



**HAL**  
open science

## Methodology for imprecise availability computing and optimization

Joanna Akrouche, Mohamed Sallak, Eric Chatelet, Fahed Abdallah, Hiba Haj Chhadé

► **To cite this version:**

Joanna Akrouche, Mohamed Sallak, Eric Chatelet, Fahed Abdallah, Hiba Haj Chhadé. Methodology for imprecise availability computing and optimization. 29th European Safety and Reliability Conference (ESREL 2019), Sep 2019, Hannover, Germany. pp.2440-2446, 10.3850/978-981-11-2724-3\_0599-cd. hal-02444841

**HAL Id: hal-02444841**

**<https://hal.science/hal-02444841>**

Submitted on 19 Jan 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Methodology for imprecise availability computing and optimization

Joanna Akrouche

*Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc, CS 60 319, 60 203 Compiègne cedex, France. E-mail : [joanna.akrouche@hds.utc.fr](mailto:joanna.akrouche@hds.utc.fr)*

Mohamed Sallak

*Sorbonne universités, Université de technologie de Compiègne, CNRS, Heudiasyc, CS 60 319, 60 203 Compiègne cedex, France. E-mail: [mohamed.sallak@hds.utc.fr](mailto:mohamed.sallak@hds.utc.fr)*

Eric Châtelet

*Université de Champagne, Université de technologie de Troyes, CNRS, Institut Charles Delaunay, 12 rue Marie Curie, CS 42060, 10004 Troyes cedex, France. E-mail: [eric.chatelet@utt.fr](mailto:eric.chatelet@utt.fr)*

Fahed Abdallah

*Lebanese University, Beirut-Hadath, Lebanon. E-mail: [fahed.abdallah76@gmail.com](mailto:fahed.abdallah76@gmail.com)*

Hiba Haj Chhadé

*Lebanese University, Beirut-Hadath, Lebanon. E-mail: [hiba.hajchhade@gmail.com](mailto:hiba.hajchhade@gmail.com)*

One of the undeniable steps in the safe design of systems is the problem of choosing the best system configuration in the most effective way so as to maximize the overall system availability and to minimize the overall system cost. The main objective of our paper is to propose a methodology of optimization of the availability of multi-states systems with multi-states components in presence of both aleatory and uncertainties. The problem is formulated as follows: let us consider several configurations of a system, each configuration consisting of components with several states of working and imprecise failure and repair rates provided in form of intervals. The objective is to find the best configuration regarding the system availabilities and costs. We first compute the imprecise steady availability of each configuration by using an original method based on Markov approaches combined with interval contraction techniques. We also compute the overall cost of each configuration. When having the availability and the cost of all the configurations, the idea is to define an objective function in terms of cost, lower and upper bounds of availability, and imprecision (length of availability interval). Then, this function is computed so that we have the best configuration according to our criterion. To illustrate our proposed methodology, we will propose a use-case describing a system formed of  $n$  components, and each component has different degraded states from perfect functioning to total failure. Each configuration of the system will have a different structure (parallel, series, series-parallel, complex,...), with different choices of component characteristics (imprecise failure and repair rates). Then, according to different criteria as high availability, low cost, or low availability imprecision, we will propose the best system configuration.

*Keywords:* Availability, multi-states system, imprecision, interval analysis, optimization, Markov approaches, the technique of contractors.

### 1. Introduction

Nowadays, availability analysis is an important part when we are dealing with some system, it is concerned with the systems whose individual components are apt to failure. Since, The Availability  $A(t)$  is the ability of an item to be in a state to perform as and when required, under given conditions, assuming that the necessary external resources are provided Levitin (2005). Operating dependability means that must be mastered when designing any system and availability analysis is well needed in any

domain do determine the lifetime of the product or the system. Availability analysis helps to calculate the ability of system to provide a required level of performance depending on the level of degradation. Many methods exist to calculate the availability of systems: Universal Generator Function (UGF) Levitin (2005), inclusion-exclusion method Mihova and Maksimova (2011), Monte Carlo simulations, and Markovian models Troffaes et al.(2015). However, systems became more complex; by the number of components, the number of states for

*Proceedings of the 29th European Safety and Reliability Conference.*

*Edited by Michael Beer and Enrico Zio*

Copyright © 2019 European Safety and Reliability Association.

