described by a semi-Markov process, $ts^c = ts^c + t^*$, where t^* is the time of next transition that is sampled from the corresponding holding time distribution with mean value taken from the matrix $\underline{\underline{\underline{Mu}}}^{c,h}$, $h \in \{1, 2, ..., H^c\}$, selected at the previous step a. The sampled value t^* is rounded to the nearest integer except when it is zero; in this case, the value is rounded to 1.

   Check $t = ts^c$ for all the components c, $c = 1, 2, ..., NC$.

f. Evaluate the product delivered to the demand nodes at time t by adopting the GTST-DMLD (see Section 3.2), taking into account the state transition of the components in the previous step e.

g. Repeat steps e.-f. until the product delivered to the demand nodes is equal to, or higher than, the optimum required: the corresponding value of recovery time (tr^u) is then

recorded for the simulation u.

h. Set $u = u + 1$ and repeat steps c.-g. a large number of times (e.g., 1000 in this work).

i. A cumulative distribution function of the recovery time of the SoS is identified for a combination of state probability matrices $\underline{\underline{\mathcal{P}}}^{c,k}$, c = 1, 2, ..., NC, $k \in \{1, 2, ..., K^c\}$, selected at step a.

j. Repeat the entire procedure (steps a.-i.) a large number of times (e.g., 10000 in this work) to explore many different combinations of probability matrices $\underline{\underline{\mathcal{P}}}^{c,k}$, c = 1, 2, ..., NC, $k \in \{1, 2, ..., K^c\}$.

At the end of the procedure, a set of cumulative distribution functions of the recovery time of the performance of the SoS is obtained.

The results are processed at step 3., where the minimum and maximum CDFs (i.e., the enveloping p-box of the CDFs) of the recovery time that bound the set of CDFs obtained at step 2. above are identified and the 99th percentiles of the distributions are computed as a measure of the recovery time.

5. RESULTS

Figure 8 shows the lower (dotted line) and upper (solid line) cumulative distribution functions of the gas and the electricity delivered at steady state to the demand nodes D1, D2 and L1, L2, respectively, in steady state, obtained by the procedure illustrated in Section 4.1. Table 1 reports the corresponding (upper and lower) probabilities that the product delivered to the demand nodes, D1, D2, L1 and L2, exceeds the following threshold values: d_1^* = 95000 cu. ft., d_2^* = 75000 cu. ft., l_1^* = 475 MWh and l_2^* = 375 MWh (i.e., the probabilities that the corresponding demands are satisfied).

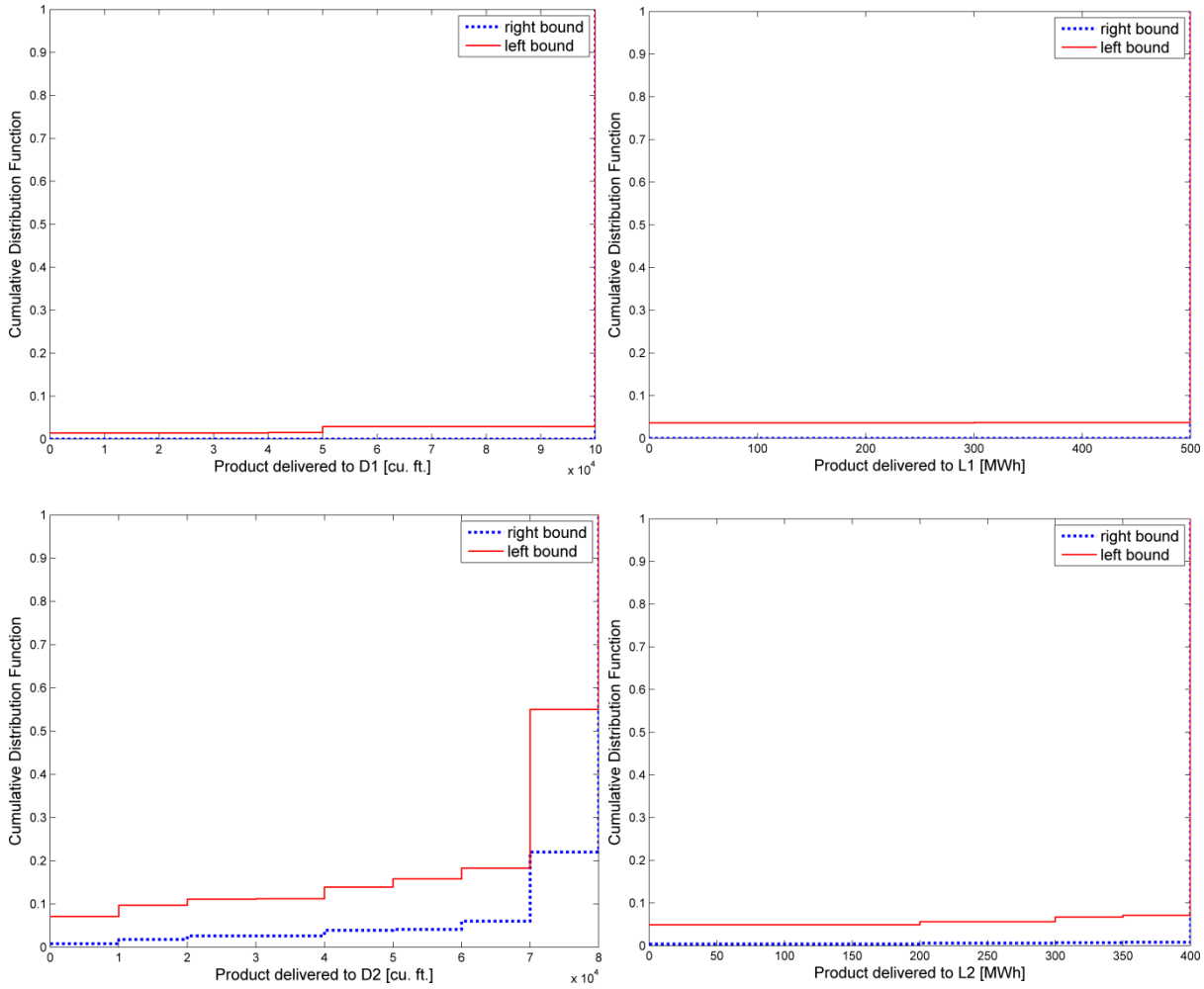


Figure 8: Right (dotted line) and left (solid line) cumulative distribution functions of the product delivered to the nodes D1, D2, L1 and L2 at steady state.

