A Multistage Stochastic Program for the Design and Management of Flexible Infrastructure Networks

Samuel Torres-Rincón[a,b,*], Mauricio Sánchez-Silva[a], Emilio Bastidas-Arteaga[b,c]

[a] *Civil and Environmental Engineering Department, Universidad de los Andes, Bogotá, Colombia*
[b] *University of Nantes, Institute for Research in Civil and Mechanical Engineering, CNRS UMR 6183, Nantes, France*
[c] *La Rochelle University, LaSIE, CNRS UMR 7356, La Rochelle, France*

Abstract

Throughout their lifetime, infrastructure network systems face unplanned events that impose pressures on their integrity, functionality, and ability to deliver value. Most of the existing infrastructure is designed to deal with the challenges imposed by uncertain external phenomena. Authors from different backgrounds have identified flexibility, changeability, and adaptability as key attributes that modern systems should have to face uncertain scenarios. Specifically, flexibility is an ability that allows a system to be *easily* adapted when necessary. The concept of flexibility is compelling, but it is not clear how to measure the value it may provide. Determining how much to pay to introduce flexibility is an essential aspect of designing flexible systems, but the dependence of this value on the future evolution of the system results in a complex decision process. The sequential nature of the process can be modeled using multistage stochastic programming. The model explicitly considers the flexibility built into the network components as a decision variable at the initial stage. The model is tested in a generic infrastructure network that must meet a stochastic demand. The results show the relationship between the value of flexibility and the life-cycle costs at the construction and operation stages.

Keywords: Flexibility, Multistage Stochastic Programming, Value of Flexibility, Infrastructure Networks

*Corresponding author
Email address: sf.torres405@uniandes.edu.co (Samuel Torres-Rincón)

1. Introduction

1.1. Background

Infrastructure networks, as any infrastructure system, are constantly exposed to ever-changing conditions. Changes in the demand, the supply sources (in the case of water and energy distribution systems), and regulations may impose large stresses on the system [1, 2, 3, 4]. This uncertainty can either challenge or boost the ability of the network to perform at required levels [5]. In the former case, networks usually rely on the robustness provided by over-dimensioned designs which can be largely inefficient, as in the case of sanitation systems [6]. In the latter case, even if the new conditions are favorable, the network may not have the tools to take advantage of the new opportunities [7]. To face these challenges, many authors have identified the need to develop flexible and adaptable systems, which have the potential to improve the sustainability and efficiency of the system under highly uncertain conditions [8, 9, 10, 11, 12].

Specifically, flexible systems are defined as systems with the ability to change as "easily" as possible [13, 14, 15, 4, 16]. This can be measured as the money or time investment required to modify the system. In the case of infrastructure networks, adaptations can be usually seen as the addition (or removal) of nodes and links or the expansion (or downgrade) of their capacity. Other changes, mostly managerial, may be possible (e.g. reversal of the flow direction). Flexibility is introduced by investing a number of resources during the design and construction phase to enable the option of deploying future adaptations at lower costs and shorter lead times. This investment is usually called the *value of flexibility* and is defined as the amount the system stakeholders' are willing to pay to introduce flexibility into the system [7, 8]. It is assumed that the enabled adaptations are capped at a certain value that depends on the amount invested at the initial stages of the project [16]. This interpretation of flexibility is similar to the concept of shell capacity described by Angelus et al. [17].

1.2. Previous works and applications

While flexibility is not limited to capacity expansion, this type of problem is the most common and has been extensively studied in the manufacturing sector [18, 19, 20]. The first work comes from Manne [21], who modeled the optimal excess capacity of pipelines, highways, and steel plants subject to a random walk demand. Luss [18] further detailed the problem as determining the size, location, and timing of future adaptations while minimizing the discounted costs of the expansion processes. This author also identified the modeling of the demand, the representation of the capacity as a continuous or discrete variable, the selection of the discount rate, and the presence of economies of scale as key elements of the problem formulation. Mathematical programming rapidly became one of the preferred methods to solve this type of problem, as shown by the work of Rajagopalan [22], who formulated a mathematical programming model to determine the optimal initial capacity and the technology acquisition decisions, for increasing demands. Later came the work of Ahmed and Sahinidis [23] who developed a multistage stochastic (MS) integer program to analyze a multi-period investment model for capacity expansion under uncertain demand and costs, and including the effect of economies of scale. Singh et al. [24] also developed an MS integer programming model for planning discrete capacity expansion of production facilities. Similarly, Huang and Ahmed [25] formulated a general MS capacity planning model with discrete capacity for the semiconductor industry.