*Published by Research Publishing, Singapore.*

ISBN: 978-981-11-2724-3; doi:10.3850/978-981-11-2724-3-0599-cd

2440

each component (perfect functioning, degraded state, failure state), the case of multi-states systems, the structure of the system. That's why availability analysis is now more complex when dealing with complex multi-states systems. What's make it even worse, the presence of uncertainties on the components data (failure rate and repair rate), since the knowledge that we have about the components data is generally uncertain.

In this article, we explain a method that helps us to calculate the imprecise availability of a complex system. Later, we will detail our case of study in the aim to optimize the availability among different configurations of system. To end up with the conclusion and some perspectives where we can develop our case to different cases.

**2. Methodology of imprecise availability analysis**

When we study the dependability of a system, we are confronted in many cases with the fact that the system and its components may have different states or modes of functioning with different performance levels. Such systems are called multi-states systems (MMS). In our work, the data that we have are uncertain, this requires methods that allow modeling and manipulation of these uncertainties such as probability theory, fuzzy set theory, belief functions theory. In this work, we propose the use of imprecise Markov models for the calculation of the availability of an MMS. In a previous paper Akrouche al. (2017), we have some preliminary work about this subject.

**2.1 Problem statement**

The main issue in our work is to calculate the availability of complex MSS in presence of different types of uncertainties. Indeed, we are interested in uncertainties about reliability data (failure rate  $\lambda$  and repair rate  $\mu$  of each component), due to the difficulty to estimate these data (new components, rare components failures, expensive components) and the transitions rates could be variable over time and affected by several factors. The proposed way to cope with these uncertainties is to use the theory of imprecise probabilities, and particularly interval probabilities. We consider our transition rates as not being precise, but instead being bounded by intervals. That's why we propose to treat the data that we have in the terms of intervals and to apply interval analysis on imprecise Markov approaches, so we could calculate the imprecise availability of a system.

**2.2 The proposed method**

In our study, we consider a complex MSS where the components and the system are multi-states and the number of components is important. The repair rates and the failure rates are represented in terms of intervals. To calculate the imprecise availability, we chose to use Markovian approaches.

*2.2.1 Imprecise Markovian approaches*

Markovian models represent a class of stochastic processes where system is "memoryless" El Falou and Châtelet (2011), i.e., the future state of the system is independent of the previous states. The time in our case is continuous. The transition rates  $q_{ij}$  are in terms of failure rates and repair rates.  $Q$  the transition matrix is the matrix containing these elements. Note that the sum of the elements of each line of  $Q$  is equal to 0. Since we focus on the imprecision of failure and repair rates, we will take each element  $q_{ij}$  as an interval  $q_{ij} = [q_{ij}; \bar{q}_{ij}]$ , where  $q_{ij}$  and  $\bar{q}_{ij}$  represent respectively the lower and upper bounds of the interval the interval. Thus, the transition matrix  $Q$  will be in the form of an interval  $Q = [\underline{Q}; \bar{Q}]$ .

*2.2.2 Evaluation of the Availability*

We can build the state equation of a system S associated with the discrete space  $E$  in involving instantaneous transition rates  $q_{ij}$ , because they exist as a system of differential equations of Chapman-Kolmogorov. We thus consider for each state  $e_i \in E$  (see Akrouche et al. (2017)):

$$P_i(t + dt) =$$

$$P(S \text{ in the state } e_i \text{ at time } t \text{ and during } [t, t + dt]) + \sum_{e_j \in E - \{e_i\}} P(S \text{ in the state } e_j \text{ at } t \text{ and in } e_i \text{ at } (t + dt)) \tag{1}$$