Table 1: Upper and lower probabilities that the product delivered to the demand nodes (D1, D2, L1 and L2) exceeds the corresponding requested threshold value.

$D1 \geq d_1^*$ = 95000 cu. ft. [lower, upper]	$D2 \geq d_2^*$ = 75000 cu. ft. [lower, upper]	$L1 \geq l_1^*$ = 475 MWh [lower, upper]	$L2 \geq l_2^*$ = 375 MWh [lower, upper]
[0.971, 1]	[0.450, 0.780]	[0.963, 1]	[0.929, 0.992]

It can be seen that in general the probability of satisfying demand nodes D1 and L1 is higher than for nodes D2 and L2: their threshold values are satisfied, in the worst case, with probability equal to 0.971 and 0.963, respectively. On the other hand, node D2 is the least supplied: the upper and lower probabilities that the product delivered to it exceeds the corresponding threshold value are low, i.e., 0.450 and 0.780, respectively. This is due to the fact that node D2 can be satisfied by only one path that presents high epistemic uncertainty in

the arc capacities (a_b, b_c, c_d and d_e). On the other hand, node L2 is satisfied with probability between 0.929 and 0.992 even if it is the farthest node from the input sources (and, thus, more affected by uncertainty due to the uncertainties in the arc capacities): this is due to the presence of two redundant paths that allow its supply by arcs E1_G1 and E2_G2.

Figure 9 illustrates the lower (dotted line) and upper (solid line) cumulative distribution functions of the time needed to restore the SoS to a level in which all the demand nodes are satisfied, starting from the worst scenario.

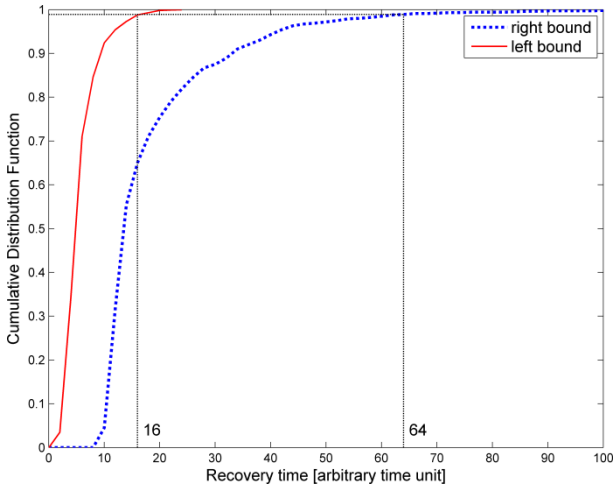


Figure 9: Right (dotted line) and left (solid line) cumulative distribution functions of the recovery time of the supply of the demand nodes, starting from the worst scenario.

The gap between the CDFs reflects the epistemic uncertainty in the transition probability values. In the Figure, the 99th percentile of the CDFs is also reported as a measure of the recovery time.

6. CONCLUSIONS

In this paper, we have introduced a system-of-systems (SoS) framework for the analysis of the robustness and recovery of critical infrastructures (CIs). The analysis by such framework builds on the construction of a GTST-DMLD for system modeling and Monte Carlo

simulation for the quantitative evaluation of the system performance at steady state. The development of the framework in practice has been shown considering the same example created by [1] consisting of two interdependent infrastructures, gas and electric power networks, and a SCADA system connected to the gas network.

In the original framework of [1], the analysis of the robustness and recovery capacity of CIs has been performed by adopting network flow algorithms combined with stochastic processes. The adoption of the GTST-DMLD modeling framework makes the analysis of the robustness and recovery capacity of CIs accessible to a different audience than the original work by [1]. Actually, there is a community of analysts who are much more comfortable using concepts inherent in the GTST-DMLD framework than using methods based on network flow algorithms and stochastic processes. The model put forth by [1] was based on the analysis methods of Operations Research, whereas the GTST-DMLD framework has its roots in the reliability and risk analysis of nuclear power plants and complex electro-mechanical systems.

The framework here developed has shown the capability of representing, modeling and quantitatively accounting for i) the dependencies and interdependencies among the components of a critical infrastructure and between different CIs, respectively, ii) the variability in the states of the components (by adopting a multi-state model), and iii) the epistemic uncertainty in the transition probabilities between different components states (by interval analysis).

The results and insights obtained can help to improve the global SoS performance by improving the structural response of specific arcs that more easily turn into damage states or by developing a more redundant network that allows the supply of the product from different paths.

Acknowledgments

The authors thank the anonymous referees for their critical comments that have helped improve the paper.

REFERENCES

[1] Nozick, L. K., Turnquist, M. A., Jones, D. A., Davis, J. R., and Lawton, C. R., 2005, "Assessing the Performance of Interdependent Infrastructures and Optimising Investments " International Journal of Critical Infrastructures, 1(2-3), pp. 144-154.

[2] Adachi, T., and Ellingwood, B. R., 2008, "Serviceability of Earthquake-Damaged Water Systems: Effects of Electrical Power Availability and Power Backup Systems on System Vulnerability," Reliability Engineering & System Safety, 93(1), pp. 78-88.

[3] Ferrario, E., and Zio, E., 2014, "Goal Tree Success Tree-Dynamic Master Logic Diagram and Monte Carlo Simulation for the Safety and Resilience Assessment of a Multistate System of Systems," Engineering Structures, 59, pp. 411-433.

[4] Apostolakis, G., 1990, "The Concept of Probability in Safety Assessments of Technological Systems," Science, 250(4986), pp. 1359-1364.

[5] NASA, 2010, "Risk-Informed Decision Making Handbook," Technical Report No. NASA/SP-2010-576 - Version 1.0.

[6] US NRC, 2009, "Guidance on the Treatment of Uncertainties Associated with Pras in Risk-Informed Decision Making" Technical Report No. NUREG-1855. US Nuclear Regulatory Commission, Washington, DC.