The prevalence of stochastic and dynamic programming models to solve capacity expansion problems in the manufacturing sector led to the adoption of these techniques in other areas such as planning and design of infrastructure networks. For instance, in the context of transportation networks, Marín and Jaramillo [26] formulated a multi-period capacity expansion problem for rapid transit network design. Karoonsoontawong and Waller [27] developed a robust optimization model for the problem of continuous traffic network capacity expansion with dynamic traffic assignment and traffic signal optimization. Gao et al. [28] considered the case where a road network is expanded either by adding new links or by increasing the capacity of the existing links, integrated with the road maintenance problem using a mixed-integer, non-linear, bi-level optimization program, using multi-period decisions. In the area of water and sewage distribution systems, Mortazavi-Naeini et al. [9] developed a multi-objective optimization approach for the planning of urban water system expansions that considers the combined effect of operating rules and infrastructure conditions. Saif and Almansoori [29] developed a model for the expansion of water desalination and power supply infrastructure using a deterministic multi-period mixed-integer linear programming formulation. Similarly, Fraga et al. [10] used dynamic programming to develop an integrated framework for the optimization of water supply system expansions, considering short and long-term water supply sources. In the case of energy production and transmission networks, Loureiro et al. [30] used the concept of real options combined with a mixed-integer linear programming model for multistage expansion planning. Cardin et al. [31] combined decision rules and stochastic programming to model flexibility in a nuclear infrastructure system, considering a random demand and including the social acceptance of nuclear power as a limitation to the system expansion capabilities. For networks in general, Taghavi and Huang [32] applied the concepts of spot market and contract capacity combined with an MS integer program to model a network with multiple sources of capacity.

In both manufacturing systems and infrastructure networks, the adaptation process requires large capital investments (in proportion with the total system size), which are sometimes irreversible [33], and are susceptible to the negative consequences of large lead times. In both cases, there is a contradictory effect from economies of scale, which may favor both large initial designs and large future adaptations. However, infrastructure networks are subject to more strict governmental regulations, where failure to comply with minimum levels of performance can result in heavy societal costs. This poses an additional problem of strategically selecting the extent to which flexibility should be incorporated. Besides, infrastructure networks are usually planned for longer service horizons, which favors delayed deployments due to discounting [34]. Furthermore, the nature of stakeholders affects the management and adaptation strategies. While manufacturing systems are generally completely privately owned, infrastructure network ownership can range from completely public to completely private, which may create conflicting interests. For instance, in energy infrastructure, maintaining a determined level of excess capacity may be valuable for society but not for profit maximization [35]. Finally, the frequency of adaptation is considerably different: in manufacturing systems is not uncommon to see changes every one or two years, while infrastructure networks may see changes in periods of five to ten years.

1.3. Objectives and scope

The works discussed in the previous section show the potential of (and the preference for) MS programming as a tool to model and solve complex sequential decision problems. Considering that the problem of designing and managing flexible infrastructure networks is in itself a sequence of decisions under uncertainty, this paper proposes a multistage stochastic program (MSP) to analyze the problem of flexibility. Operational and maintenance costs are explicitly considered, to take into account the relative importance of these costs in the systems modeled and the impact these costs may have on the preference of the model for flexible solutions. The model uses scenario trees generated by Monte-Carlo simulation and k-medoids clustering to simulate and discretize the random process. The novelty of the model is that it explicitly considers the flexibility range introduced by design as a decision variable, which will restrict all the future planned adaptations. The proposed model results in an alternative methodology to real options as a tool to determine the *value of flexibility*. By modeling flexibility as a decision variable, the program decides how much should be paid at the initial stage to lower future expenses.

The article is organized as follows: 2 gives an overview of the concept of flexibility in infrastructure systems and its numerical representation. A review of the theory behind two-stage and MSPs is presented in 3. 4 provides the formulation of the proposed MSP. 5 presents a methodology to solve the MSP using an approximate approach. Finally, **??** provide a numerical example with a generic network and a summary of the key results.

2. Designing for Flexibility

The concept of flexibility is considered to be not "academically mature" [14] due to the lack of a precise and universally accepted definition. In this study, flexibility will be understood as defined by [7, 15], and [16]. The authors define flexibility as the ability of a system to *easily* adapt any of its components (or subsystems). The effort necessary to complete an adaptation is measured as the number of required resources, which are usually represented as a monetary quantity. Furthermore, any adaptation will be limited to a maximum value given by the system's context and the resources invested to make the system flexible. These additional costs may be incurred at the initial stage of the project to introduce the option to adapt some of the system components. These expenses may come from research and development activities or by installing physical elements that will facilitate future adaptations. For instance, the floating platforms used in offshore wind farms could be designed to be expanded or be built larger-than-required to facilitate future adaptations to the turbine sizes.

The problem of determining how much should be invested to have flexibility in the system is usually known as measuring the *value of flexibility*. This is a complex problem because capturing the value of having the option to modify the system under certain circumstances is not a straightforward process [8]. Different approaches have been developed based on different assumptions and modeling frameworks. For instance, the Real Options Analysis (ROA) method was developed by adapting the concept of *options* in financial markets to physical assets [36, 37]. Under this approach, the value of flexibility is measured as the pricing of an option that provides the right but not the obligation to modify the system in some predefined way. In a different approach, Cardin et al. [7] defined the value of flexibility as the difference between the expected net present value (ENPV) of a flexible system and an inflexible system subjected to the same conditions. Similarly, Špačková and Straub [15] defined it as the additional investment that should be made to have a flexible system compared with the inflexible alternative, using Markov Decision Processes (MDP) to model the sequential decision process. Following these works, the value of flexibility will be understood in this paper as the maximum value that should be paid at the start of the system life-cycle to minimize the discounted costs of future adaptations and maximize the received utility.