From the system of equations of all the states, we get the following formula

$$\dot{P}(t) = P(t) \cdot Q \tag{2}$$

Where  $P(t)$  is the probability states vector of the system at a time  $t$ . We are interested in the stationary states probabilities that's why we need to compute the asymptotic availability. Using Eq.(2), we obtain the following asymptotic solution

$$\Pi \cdot Q = 0 \quad (3)$$

where  $\Pi = [\pi_i]$  is the steady probability vector of the system to be in a state  $i$ ;  $\lim_{t \rightarrow \infty} P_i(t) = \pi_i$  and we always have the property  $\sum_{i=1}^n \pi_i = 1$ . Eq. (3) is a system of equations that we need to solve in order to find the intervals  $\pi_i = [\underline{\pi}_i; \overline{\pi}_i]$ . Then, the asymptotic availability will be computed as the sum of the probabilities  $\pi_i$  corresponding to the working states. Solving the system of these equations isn't simple, therefore we introduce a method inspired from interval analysis: the technique of contraction in Jaulin et al. (2001).

### 2.2.3 The technique of contraction

The contraction technique in Jaulin et al.(2001), was introduced for the first time in 1970 in the artificial intelligence field. It is a method of constraint satisfaction problem and was later developed in interval analysis methods. This technique makes it possible to contract an interval  $[x]$  by operators named contractors in order to obtain an interval  $[x']$ , such that  $S \subset [x'] \subset [x]$  where  $S$  is the exact interval (note that this technique is more detailed in the appendix). We will use the contractor Forward-Backward propagation for several reasons. It is applicable for all kinds of equations, besides calculations are much simpler than with other contractors, the obtained intervals after the contraction are guaranteed that they belong to the initials intervals. The Forward-Backward propagation technique consists of two steps, the first is the "Forward": it proposes to make an arithmetic calculation of the intervals. The second step is the "Backward": it proposes to perform the same arithmetic calculation already made but in the opposite order, which means from the last operation made to the first one, taking into account in each computation step to take the intersections of  $[x']$  and  $[x]$ . In our case the intervals  $[x]$  are the  $[\pi_i]$  that we need to contract them to obtain the smallest intervals, so we could calculate the availability of the system.

### 2.2.4 Methodology of the proposed method

To calculate the availability of a complex MSS with multi-state components, we first need to illustrate its Markov graph. So, we could present all the transitions from one state to another, from the total working state to the total failure state of the system. We define the interval transition matrix in terms of failure rates and repair rates of dimension  $(n \times n)$ . To determine the interval availability of the system, first we take the initial vector  $\Pi$  as  $\Pi = [[0,1][0,1] \dots [0,1]]$ , and we apply Eq. (3) where we obtain a system of  $n$

equations with the last equation representing the fact that  $\sum_{i=1}^n \pi_i = 1$ . We apply the Forward-Backward on the intervals  $\pi_i = [\underline{\pi}_i; \overline{\pi}_i]$ , to reduce  $\Pi$  as much as it is possible. We keep repeating the contraction on all the equations and several times, until the vector  $\Pi$  converges. When we obtain  $\Pi$ , we calculate the imprecise availability of the system  $A$  (in form of interval  $A = [\underline{A}; \overline{A}]$ ) by computing the sum of  $\pi_i$  over all the working states. To verify if the results obtained by the technique of contraction is accurate, we will compare it to the availability obtained by taking all the possible transition matrix by doing a combination between the lower bounds and upper bounds of the failure rates interval and the repair rates interval, so for each combination we obtain a transition matrix, where we solve the system from Eq.(3) and find the corresponding  $\Pi$  and the availability, after that we compare all the obtained availabilities from all combinations, so we can choose the corresponding availability interval where its bounds are the lowest availability (lower bound) and the highest availability (upper bound). We will call this method the "Exact method". In addition, we can verify the results, by taking the center of each interval of our data, so we will get one transition matrix formed from all the midpoint values of the data, we solve Eq (3) to get the precise vector  $\Pi$ , where we could finally calculate the precise availability, which it has to belong to the interval obtained by the contraction technique and to the interval obtained by the exact method. We will call this method the "Precise method". To understand the method for complex MSS, with all the details, an example is presented in a previous paper Akrouche et al.(2018).