[7] Sallak, M., Schon, W., and Aguirre, F., 2013, "Reliability Assessment for Multi-State Systems under Uncertainties Based on the Dempster-Shafer Theory," IIE Transactions, 45(9), pp. 995-1007.

[8] Aven, T., and Zio, E., 2011, "Some Considerations on the Treatment of Uncertainties in Risk Assessment for Practical Decision Making," Reliability Engineering & System Safety, 96(1), pp. 64-74.

[9] Bernardo, J. M., and Smith, A. F. M., 1994, *Bayesian Theory*, Wiley, Chichester.

[10] Coolen, F. P. A., and Utkin, L. V., 2007, "Imprecise Probability: A Concise Overview," T. Aven and J. E. Vinnem, eds., London: Taylor & Francis, Stavanger, Norway, pp. 1959-66.

[11] De Finetti, B., 1974, *Theory of Probability*, Wiley, New York.

[12] Hu, Y. S., and Modarres, M., 1999, "Evaluating System Behavior through Dynamic Master Logic Diagram (DMLD) Modeling," Reliability Engineering & System Safety, 64(2), pp. 241-269.

[13] Kozine, I. O., and Utkin, L. V., 2002, "Processing Unreliable Judgements with an Imprecise Hierarchical Model," Risk, Decision and Policy, 7(03), pp. 325-339.

[14] Kuznetsov, V. P., 1991, *Interval Statistical Models (in Russian)*, Radio i Svyaz, Moscow.

[15] Walley, P., 1991, *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall, New York.

[16] Beer, M., and Ferson, S., 2013, "Special Issue of Mechanical Systems and Signal Processing "Imprecise Probabilities - What Can They Add to Engineering Analyses?"" Mechanical Systems and Signal Processing, 37(1-2), pp. 1-3.

[17] Beer, M., Ferson, S., and Kreinovich, V., 2013, "Imprecise Probabilities in Engineering Analyses," Mechanical Systems and Signal Processing, 37(1-2), pp. 4-29.

[18] Blockley, D., 2013, "Analysing Uncertainties: Towards Comparing Bayesian and Interval Probabilities'," Mechanical Systems and Signal Processing, 37(1-2), pp. 30-42.

[19] Crespo, L. G., Kenny, S. P., and Giesy, D. P., 2013, "Reliability Analysis of Polynomial Systems Subject to P-Box Uncertainties," Mechanical Systems and Signal Processing, 37(1-2), pp. 121-136.

[20] Jalal-Kamali, A., and Kreinovich, V., 2013, "Estimating Correlation under Interval Uncertainty," Mechanical Systems and Signal Processing, 37(1-2), pp. 43-53.

[21] Mehl, C. H., 2013, "P-Boxes for Cost Uncertainty Analysis," Mechanical Systems and Signal Processing, 37(1-2), pp. 253-263.

[22] Ferson, S., and Ginzburg, L. R., 1996, "Different Methods Are Needed to Propagate Ignorance and Variability," Reliability Engineering & System Safety, 54(2-3), pp. 133-144.

[23] Ferson, S., and Hajagos, J. G., 2004, "Arithmetic with Uncertain Numbers: Rigorous and (Often) Best Possible Answers," Reliability Engineering & System Safety, 85(1-3), pp. 135-152.

[24] Ferson, S., Kreinovich, V., Hajagos, J., Oberkampf, W., and Ginzburg, L., 2007, "Experimental Uncertainty Estimation and Statistics for Data Having Interval Uncertainty." Sandia National Laboratories, SAND2007-0939, Setauket, New York 11733.

[25] Ferson, S., Moore, D. R. J., Van Den Brink, P. J., Estes, T. L., Gallagher, K., Connor, R. O., and Verdonck, F., 2010, "Bounding Uncertainty Analyses." Application of Uncertainty Analysis to Ecological Risks of Pesticides, W. J. Warren-Hicks and A. Hart, eds., CRC Press, 89-122.

[26] Ferson, S., and Tucker, W. T., 2006, "Sensitivity in Risk Analyses with Uncertain Numbers," Sandia National Laboratories, SAND2006-2801, Setauket, New York.

[27] Buckley, J.J., 2004, "Fuzzy Markov Chains." Fuzzy Probabilities and Fuzzy Sets for Web Planning, Springer, Berlin, 35-43.

[28] Kalos, M. H., and Whitlock, P. A., 1986, *Monte Carlo Methods. Volume: Basics*, Wiley, New York, NY.

[29] Zio, E., 2013, *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*, *Springer Series in Reliability Engineering*, Springer, London.

[30] MSSP, 2013, "Special Issue of Mechanical Systems and Signal Processing "Imprecise Probabilities-What Can They Add to Engineering Analyses?"", Mechanical Systems and Signal Processing, 37(1-2), pp. 1-263.

[31] Muscolino, G., and Sofi, A., 2013, "Bounds for the Stationary Stochastic Response of Truss Structures with Uncertain-but-Bounded Parameters," Mechanical Systems and Signal Processing, 37(1-2), pp. 163-181.

[32] Pannier, S., Waurick, M., Graf, W., and Kaliske, M., 2013, "Solutions to Problems with Imprecise Data"an Engineering Perspective to Generalized Uncertainty Models," Mechanical Systems and Signal Processing, 37(1-2), pp. 105-120.

[33] Reid, S. G., 2013, "Probabilistic Confidence for Decisions Based on Uncertain Reliability Estimates," Mechanical Systems and Signal Processing, 37(1-2), pp. 229-239.

[34] Sankararaman, S., and Mahadevan, S., 2013, "Distribution Type Uncertainty Due to Sparse and Imprecise Data," Mechanical Systems and Signal Processing, 37(1-2), pp. 182-198.

[35] Zhang, H., Dai, H., Beer, M., and Wang, W., 2013, "Structural Reliability Analysis on the Basis of Small Samples: An Interval Quasi-Monte Carlo Method," Mechanical Systems and Signal Processing, 37(1-2), pp. 137-151.

[36] Ferson, S., 2005, "Bayesian Methods in Risk Assessment," Applied Biomathematics, Setauket, New York, www. ramas.com/bayes.pdf.

[37] Karanki, D. R., Kushwaha, H. S., Verma, A. K., and Ajit, S., 2009, "Uncertainty Analysis Based on Probability Bounds (P-Box) Approach in Probabilistic Safety Assessment," Risk Analysis, 29(5), pp. 662-675.