In the area of chemical processes, the concept of Flexibility Analysis (FA) has been proposed to optimize the design and operation of chemical plants under uncertainty [38, 39, 23, 40]. The main objective of this approach is to determine the optimal design and control variables that allow feasible operation for the whole range of uncertain parameters. If feasible operation is achieved, then the process is considered flexible enough. To guarantee feasibility, FA uses the worst-case approach [40] as in the case of robust optimization. In fact, FA and robust optimization share many concepts and methods, even if historically they evolved separately. One key difference, however, is the use of recourse variables (the control variables) to modify the plant's response to particular realizations of the uncertain parameters. While recourse variables are traditionally not used in robust optimization, they are a key concept in stochastic optimization. By combining concepts from these optimization approaches, FA can be used to determine the optimal design characteristics of flexible chemical plants.

The formulation of the flexibility concept and the optimization models proposed in FA may suggest that the approach can be extended to other engineering systems. Indeed, the design and management of flexible infrastructure networks can be properly modeled using the representation of design, state, and control variables from the FA approach. The recourse options provided via control variables can represent adaptation decisions instead. In both cases, flexibility exists to face the uncertainty in external phenomena. Despite these similarities, the concept of flexibility in infrastructure systems (and networks) is more closely related to the concept of Real Options [36, 37], while the definition of flexibility in FA is more similar to the concept of robustness in infrastructure systems (as defined in [13]). The problem of how much should be paid today to have the option to modify the system in the future is a very important notion in flexibility for infrastructure systems that does not exist in FA. Furthermore, the changes implemented using control variables in chemical processes are usually reversible, while modifying infrastructure systems involves a sequential decision process that results in incremental modifications. The sequential nature of the decision process requires a modeling approach that considers the temporal interdependence of the decisions (such as dynamic or stochastic programming) for which the tools developed in FA are insufficient. For these reasons and despite the conceptual similarities, the FA approach is not used in this paper and rather an MS programming model is proposed.

2.1. Numerical representation of flexibility

To formulate and solve the problem of the value of flexibility, it is useful to have a numerical representation of flexibility. Unfortunately, the lack of consensus in the conceptual definition is also present in the formal description. Various authors have proposed an array of indices with different levels of complexity [38, 13, 41, 42] but none has been universally adopted. This study will measure flexibility using the *flexibility vector* defined by Torres-Rincón et al. [16], shown in Equation 1. This formulation explicitly represents two characteristics usually associated with the concept flexibility: the presence of the option to change, and a measurement of the effort required to complete the change. The flexibility vector for the design or operation variable i has two dimensions: the first dimension measures the effort necessary to complete an adaptation as the ratio between the unitary cost of modifying the component without flexibility (without being specifically designed to be adapted) and the unitary cost of performing an adaptation when flexibility was introduced $c_{nf,i}/c_{f,i}$; the second dimension measures the size of the available adaptation space as the ratio between the maximum value the design or operation property can take divided by its initial value $x_{max,i}/x_{0,i}$. Therefore, a system can increase its flexibility by increasing the number of states that can reach through low-cost adaptations, or by reducing the cost of such adaptations.

$$\mathbf{fv}_{i,t} = \left[\frac{c_{nf,i} - c_{f,i}}{c_{f,i}}, \frac{x_{max,t,i} - x_{0,i}}{x_{0,i}} \right] \tag{1}$$

For instance, Figure 1 shows two flexibility vectors of the same magnitude. Vector A, however, has a larger cost component (horizontal axis), while vector B has a larger adaptation space component (vertical axis). Clearly, the expected behavior of both designs under the same external conditions will be different. The vectorial representation of flexibility allows capturing the complex nature of the property.