## 3. Optimization

In the previous part, we explained a method to calculate the availability of a multi-state system with a given structure in the presence of uncertainties. In this part, we will propose a general methodology to solve the problem of optimal design of multi-state systems. The problem is to find the best architecture among  $n$  configurations of which we have a better availability with the best cost for that we seek to minimize the overall cost of the multi-state system. As soon as we get the optimal availability and optimal cost we can build the different optimal architectures corresponding to the available and the cost obtained.

### 3.1 The methodology

We present the different steps of our methodology for optimal design of MSS:

- Define the different system structures that we have:
  - a) The number of components in each structure.
  - b) Failure rates and repair rates.
  - c) The working states of each component for each structure.
  - d) The different costs for each component in each structure.
- Our aim is to choose the components and their connection that will achieve the optimal MSS structure with minimal overall cost.
- Modelling the structure of each MSS by a block diagram (or a Markov chain). We use the method defined in the first part to find the  $n$  availabilities for the  $n$  structures with the  $n$  cost  $C$ , where  $C = \sum_{i=1}^k C_i$ ,  $k$  is the number of components for a structure.
- To ensure the constraints imposed, we must ensure that the availability  $A_{Optimal}$  of the MSS remains above a minimum value  $A_{min}$  and below a maximum value  $A_{max}$ . For this purpose, we will study the influence of imprecision use an availability correction, and  $A_{min}$  is determined. The parameter  $\alpha$  makes it possible to minimize the cost. Our aim is to find the best system with the components that gives the best availability with a low cost. Therefore, we need to reduce the cost and raise the availability interval. We increase the availability interval of a system, by increasing its lower bound and reducing the width of the availability interval. The objective function is in terms of the sum of the total cost of the components, the average of the availability and the lower bound of the availability interval, since it is important to define the best availability with the lower cost. The objective function is given by:

$$f = -\alpha_1 C / 10000 - \alpha_2 (\bar{A} - \underline{A}) + \alpha_3 \underline{A} \quad (4)$$

With  $\alpha_i$  are constants defined by the user, such that  $\sum_i \alpha_i = 1$ .

### 3.2 Numerical example

Let's take an example of a system of 4 binary components, where all the information (failure rate  $\lambda \times 10^{-3}$  per hour, repair rate  $\mu \times$

$10^{-2}$  per hour and the cost  $C$  in Euros) for each component are represented in Table 1.

Table 1. Given data for each component

Component	$\lambda$	$\mu$	$C$
1	[3,4]	[2.5,3.1]	1500
2	[4,4.8]	[2,3]	1200
3	[3.5,4.1]	[3,3.5]	1700
4	[2,2.8]	[2,2.8]	2000

From the four components, we can find 26 possible architecture of systems, with the cost of each system is the sum of the costs of components (6400). Each system will have different interval availability, that we will calculate it by using the method described in the previous section. Table 2 shows all the possible architectures and the corresponding availability for each system.

Table 2. The availability for each architecture of 4 components.

System	Architecture	A
1	Series	[0.4481,0.9996]
2	Parallel	[0.999,1]
3	Parallel (2,3,4)-series(1)	[0.8348,0.9999]
4	Parallel (3,4)-series(1,2)	[0.622,0.9998]
5	Parallel (2,4)-series(1,3)	[0.6926,0.9998]
6	Parallel (2,4)-series(1,3)	[0.6607,0.9999]
7	Parallel (1,3,4)-series(2)	[0.7568,0.9999]
8	Parallel (1,4)-series(2,3)	[0.6364,0.9998]
9	Parallel (1,3)-series(2,4)	[0.6645,0.9998]