[38] Limbourg, P., and De Rocquigny, E., 2010, "Uncertainty Analysis Using Evidence Theory - Confronting Level-1 and Level-2 Approaches with Data Availability and Computational Constraints," Reliability Engineering & System Safety, 95(5), pp. 550-564.

[39] Möller, B., Graf, W., and Beer, M., 2003, "Safety Assessment of Structures in View of Fuzzy Randomness," Computers & Structures, 81(15), pp. 1567-1582.

[40] Pedroni, N., and Zio, E., 2012, "Empirical Comparison of Methods for the Hierarchical Propagation of Hybrid Uncertainty in Risk Assessment, in Presence of Dependences," International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, 20(4), pp. 509-557.

[41] Pedroni, N., Zio, E., Ferrario, E., Pasanisi, A., and Couplet, M., 2013, "Hierarchical Propagation of Probabilistic and Non-Probabilistic Uncertainty in the Parameters of a Risk Model," Computers & Structures, 126, pp. 199-213.

[42] Brissaud, F., Barros, A., Bérenguer, C., and Charpentier, D., 2011, "Reliability Analysis for New Technology-Based Transmitters," Reliability Engineering & System Safety, 96(2), pp. 299-313.

[43] Zio, E., 2009, *Computational Methods for Reliability and Risk Analysis*, *Series on Quality, Reliability and Engineering Statistics*, World Scientific Publishing Co. Pte. Ltd., Singapore.

[44] Barry, L. N., 1995, *Stochastic Modeling: Analysis and Simulation*, McGraw-Hill, New York.

[45] Williams, P. M., 1976, *Indeterminate probabilities*, Formal Methods in the Methodology of Empirical Sciences, M. Przełęcki, K. Szaniawski, R. Wójcicki, and G. Malinowski, eds., Reidel, Dordrecht, Holland, 229-246.

[46] Lindley, D. V., 2006, *Understanding Uncertainty*, Wiley, Hoboken, NJ.

APPENDIX A: IMPRECISE (INTERVAL) PROBABILITIES

To understand the meaning of imprecise probabilities (or interval probabilities) consider an event *A*. Uncertainty about whether it occurs is represented by a lower probability $\underline{P}(A)$ and an upper probability $\overline{P}(A)$, giving rise to a probability interval $[\underline{P}(A), \overline{P}(A)]$, where $0 \leq \underline{P}(A) \leq \overline{P}(A) \leq 1$. The difference $\Delta P(A) = \overline{P}(A) - \underline{P}(A)$ is called the *imprecision* in the representation of the event *A*. Single-valued probabilities are a special case of no imprecision and the lower and upper probabilities coincide.

Peter M. Williams [45] developed a mathematical framework for imprecise probabilities, based on de Finetti's betting interpretation of probability [11]. This foundation was further developed independently by Vladimir P. Kuznetsov and Peter Walley (the former only published in Russian), see [14] and [15]. Following de Finetti's betting interpretation, the lower probability is interpreted as the maximum price for which one would be willing to buy a bet which pays 1 if *A* occurs and 0 if not, and the upper probability as the minimum price for which one would be willing to sell the same bet. If the upper and lower values are equal, the interval is reduced to a precise probability. These references, and [15] in particular, provide an in-depth analysis of imprecise probabilities and their interpretations, with a link to applications to probabilistic reasoning, statistical inference and decisions.

It is however also possible to interpret the lower and upper probabilities using the reference to a standard interpretation of a subjective probability $P(A)$: such an interpretation is indicated by [46, p. 36]. Consider the subjective probability $P(A)$ and say that the analyst states that his/her assigned degree of belief is greater than the urn chance of 0.10 (the degree of belief of drawing one particular ball from an urn which include 10 balls) and less than the urn chance of 0.5. The analyst is not willing to make any further judgement. Then, the interval [0.10, 0.50] can be considered an imprecision interval for the probability $P(A)$.

Of course, even if the assessor assigns a probability $P(A) = 0.3$, one may interpret this probability as having an imprecision interval [0.25, 0.34] (as a number in this interval is equal to 0.3 when displaying one digit only), interpreted analogously to the [0.1, 0.5] interval. Hence imprecision is always an issue in a practical uncertainty analysis context. This imprecision is commonly viewed as a result of measurement problems. The reference to the urn lottery provides a norm to which assessors should aspire, but measurement problems may make the assessor unable to behave according to it. See also discussion in [9, p. 32].

However, other researcher and analysts have a more positive view on the need for such intervals, see discussions in [8, 22-26]: imprecision intervals are required to reflect phenomena as discussed above, for example when experts are not willing to express their knowledge more precisely than by using probability intervals.

Imprecise probabilities are also linked to the relative frequency interpretation of probability [10]. The simplest case reflects that the "true" frequentist probability p is in the interval $[\underline{P}(A), \overline{P}(A)]$ with certainty. More generally and in line with the above interpretations of imprecision intervals based on subjective probabilities $P(\cdot)$, a two-level uncertainty characterization can be formulated (see, e.g., [13]): $[\underline{P}(A), \overline{P}(A)]$ is an imprecision interval for the subjective probability $P(a \leq p \leq b)$ where a and b are constants. In the special case that $\underline{P}(A) = \overline{P}(A)$ (= q, say) we are led to the special case of a $q \cdot 100\%$ credibility interval for p (i.e., with subjective probability q, the true value of p is in the interval $[a, b]$). For further details, the reader is referred to the recent Special Issue on imprecise probabilities appeared on the Journal of Mechanical Systems and Signal Processing [30].

APPENDIX B: PROCESSING EPISTEMIC UNCERTAINTY BY INTERVAL ANALYSIS: DETAILED OPERATIVE STEPS

The operative steps carried out to process the epistemic uncertainty by interval analysis, needed for the robustness and recovery analyses of Sections 4.1 and 4.2, are illustrated in what follows.