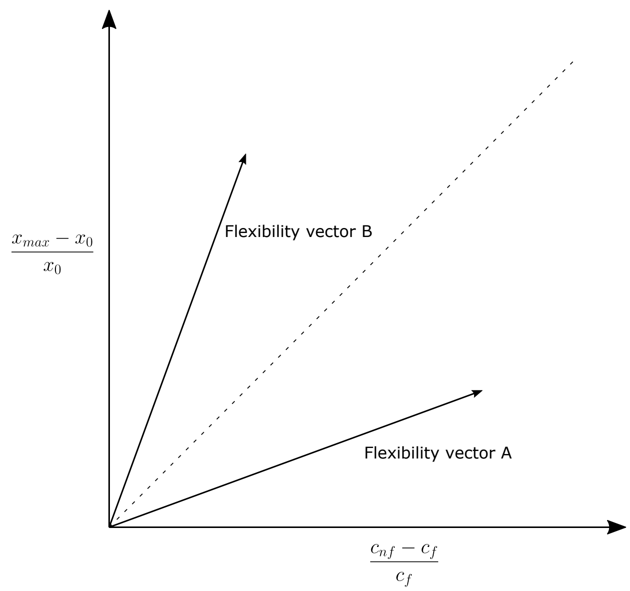


Fig. 1. Flexibility vector representation

2.2. Managing flexibility through policies

The elements of flexibility described previously are deeply interconnected with the physical characteristics of the system. A system is specifically designed and built to be flexible, which affects its morphology. Therefore, future management process to decide the optimal timing and magnitude of adaptations will be constrained by the limitations of the design [13, 41, 7, 31, 4, 16]. The management process is also conditioned by external factors and the system's current state, but also on particular preferences of the stakeholders, technical limitations, user requirements, and regulatory frameworks. For instance, risk-averse stakeholders may prefer to build large systems from the beginning with enough flexibility to perform small adaptations, while stakeholders with a higher risk tolerance may favor smaller initial systems with high flexibility to deploy large adaptations. These complex interactions between external elements and individual preferences can be modeled in the form of *policies*.

Policies are functions that map a set of states to decisions [31, 43]. Depending on the modeling framework, policies can be deterministic or stochastic with variable degrees of complexity. For many applications, however, a simplified framework can be formulated using "*if-then*" conditionals that instruct when to trigger an adaptation process [44]. Under this formulation, the conditional threshold and the size of interventions become the main parameters to be defined. These elements can be used to construct more complex policies. For instance, a more complex policy can be constructed from the if-then conditional by defining the magnitude of the adaptation as the optimal change that minimizes the expected present value of the associated costs and maximizes the expected utility, while constraining the desired performance to remain at certain levels.

Generally, flexibility management requires not one but a complete sequence of decisions. The sequence starts with an initial decision to define the design characteristics of the system, i.e., the initial dimensions and the flexibility (maximum adaptation range and adaptation costs). Once the system is fully commissioned and starts operation, a monitoring process is required to verify the adaptation conditions defined by the policy. The successful implementation of any policy is conditioned on an inspection and monitoring program. When the conditions are met, a decision is made concerning the magnitude of the adaptation. The complexity of this problem lies in that every decision made will affect the future state of the system and, in consequence, the input for the policy in the future. Furthermore, the future state will also depend on uncertain external conditions that may not be stationary (changes in traffic, demand, climate change, etc.) [12]. Thus, the problem of managing flexibility is a classic example of sequential decision problems.

In summary, designing and managing flexible systems requires at least the following tasks (for one flexible component):

1. determine the optimal initial design $\mathbf{x}_0$,
2. determine the optimal initial flexibility level $\mathbf{f}_0$,
3. define the policy π to represent the management decisions,
4. following π, determine the optimal timing $\tau_1, \tau_2, \ldots$, and size $y_{\tau_1}, y_{\tau_2}, \ldots$ of future adaptations.

The model presented in this paper addresses the first two items. Figure 2 presents a diagram that synthesizes this design and management process.

2.3. Sequential decision-making

Typical decision-making models require an agent and an environment. At time t, the agent observes the state of the environment s_t and selects an action a_t according to the policy π. Then, the environment produces a reward r_t for the agent and evolves to a new state s_{t+1} following a probabilist model that depends on the sequence of previous states $s_{1:t}$ and actions $a_{1:t}$ [45]. The