10	Parallel (1,2)- series(3,4)	[0.6751,0.9998]
11	Parallel ((1,2)- (3,4))series	[0.937,1]
12	Parallel ((1,3)- (2,4))series	[0.9343,1]
13	Parallel ((1,4)- (2,3))series	[0.9383,1]
14	Series ((1,2)- (3,4))parallel	[0.8922,0.9999]
15	Series ((1,3)- (2,4))parallel	[0.895,0.999]
16	Series ((1,4)- (2,3))parallel	[0.891,1]
17	Series ((1)- (2)- (3,4))parallel	[0.9885,1]
18	Series ((1)- (3)- (2,4))parallel	[0.9916,1]
19	Series ((1)- (4)- (2,3))parallel	[0.9897,1]
20	Series ((1)- (2,3,4))parallel	[0.9253,1]
21	Series ((2)- (1,3,4))parallel	[0.9015,1]
22	Series ((3)- (1,2,4))parallel	[0.9343,1]
23	Series ((4)- (1,2,3))parallel	[0.9162,1]
24	Series ((2)- (3)- (1,4))parallel	[0.9896,1]

25	Series ((2)- (4)- (1,3))parallel	[0.9877,1]
26	Series ((3)- (4)- (1,2))parallel	[0.9906,1]

When we apply Eq.3 on all the architectures in Table 2, we find that the best result is  $\alpha$ , where

$\alpha$ , which means that the second architecture of the parallel system is the optimal architecture. We notice that even when we change the values of  $\alpha$  the parallel system (Figure 1) has always the optimal function, while the ranking of the best architectures change for the rest.

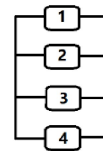


Figure 1. Parallel system of 4 components

#### 4. Conclusion

With a simple example, where we have a small number of binary components that make a simple system, we can develop the methodology of optimization of the availability described in this article, to more complex MSS where we have a large number of multi-states components. In this way we can choose the best architecture of any system, that survive the most with an optimal cost of products.

#### Appendix A. Basics of the contraction technique

Consider  $n$  variables  $x_i \in \mathbb{R}, i = 1, \dots, n$ , linked by  $f$  relations (or constraints) Jaulin et al.(2001) of the form:  $f_j(x_1, \dots, x_n) = 0, j = 1, \dots, f$ , it can be written in vector notation as:  $f(x) = 0$ . Each variable  $x_i$  is known to belong to a domain  $D_i$ . For simplicity, these domains will be intervals, denoted by  $[x_i]$ . Define the vector  $x = (x_1, \dots, x_n)^T$  and the prior domain for  $x$  is a box as:  $[x] = [x_1] \times \dots \times [x_n]$ . Let  $f$  be the function whose coordinate functions are the  $f_j$ 's. Equation  $f(x) = 0$  corresponds to a constraint satisfaction problem (CSP)  $H$ , which can be formulated as:  $(H: f(x) = 0, [x])$ . The solution set of  $H$  is defined as:  $S = \{x \in [x] \mid f(x) = 0\}$ . Contracting  $H$  means replacing  $[x]$  by a smaller domain  $[x']$  such that the solution set remains unchanged, i.e.

$S \subset [x'] \subset [x]$ . There exists an optimal contraction of H, which corresponds to replacing  $[x]$  by the smallest box that contains S. A contractor for H is any operator that can be used to contract it. A contractor C is defined as an operator used to contract the initial domain of the CSP, and thus to provide a new box. Several contractors exist, each works in a different manner and is efficient only for specific CSPs and for certain cases in Jaulin et al.(2001). The most popular contractor, which will be used in our approach, is the “Forward-backward propagation (FBP) contractor”. This technique is known for its simplicity and easiness, it is also more general than the others since it works on all type of systems see Jaulin et al. (2001). It also gives guaranteed results which means that during the contraction we always get an interval belonging to the initial interval. The Forward-backward (FBP) contractor  $C \downarrow \uparrow$ , is a classical algorithm in constraint programming for contracting. This contractor makes it possible to contract the domains of the CSP H by taking into account each one of the  $n_f$  constraints apart. The algorithm works in two steps . The forward step applies interval arithmetic to each operator of the function  $y=f(x)$ , from the variable's domain ( $[x]$ ) up to the function's domain ( $[y]$ ), this step considers the direct forms of the equations. The backward step sets the interval associated to the new function's domain  $[y]$  to  $[0, 0]$  (imposes constraint satisfaction, since we are solving  $f(x)=0$ ) and, then, applies backward arithmetic from the function's domain to the variable's domain, which means using the inverse of the functions that appear in the equations  $f(x)$ . The following example explains the procedure of the FBP technique.