To recall the notation, the algorithm requires the following inputs:

- A state transition probability matrix $\underline{\underline{P}}^c$, c = 1, ..., NC, composed by probability intervals $\underline{\underline{P}}^c = \{[\underline{p}_{ij}^c, \overline{p}_{ij}^c] : c = 1, ..., \mathrm{NC}, i, j = 1, ..., \mathrm{S}^c\}$ for all the NC components c of the system, where i,j are indices representing the state of the component c and S^c is the total number of states of component c. The state transition probability matrix $\underline{\underline{P}}^c$ assumes this form:

$$\underline{\underline{P}}^c = \begin{array}{c|cccc} i/j & 1 & 2 & \dots & \mathrm{S}^c \\ 1 & [\underline{p}_{11}^c, \overline{p}_{11}^c] & [\underline{p}_{12}^c, \overline{p}_{12}^c] & \dots & [\underline{p}_{1\mathrm{S}^c}^c, \overline{p}_{1\mathrm{S}^c}^c] \\ 2 & [\underline{p}_{21}^c, \overline{p}_{21}^c] & [\underline{p}_{22}^c, \overline{p}_{22}^c] & \dots & [\underline{p}_{2\mathrm{S}^c}^c, \overline{p}_{2\mathrm{S}^c}^c] \\ \dots & \dots & \dots & \dots & \dots \\ \mathrm{S}^c & [\underline{p}_{\mathrm{S}^c 1}^c, \overline{p}_{\mathrm{S}^c 1}^c] & [\underline{p}_{\mathrm{S}^c 2}^c, \overline{p}_{\mathrm{S}^c 2}^c] & \dots & [\underline{p}_{\mathrm{S}^c\mathrm{S}^c}^c, \overline{p}_{\mathrm{S}^c\mathrm{S}^c}^c] \end{array}$$

- A holding time distribution matrix $\underline{\underline{T}}^c$, $c \in \{1, 2, ..., \mathrm{NC}\}$, for the NS components described by a semi-Markov process with epistemically uncertain mean μ_{ij}^c represented by an interval of values, $\underline{\underline{T}}^c = \{th_{ij}^c \approx N(\mu_{ij}^c, \sigma_{ij}^c) : \mu_{ij}^c \in [\underline{\mu}_{ij}^c, \overline{\mu}_{ij}^c], i, j = 1, ..., \mathrm{S}^c\}$:

$$\underline{\underline{T}}^c = \begin{array}{c|cccc} i/j & 1 & 2 & \dots & \mathrm{S}^c \\ 1 & \mathrm{N}([\underline{\mu}_{11}^c, \overline{\mu}_{11}^c], \sigma_{11}^c) & \mathrm{N}([\underline{\mu}_{12}^c, \overline{\mu}_{12}^c], \sigma_{12}^c) & \dots & \mathrm{N}([\underline{\mu}_{1\mathrm{S}^c}^c, \overline{\mu}_{1\mathrm{S}^c}^c], \sigma_{1\mathrm{S}^c}^c) \\ 2 & \mathrm{N}([\underline{\mu}_{21}^c, \overline{\mu}_{21}^c], \sigma_{21}^c) & \mathrm{N}([\underline{\mu}_{22}^c, \overline{\mu}_{22}^c], \sigma_{22}^c) & \dots & \mathrm{N}([\underline{\mu}_{2\mathrm{S}^c}^c, \overline{\mu}_{2\mathrm{S}^c}^c], \sigma_{1\mathrm{S}^c}^c) \\ \dots & \dots & \dots & \dots & \dots \\ \mathrm{S}^c & \mathrm{N}([\underline{\mu}_{\mathrm{S}^c 1}^c, \overline{\mu}_{\mathrm{S}^c 1}^c], \sigma_{\mathrm{S}^c 1}^c) & \mathrm{N}([\underline{\mu}_{\mathrm{S}^c 2}^c, \overline{\mu}_{\mathrm{S}^c 2}^c], \sigma_{\mathrm{S}^c 2}^c) & \dots & \mathrm{N}([\underline{\mu}_{\mathrm{S}^c\mathrm{S}^c}^c, \overline{\mu}_{\mathrm{S}^c\mathrm{S}^c}^c], \sigma_{\mathrm{S}^c\mathrm{S}^c}^c) \end{array}$$

By way of example and for clarity of illustration, in the following we refer to component c = S2_DS2 of Figure 1, whose transition probability matrix $\underline{\underline{P}}^c$ and holding time distributions $\underline{\underline{T}}^c$ are reported in Figure 2.

The algorithm proceeds as follows:

B1. Select a component c, $c \in \{1, 2, ..., NC\}$, and a row i, $i \in \{1, 2, ..., S^c\}$, of matrix $\underline{\underline{P}}^c$ whose dimension is S^c x S^c (see Figure B1, left): for component c = S2_DS2, $\underline{\underline{P}}^c$ has dimension 3 x 3. Letting the probabilities p_{ij}^c, j = 1, 2, ..., S^c, vary within the corresponding intervals $[\underline{p}_{ij}^c, \overline{p}_{ij}^c]$, identify all the possible combinations of the probability values in row i (Figure B1, middle, with reference to row i = 2). Given the assumption that the component states are exhaustive (eq. (3) in Section 4), only those combinations of probabilities guaranteeing $\sum_{j=1}^{S^c} p_{ij}^c = 1$ are considered (Figure B1, right). The total number of suitable combination for row i is referred to as $Z^{c,i}$.

If component c is described by a semi-Markov process, select also row i of matrix $\underline{\underline{T}}^c$. Letting the mean values, μ_{ij}^c , j = 1, 2, ..., S^c, of the holding time distributions vary within the corresponding intervals $[\underline{\mu}_{ij}^c, \overline{\mu}_{ij}^c]$, identify all the possible combinations of the mean values of row i (Figure B2). The total number of combinations obtained for the mean is referred to as $M^{c,i}$ for row i.

Repeat this step 1. for all the rows i = 1, 2, ..., S^c, of the matrices $\underline{\underline{P}}^c$ and $\underline{\underline{T}}^c$.

At the end of this step, $\sum_{i=1}^{S^c} Z^{c,i}$, $c \in \{1, \ldots, NC\}$, vectors of probability values and $\sum_{i=1}^{S^c} M^{c,i}$, $c \in \{1, \ldots, NC\}$, vectors of mean values are obtained. For example, in

Figure B3 (top) 15 transition probability vectors ($\sum_{i=1}^{S^c} Z^{c,i} = 15$, c = S2_DS2, i = 1, …, S^c = 3) are obtained for component S2_DS2: one vector for row i = 1 ($Z^{c,1} = 1$), 7 vectors for row i = 2 ($Z^{c,2} = 7$) and 7 vectors for row i = 3 ($Z^{c,3} = 7$).