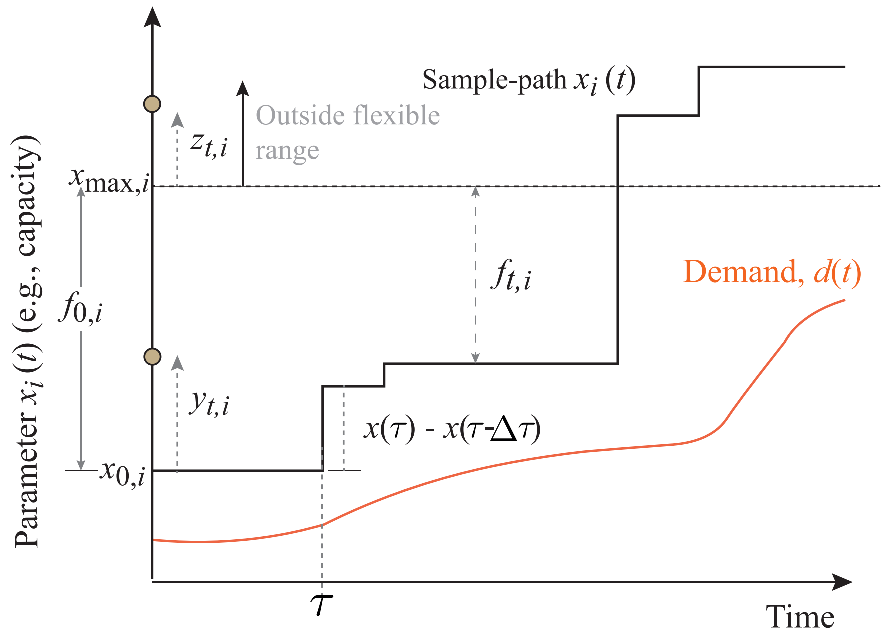


Fig. 2. Description of the main elements for flexible designs

agent observes this new state and selects another action a_{t+1}. If the new state only depends on the previous state and action, then the decision model is referred to as a Markov Decision Process (MDP). While the basic version assumes that the probabilistic model is known and the environment is fully observable, these conditions are not a requirement. For instance, Partially Observable Markov Decision Processes (POMDPs) are a generalized version of MDPs.

Solving a sequential decision problem is not a unique problem; each model may have many different elements that need to be determined, which requires different methods. For instance, a key problem is finding the optimal policy, i.e., the optimal function that selects an action a_t for each pair of action and state history $a_{1:t-1}$, $s_{1:t-1}$. Problems of optimal policy are usually solved using methods such as dynamic programming (DP), approximate dynamic programming (APD), reinforcement learning, genetic algorithms, and Monte Carlo tree searches. A different problem is finding the optimal initial decisions that optimize a certain performance measurement over the system lifetime, i.e., the optimal design that minimizes discounted adaptation costs. In this case, the optimal initial decisions are determined considering future sequences of decisions that may result from a particular decision rule and a probability model, even if the optimal sequence of decisions is not determined. Decision problems focused on finding a robust, optimal initial decision can be solved using MS programming [46, 47, 48].

The problem of flexibility and the value of flexibility as presented before is better suited to be solved using MS programming. The next section presents an overview of the theory behind MSPs and 4 presents the proposed MS model for the problem of flexibility.

3. Stochastic Programming

3.1. Two-stage program with fixed recourse

The simplest stochastic programming model is the linear two-stages model with recourse shown in Equation 2 [48], where the objective is to minimize the costs at the initial stage $t = 0$ and at the recourse stage $t = 1$:

$$\begin{aligned}
\min_{\mathbf{x}_0} \ f = \ & \underbrace{\mathbf{c}^T\mathbf{x}_0}_{\text{Initial stage costs}} \quad + \quad \underbrace{\mathbb{E}_{\boldsymbol{\xi}}[\min \ \mathbf{q}(\omega)^T\mathbf{x}_1(\omega)]}_{\text{Recourse costs}} \\
\text{s.t. } & \mathbf{A}\mathbf{x}_0 = \mathbf{b} \\
& \mathbf{H}(\omega)\mathbf{x}_0 + \mathbf{J}\mathbf{x}_1(\omega) = \mathbf{h}(\omega) \\
& \mathbf{x}_0 \geq 0, \ \mathbf{x}_1 \geq 0
\end{aligned} \tag{2}$$

At the initial stage, the decision $\mathbf{x}_0 \in \mathbb{R}^{n_1}$ is made considering that the first stage data is known: matrix $\mathbf{A} \in \mathbb{R}^{m_1 \times n_1}$ and vector $\mathbf{b} \in \mathbb{R}^{m_1}$. At the second stage, a random event $\omega \in \Omega$ is realized (Ω represents the set of all random outcomes) and the second stage information is revealed: technology matrix $\mathbf{H}(\omega) \in \mathbb{R}^{m_2 \times n_1}$, recourse cost vector $\mathbf{q}(\omega) \in \mathbb{R}^{n_2}$, and right-hand side vector $\mathbf{h}(\omega) \in \mathbb{R}^{m_2}$. Then, the second stage decision $\mathbf{x}_1(\omega) \in \mathbb{R}^{n_2}$, the *recourse decision*, is made. This model is classified as with fixed recourse because the recourse matrix $\mathbf{J} \in \mathbb{R}^{m_2 \times n_2}$ is assumed deterministic. This assumption implies that the effect of the recourse decision $\mathbf{x}_1$ in the constraints is known, which greatly simplifies the solution of the problem and still can be used to model a wide range of real life situations.

Each element in the random matrix and vectors $\mathbf{H}(\omega)$, $\mathbf{q}(\omega)$, and $\mathbf{h}(\omega)$ is a random variable. Therefore, the random vector $\boldsymbol{\xi} = (\mathbf{q}(\omega), \mathbf{h}(\omega), vec(\mathbf{H}(\omega)))$ with support $\Xi \subset \mathbb{R}^d$ can be constructed to represent all the data that depends on the random event ω. By taking the expectation with respect to $\boldsymbol{\xi}$, the estimation of the second stage costs is considering all possible realizations of ω (at least those with non-zero probability).