*Example:* Consider the constraint  $y = -5x_1 + 2x_2 = 0$  and the initial box-domain  $[x] = [1, 4] \times [-3, 7]$ . This constraint can be decomposed into three primitive constraints (i.e. constraints associated with a unique elementary function: multiplication or addition) by introducing two intermediate variables  $a_1$  and  $a_2$  defined as:  $a_1 = -5x_1$  and  $a_2 = 2x_2$ . Initial domains for these variables are determined as follows:

$$\begin{aligned} a_1 &= -5x_1 = -5 \times [1,4] = [-20,-5] \\ a_2 &= 2x_2 = 2 \times [-3,7] = [-6,14] \\ y &= a_1 + a_2 = [-20,5] + [-6,14] = [-26,9] \quad (A.1) \end{aligned}$$

and this step is called the "forward propagation". A method for contracting H with respect to the constraint  $f(x) = -5x_1 + 2x_2 = 0$  is to contract each of the primitive constraints in (A.1) until the contractors become inefficient. For this example: Since  $f(x) = 0$ , the domain for y should be taken equal to  $\{0\}$ , so we can add the step:  $[y] := [y]$

$\cap \{0\}$ . If  $[y]$  as computed in (A.1) turns out to be empty, then the CSP has no solution. Else,  $[y]$  is replaced by 0, which is the case in this example. After, a backward propagation is performed, updating the domains associated with all the variables to get:

$$\begin{aligned} [a1] &:= ([y] - [a2]) \cap [a1] = [-14, -5] \\ [a2] &:= ([y] - [a1]) \cap [a2] = [5,14] \\ [x1] &:= ([a1] / -5) \cap [x1] = [1,14/5] \\ [x2] &:= ([a2] / 2) \cap [x2] = [5/2,7] \quad (A.2) \end{aligned}$$

Thus, we obtain the new box:

$[x](1) = [1, 14/15] \times [5/2, 7]$ , which is the result of the first FBP contraction. Iterating this procedure, the resulting sequence of boxes  $[x](k)$  converges towards the smallest possible domain, after which the domains no longer change following another iteration of FBP.

**References**

Akrouche, J., M. Sallak, E. Châtelet, F. Abdallah and H. Haj Chhadé (2017). *A contribution to the evaluation of imprecise availability of complex systems using Markov models*. UNCECOMP 2017, Greece 1517.

Akrouche, J., M. Sallak, E. Châtelet, F. Abdallah and H. Haj Chhadé (2018). *New method for availability computing of complex systems using Imprecise Markov models*. MIMAR 2018.

El Falou, M. and E. Châtelet (2011), *Analytical Uncertainty Propagation for Availability Assessment of Stochastic Multi-state Systems*, International Journal of Performability Engineering, 7(3): 251-261.

Jaulin, L., M. Kieffer, O. Didrit, E. Walter (2001). *Applied Interval Analysis*. Springer.

Levitin, G. (2005). *The universal generating function in reliability analysis and optimization*. Springer.

Mihova, M. and N. Maksimova (2011). *Estimation of minimal path vectors of multi state two terminal networks with cycles control*. Mathematica Balkanica, 25(4): 437-447

Troffaes, M., J. Gledhill, D. Skulj, S. Blake (2015). *Using Imprecise Continuous Time Markov Chains for Assessing the Reliability of Power Networks with Common Cause Failure and Non-Immediate Repair*. 9th International Symposium on Imprecise Probability: Theories and Application, Pescara, Italy.