B2. Obtain K^c transition probability matrices $\underline{\underline{\mathcal{P}}}^{c,k}$ [S^c x S^c], k = 1, …, K^c, for component c, $c \in$ {1, …, NC}, by performing the combinations of all the $Z^{c,i}$ vectors obtained for all the rows i, i = 1, …, S^c, at the previous step B1 (Figure B3, bottom).

If the component c is described by a semi-Markov process, find also H^c matrices $\underline{\underline{Mu}}^{c,h}$ [S^c x S^c], h = 1, 2, ..., H^c, of the mean values of the holding time distribution by performing the combinations of all the $M^{c,i}$ vectors obtained for all the rows i, i = 1, …, S^c, at the previous step B1.

B3. Repeat steps B1.-B2. for each component (c = 1, 2, ..., NC) of the SoS. All the NC components are, then, associated with a set of possible transition probabilities matrices $\underline{\underline{\mathcal{P}}}^{c,k}$, k = 1, …, K^c (resulting from the imprecise transition probabilities). In addition, the components described by a semi-Markov process (i.e., NS components) are also associated with a set of H^c matrices, $\underline{\underline{Mu}}^{c,h}$, h = 1, 2, ..., H^c, containing the mean values of the corresponding holding time distributions.

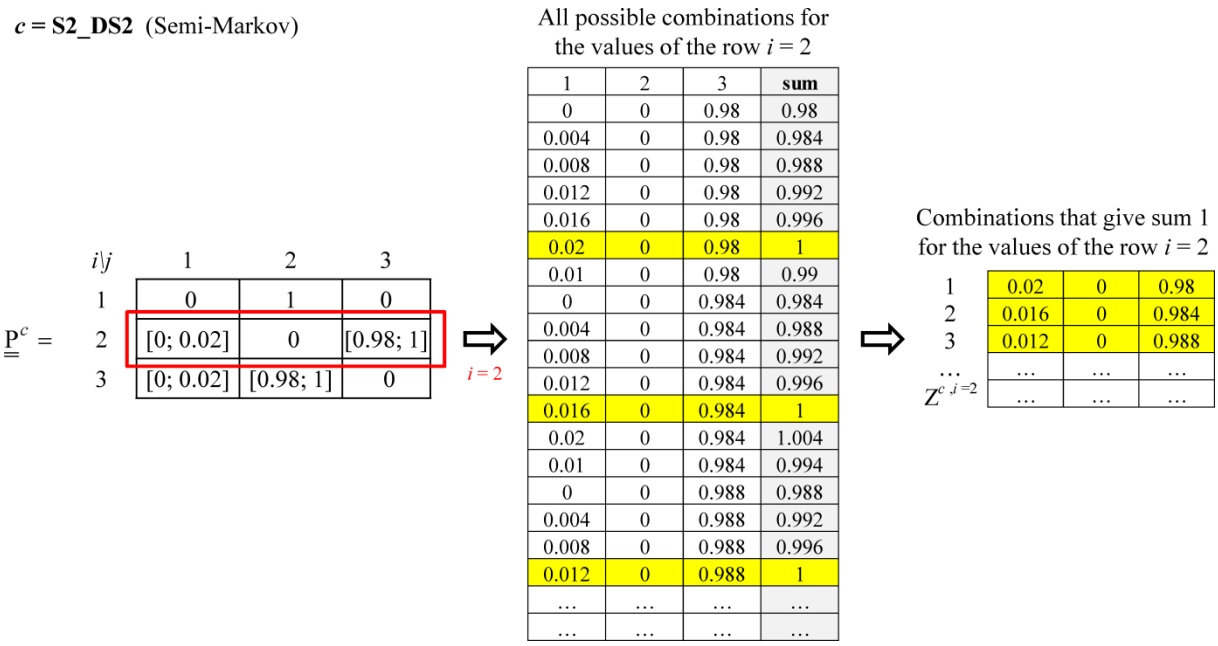


Figure B1: Exemplification of step B1 for the row i = 2 of the probability matrix $\underline{\underline{P}}^c$, c = S2_DS2, to identify $Z^{c,i}$ combinations of transition probability values.