A key characteristic of stochastic programs (two-stages or multistage) is that the recourse decisions $\mathbf{x}_1$ will be different for each realization of the random event ω. Therefore, if the random parameter distribution is discretized into r intervals, there will be r different $\mathbf{x}_1$ solutions, which is not very useful from a planning perspective. The true value of stochastic programs lies in the initial stage solutions $\mathbf{x}_0$ because they are unique for all trajectories of the random parameter. These decisions are being selected to guarantee that the expected value of future costs (which depend on $\boldsymbol{\xi}$ and $\mathbf{x}_1$) will be minimal. The next section presents how this model can be extended to the multistage case.

3.2. Multistage stochastic programs

Multistage stochastic programming is a modeling framework that generalizes the two-stage problem to a sequence of recourse decisions. In the two-stage problem, the realization of the uncertain parameter becomes known at the second stage and the decision-maker has the option to make a second decision $\mathbf{x}_1$ -the recourse- to adjust the initial decision $\mathbf{x}_0$. When the approach is extended to the multistage case, every recourse decision $\mathbf{x}_t$ will be selected based on the expected value of this decision for stage $t+1$ assuming that every future decision will be optimal [48]. This results in a particular sequence of recourse decisions for every sequence of uncertain parameters.

The general MS programming model can be formulated as follows [47]:

$$\begin{aligned} \min_{\mathbf{x}_1 \in \mathcal{X}_1} \quad f_1(\mathbf{x}_1) + \mathbb{E}[\inf_{\mathbf{x}_2 \in \mathcal{X}_2(\mathbf{x}_1, \boldsymbol{\xi}_2)} \quad f_2(\mathbf{x}_2, \boldsymbol{\xi}_2) + \mathbb{E}[\dots \\ + \mathbb{E}[\inf_{\mathbf{x}_T \in \mathcal{X}_T(\mathbf{x}_{T-1}, \boldsymbol{\xi}_T)} \quad f_T(\mathbf{x}_T, \boldsymbol{\xi}_T)] \dots] \end{aligned} \tag{3}$$

where $\mathbf{x}_t \in \mathbb{R}^{n_t}$ is the vector of decisions variables at time t; $\boldsymbol{\xi}_t \in \mathbb{R}^{M_t}$ is the vector of uncertain parameters at time t (with M_t as large as $n_t + m_t + n_t \times m_t$); functions $f_t(\mathbf{x}_t, \boldsymbol{\xi}_t)$ define the cost of making the decision $\mathbf{x}_t$; and $\mathcal{X}_t$ represent the set of constraints that define the feasibility regions.

By using an expectation functional in the recourse functions, it is being assumed that the decision-maker is risk-neutral. Risk-averse attitudes can be modeled by replacing the expectation functional with risk functionals that consider both the mean and the variance of the random outcome in the optimization process. Functionals such as mean-variance, semi-deviations, weighted mean deviations from quantiles, and average value at risk can be used to generate solutions with limited risk [47]. Nonetheless, for the case of infrastructure networks where the performance is considered long-term and the structural integrity is not being subject to optimization, the expectation functional can be used as long as the number of scenarios considered is large enough for the Law of Large Numbers to apply. Other requirements defined by the decision-maker, e.g. performance levels, can be specified either as hard constraints in the feasibility set $\mathcal{X}_t$ or the deviations can be penalized in the cost function.

In an MSP there may be four uncertain elements: future costs, right-hand side vector, technology matrix, and recourse matrix. The right-hand side vector is usually associated with demand, the technology matrix with the initial decision response, and the recourse matrix with the recourse decisions response. For instance, the technology matrix could represent the productivity of the initial conditions while the recourse matrix could represent the productivity of the recourse decisions, and both affect the system's ability to serve the random demand given by the right-hand side vector. Clearly, the decision of which elements are considered random in the model depends on the type of system and the modeler's assumptions.

The decision on how to model the random parameters depends entirely on the type of problem. Historical data, econometric projections, experts' opinions, all can be used to define the probability distributions of the elements selected to be random. The scenario generation techniques discussed in the following section are used to transform these probability distributions and the filtration structure of the information into discrete approximations in the form of scenario trees that can be used to solve numerically the MSP.

An important property of MSPs is that the information available is represented as σ-algebras $\mathcal{F}_t$. As the stages move forward, the information available increases, which is modeled as a sequence of increasing σ-algebras: $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$. This growing sequence of σ-algebras is known as a *filtration*. This is an important property of the problem because it implies that decisions at each stage x_t can only consider the information available up to that stage [49]. This requirement can be explicitly or implicitly modeled in the constraint set and it is known as *non-anticipativity constraints*.

3.3. Challenges in stochastic programming

The MS programming approach, while useful to capture the presence of uncertainty in planning problems, faces limitations due to its computational complexity. The first limitation is defined by the representation of the uncertain parameters in the model. Regardless of whether the distribution of the random processes is known or if only sample paths are available, this information has to be transformed somehow to be entered as an input to the model. Scenario trees are the usual approach to represent the uncertain elements in MSP and constructing them implies the finite discretization of the original distributions (if they are continuous in the first place).