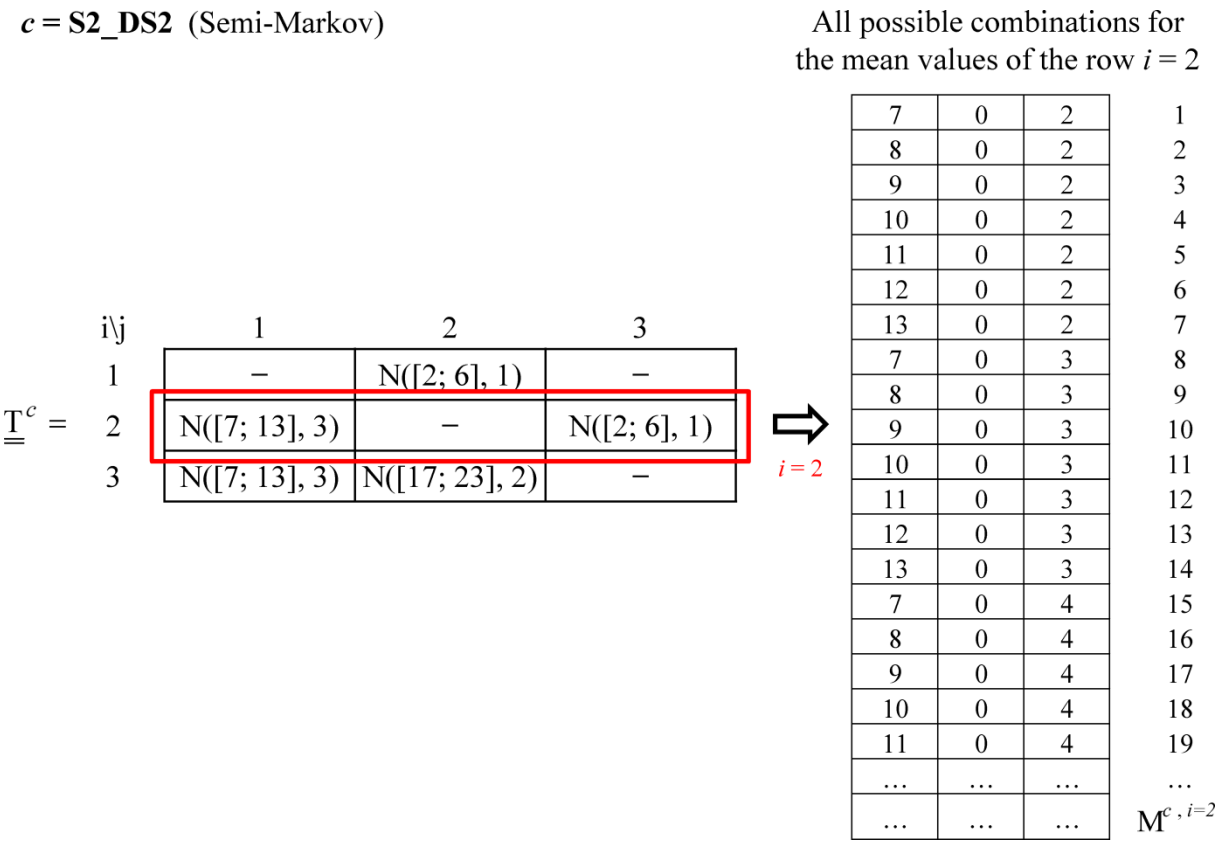


Figure B2: Exemplification of step B1 for the row i = 2 of the holding time distribution matrix $\underline{\underline{T}}^c$, c= S2_DS2, to identify $M^{c,i}$ combinations of mean values.

***c* = S2_DS2** (Semi-Markov)

Combinations that give sum 1 for the values of the row $i = 1$

$Z^{c,i=1}=1$

0	1	0

Combinations that give sum 1 for the values of the row $i = 2$

$Z^{c,i=2}=7$

0.02	0	0.98
0.016	0	0.984
0.012	0	0.988
0.008	0	0.992
0.004	0	0.996
0	0	1
0.01	0	0.99

Combinations that give sum 1 for the values of the row $i = 3$

$Z^{c,i=3}=7$

0.02	0.98	0
0.016	0.984	0
0.012	0.988	0
0.008	0.992	0
0.004	0.996	0
0	1	0
0.01	0.99	0

Transition probability matrices:

$\underline{\underline{\mathcal{P}}}^{c,1} =$

$i\backslash j$	1	2	3
1	0	1	0
2	0.02	0	0.98
3	0.02	0.98	0

$\cdots$

$\underline{\underline{\mathcal{P}}}^{c,K^c} =$

$i\backslash j$	1	2	3
1	0	1	0
2	0.01	0	0.99
3	0.01	0.99	0

Figure B3: Exemplification of step B2 to identify a set transition probability matrix $\underline{\underline{\mathcal{P}}}^{c,k}$, $k = 1, \ldots, K^c$, *for component c = S2_DS2, given the* $\sum_{i=1}^{S^c} Z^{c,i}$ *vectors obtained at step B1.*

Steps B1.-B3. above are needed in the evaluation of the recovery time and they precede step 2. of the algorithm of Section 4.2. Instead, in order to evaluate the steady state probabilities necessary to perform the robustness analysis of Section 4.1, the procedure continues as follows:

B4. Select a component c and compute the steady state probability vectors $\underline{\Pi}^{c,k}$ (or $\underline{\xi}^{c,k}$ if c is described by a semi-Markov process), $k = 1, \ldots, K^c$, one for each transition probability matrix $\underline{\underline{\mathcal{P}}}^{c,k}$, $k = 1, \ldots, K^c$, obtained at the previous step B3. If component c is described by a Markov process, eq. (4) (Section 4.1) is adopted; otherwise, if component c is described by a semi-Markov process, the output of eq. (4) is weighted by the expected time of residence, τ^i, in a given state i, $i = 1, \ldots, S^c$ [44]: $\xi^{c,k,i} = \Pi^{c,k,i} \cdot \tau^i \,/ \sum_{j=1}^{S^c} \Pi^{c,k,j} \cdot \tau^j$, $i = 1, \ldots, S^c$, $k = 1, \ldots, K^c$. For illustration purposes,

Figure B4 shows examples of the matrices $\underline{\underline{\mathcal{P}}}^{c,k}$, $k \in \{1, \ldots, K^c\}$, and $\underline{\underline{\underline{Mu}}}^{c,h}$, $h \in \{1, \ldots, H^c\}$ for component c = S2_DS2. Then, the procedure for evaluating the steady state probability vectors $\underline{\Pi}^{c,k}$ and $\underline{\xi}^{c,k}$ for Markov and semi-Markov processes, respectively, is detailed.

B5. Compute the minimum and maximum steady state probabilities $\Pi_{\min}^{c,i}$ and $\Pi_{\max}^{c,i}$, c = 1, 2, ..., NC, for each row (i.e., component state) i, $i = 1, \ldots, S^c$, as follows:

$\Pi_{\min}^{c,i} = \min_k (\Pi^{c,1,i}, \Pi^{c,2,i}, ..., \Pi^{c,k,i}, ..., \Pi^{c,K^c,i})$ and

$\Pi_{\max}^{c,i} = \max_k (\Pi^{c,1,i}, \Pi^{c,2,i}, ..., \Pi^{c,k,i}, ..., \Pi^{c,K^c,i})$, if component c is described by a Markov process, or $\Pi_{\min}^{c,i} = \min_k (\xi^{c,1,i}, \xi^{c,2,i}, ..., \xi^{c,k,i}, ..., \xi^{c,K^c,i})$ and

$\Pi_{\max}^{c,i} = \max_k (\xi^{c,1,i}, \xi^{c,2,i}, ..., \xi^{c,k,i}, ..., \xi^{c,K^c,i})$, if component c is described by a semi-Markov process. Each component c, c = 1, 2, ..., NC, is then associated with a vector of imprecise (interval) steady state probabilities:

$$\underline{\Pi}^c \in \begin{array}{c|c} i & \\ 1 & [\Pi_{\min}^{c,i=1}, \Pi_{\max}^{c,i=1}] \\ 2 & [\Pi_{\min}^{c,i=2}, \Pi_{\max}^{c,i=2}] \\ \ldots & \ldots \\ S^c & [\Pi_{\min}^{c,i=S^c}, \Pi_{\max}^{c,i=S^c}] \end{array}$$

B6. Letting the steady state probabilities $\Pi^{c,i}$, i = 1, 2, ..., S^c, of component c vary within the corresponding intervals $[\Pi_{\min}^{c,i}, \Pi_{\max}^{c,i}]$, identify all the possible combinations of the probability values to obtain a set of Q^c steady state probability vectors (obviously the sum of the components of each vector is equal to 1) (see step 2.a. of Section 4.1).