To properly represent the original distribution a large number of points are required. Furthermore, the nested structure of scenario trees results in exponential growth of the number of scenarios with the number of stages. These two issues (many points and exponential growth) render the problem intractable if a detailed representation of the underlying distribution is desired in a model with many stages. For this reason, methods for the generation and reduction of scenario trees constitute a very active area of research. Høyland and Wallace [50] developed an approach based on minimizing a distance measure (e.g. squared norm) between a predefined statistical property and the statistical property from the approximation. This approach is known as moment matching and has been further expanded in various works (e.g. [51]). However, as shown in [50], matching some moments of the distributions is not enough to guarantee that the solutions of the stochastic program will be similar. For this reason, Heitsch and Römisch [52] developed scenario generation heuristics based on the concept of stability. Stability is a property in stochastic programs that guarantees that the optimal value does not change excessively by changing the scenario tree if both come from the same distribution. The approaches developed by Heitsch and Römisch [52] recursively reduce and bundle scenarios using the concept of filtration distance. The more recent approach developed by Pflug and Pichler [53], based on the concept of nested distance, allows generating the structure of the tree dynamically to meet a prescribed precision.

The second limitation is associated with the solution of the optimization problem. The nested structure found in MSPs severely complicates the problem and special algorithms are sometimes required to provide a solution. Two general approaches are usually recognized in the literature [54, 55, 56]: Primal and Dual decomposition. In the primal decomposition approach, the problem is divided into a master problem for the current stage and sub-problems for the previous stages. During each iteration, cuts are generated to linearly approximate the recourse function and generate a candidate solution $\mathbf{x}^*$ [48]. Examples of primal decomposition algorithms can be found in the works of Birge [57], Ruszczyński [58], and Ruszczyński [59]. In the dual decomposition approach, non-anticipativity constraints are relaxed and moved to the Lagrangian. The division between a master problem and many sub-problems also exists in this approach, but here the sub-problems are connected with scenarios instead of stages. As in the case of primal decomposition, most of the advances came during the late 80s and early 90s with the Progressive Hedging Algorithm [60] and the Diagonal Quadratic Approximation [61]. Most recent efforts have focused on extending these approaches to mixed-integer problems, as in the Branch and Price approach from Lulli and Sen [62].

Nonetheless, not all the MSP required special algorithms to be solved. Alternative approaches exist to exploit particular structures of the problem. For instance, if Block-Separability Recourse (BSR) exists, the MSP can be analyzed as a two-stage problem with aggregate-level decisions being made at the first stage and planning-level decisions happening during the second stage. Commercial solvers are also capable of handling MSPs with linear constraints and non-linear objectives [63] while state-of-the-art solvers can handle problems in the deterministic equivalent formulation [56].

4. Multistage Stochastic Programming Model for Flexibility

4.1. Problem description

An important challenge faced by infrastructure network systems consists of managing changes in demand such as traffic, consumption, or population growth efficiently. This can be accomplished with sporadic adaptations in size, capacity, or any other performance parameter to maintain minimum safety and operational standards. Decisions on the timing and sizes of these interventions should be carefully made to maintain an acceptable level of performance while keeping the costs minimal. In this section, an MSP has been adapted to model this problem.

Consider a network represented as a directed graph $G(E, V)$, where E is the set of edges and V a set of nodes, such that every edge $e_{uv} \in E$ connects nodes u and v with $u, v \in V$. Each edge is associated with a flow $w_t(u, v)$ occurring at time t. This flow can represent traffic, electricity, water, information, etc., moving through typical infrastructure network systems such as highways, power distribution networks, water supply and sewage distribution networks, and telecommunication networks. Simplifying the notation, each edge e can be characterized by a vector of design and operational variables $\mathbf{x}_{t,e} \in \mathbb{R}^{n_e}$ that may change over time (e.g., number of lanes in a highway or capacity in a power transmission network), where n_e represents the number of variables considered for edge e. These potential changes may occur as a result of some external random phenomena (e.g., change in traffic demand or flow), described by event $\omega \in \Omega$, and are represented by the variables $\mathbf{y}_{t,e} \in \mathbb{R}^{n_e}$, in the case of flexible adaptations, and $\mathbf{z}_{t,e} \in \mathbb{R}^{n_e}$ for unplanned adaptations.

An additional element is the flexibility built into each design or operational variable, which is represented in the model as $\mathbf{f}_{0,e} \in \mathbb{R}^{n_e}$ (a decision variable), the maximum range of change allowed by design for $\mathbf{x}_{t,e}$. This is not the flexibility vector $\mathbf{fv}$ presented in Equation 1, but the difference between x_{max} and x_0. For simplicity, $\mathbf{f}_{0,e}$ will be regarded as the flexibility from now on. The flexibility decisions are made at the initial stage, together with the decision concerning initial design and operation properties $\mathbf{x}_{0,e}$, and their purpose is to restrict the flexible adaptations $\mathbf{y}_{t,e}$ that can occur in the future. These modeling decisions allow introducing the problem of balancing three approaches: investing at the initial stage to reduce future adaptation costs, investing at the initial stage to have robust/unchangeable systems, and not investing and risking larger future adaptation costs. This is, deciding the optimal values of $\mathbf{x}_{0,e}$, $\mathbf{f}_{0,e}$, $\mathbf{y}_{t,e}$, and $\mathbf{z}_{t,e}$. It is expected that most solutions will involve a combination of the three strategies.