B7. Repeat steps B4.-B6. for each component (c = 1, 2, ..., NC) of the SoS.

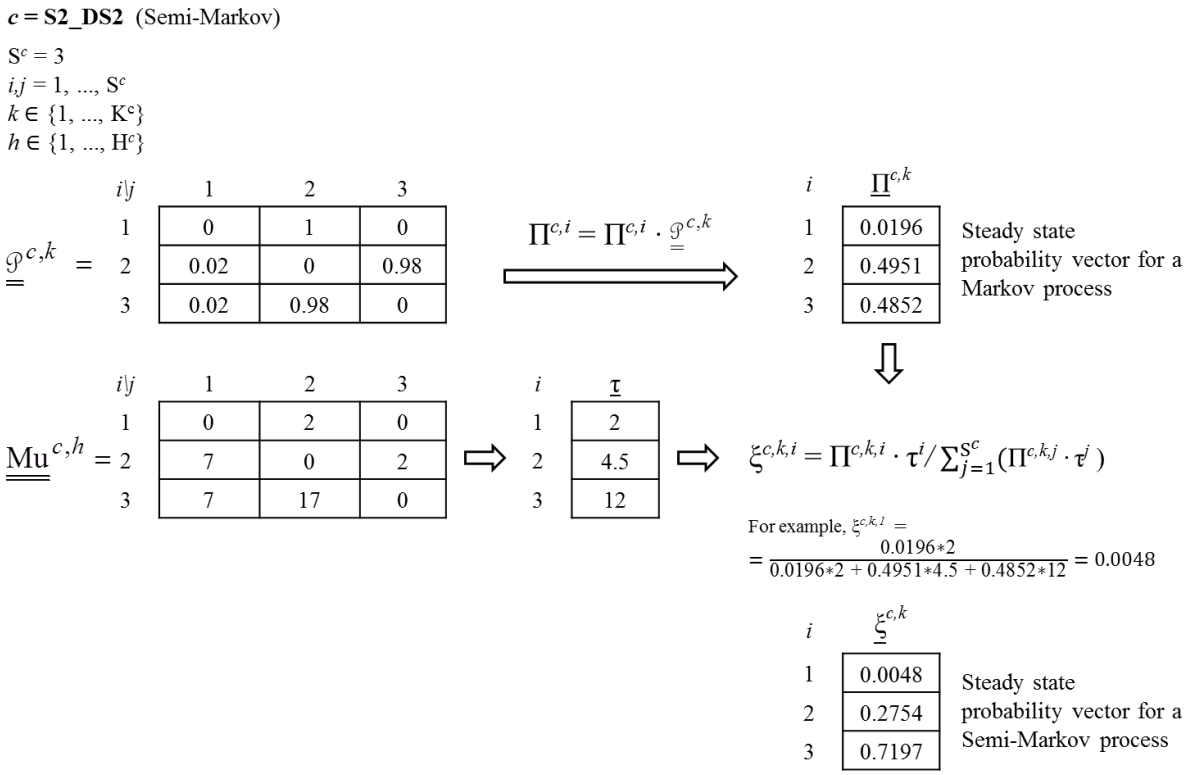


Figure B4: Exemplification of step B4 to identify the steady state probability vectors for Markov and semi-Markov processes.

Notice that in the procedure above (steps B1.-B7.) extreme lower and upper steady state probabilities $\underline{\Pi}^c_{\min}$ and $\underline{\Pi}^c_{\max}$, respectively, are obtained by resorting to a exhaustive greedy search: this amounts to identifying (in principle) all the possible combinations between (in principle) all the possible probability values in the corresponding intervals. For example, in step B1 the probabilities p^c_{ij} are allowed to range within their intervals $[\underline{p}^c_{ij}, \overline{p}^c_{ij}]$: for the sake of practical computation we identify, e.g., 7 discrete values within each interval $[\underline{p}^c_{ij}, \overline{p}^c_{ij}]$. If we assume that the number of states is $S^c = 3$, then the total number of possible combinations between the transition probability values is 343; if the number of states is 7, i.e., $S^c = 7$, the number of possible combinations increases to 823543. Obviously, the higher the number of discrete values taken within the intervals $[\underline{p}^c_{ij}, \overline{p}^c_{ij}]$, the more precise the results, but the more

prohibitive the computational cost. For these reasons, when the dimension of the transition probability matrix increases, we need to resort to alternative (intelligent) techniques: in other words, in order to obtain the lower and upper steady state probabilities $\underline{\Pi}^{c}_{\min}$ and $\underline{\Pi}^{c}_{\max}$, respectively, we do not analyze all the possible combinations between all values of $p^{c}_{ij} \in [\underline{p}^{c}_{ij}, \overline{p}^{c}_{ij}]$; instead we intelligently explore only those combinations that driving the search appear as the most "promising" for the maximization and minimization of $\underline{\Pi}^{c}$. In this work, we resort to Genetic Algorithms (GAs) for the analysis of arcs a_b, b_c, c_d, d_e, whose transition probability matrices have size 7 x 7. In particular, we run the Matlab function "ga" twice to find the minimum and maximum steady state probability vectors $\underline{\Pi}^{c}_{\min}$ and $\underline{\Pi}^{c}_{\max}$, respectively. In more details, eq. (4) of Section 4.1 represents the function to be optimized (i.e., minimized and maximized, respectively) by the GA, eq. (3) of Section 4.1 represents the equality constraints to satisfy and eq. (2) shows the upper and lower bounds of the transition probabilities p^{c}_{ij} needed in (4).