4.2. Definition of costs

Each decision variable of the model has a cost function associated. These cost functions can represent costs related with activities of design, construction, adaptation, operation, and maintenance of the network edges. The model considers the initial stage costs (design and construction) as known, while the costs associated with future activities (adaptation, operation, and maintenance) can be affected by uncertainty. This distinction is represented in the following cost functions: for the initial stage, a_e is the cost function of building edge e with the initial design parameters $\mathbf{x}_{0,e}$, and b_e is the cost function of adding features to network edge e that facilitate future changes, i.e., the cost of adding flexibility to the system. For example, b_e can represent the additional construction costs for a larger foundation in a building to facilitate future expansions. For the recourse stages, $c_e(\mathbf{y}_{t,e}, \omega)$ is the cost function of adapting the network edge parameters by $\mathbf{y}_{t,e}$ units within the flexible range $\mathbf{f}_{0,e}$ (i.e., the initial design value of variable i plus the sum of all the flexible adaptations over the lifetime must be less or equal than the initial value plus the flexibility $x_{0,e}(i) + \sum_{t=1}^{T} y_{t,e}(i) \leq x_{0,e}(i) + f_{0,e}(i)$), and $d_e(\mathbf{z}_{t,e}, \omega)$ is the cost function of performing an adaptation $\mathbf{z}_{t,e}$ outside the flexible range. This option is included in the model to represent the fact that any infrastructure network can be modified if enough resources are invested; however, this does not mean that the alternative will be efficient. Additional costs related with operation are the flow cost $q(w_e, \omega)$, the operation and maintenance (O&M) cost $g_e(\mathbf{x}_{t,e}(i), \omega)$, and the function of revenue received from exploiting the network $h_e(\omega)$.

The parameters of these cost functions depend on, first, the type of function used for the representation and, second, the procedure performed to obtain the data. For many applications, a linear or convex function is enough to represent the required behavior. However, to represent specific conditions, e.g. economies of scale [7], concave functions may be needed. The values of these parameters can be obtained from autoregressive models, reports in the literature, experts' opinions, industry surveys, and sensitivity analyses. If the model considers random costs, then the cost function parameters become random variables and additional parameters for their distributions must be defined (using historical data, available literature, surveys, etc.).

In the model proposed, all the cost functions can be represented using convex functions. If a typical linear cost representation is used, then the only cost parameter is the unitary cost coefficient. In this case, the only requirement is that the unitary costs of functions d_e (unplanned adaptations) must be larger than the unitary costs of functions c_e(planned adaptations); otherwise, the model will never assign value to the flexible option. If a different convex function is used, then it must be guaranteed that $d_e(x) > c_e(x)\ \forall\ x > 0$. This does not mean that flexibility will always be preferred due to the additional cost be of introducing flexibility at the initial stage. However, if the condition is not met, then flexibility will never be a viable option from a cost standpoint.

4.3. MSP problem formulation

The elements described in the previous sections are combined in the following objective function for an MSP to analyze flexible infrastructure networks:

$$
\begin{aligned}
\min_{\mathbf{x}_0,\mathbf{f}_0} \quad & \underbrace{\sum_{e\in E}\sum_{i=1}^{n_e} a_e(\mathbf{x}_{0,e}(i))}_{\text{Initial design costs}} + \underbrace{\sum_{e\in E}\sum_{i=1}^{n_e} b_e(\mathbf{f}_{0,e}(i))}_{\text{Flexibility introd. cost}} \\
& + \mathbb{E}_{\boldsymbol{\xi}_1}\Bigg[\min \gamma_1 \Bigg(\underbrace{\sum_{e\in E}\sum_{i=1}^{n_e} c_e(\mathbf{y}_{1,e}(i),\omega)}_{\text{Flexible adaptations cost}} + \underbrace{\sum_{e\in E}\sum_{i=1}^{n_e} d_e(\mathbf{z}_{1,e}(i),\omega)}_{\text{Unplanned adaptations cost}} + \underbrace{\sum_{e\in E} q(w_{1,e},\omega)}_{\text{Flow cost}} \\
& + \underbrace{\sum_{e\in E}\sum_{i=1}^{n_e} g_e(\mathbf{x}_{1,e}(i),\omega)}_{\text{O\&M cost}} \pm \underbrace{h(\omega)}_{\text{Exploitation cost/revenue}} \Bigg) + \dots \\
& + \mathbb{E}_{\boldsymbol{\xi}_T}\Bigg[\min \gamma_T \Bigg(\sum_{e\in E}\sum_{i=1}^{n_e} c_e(\mathbf{y}_{T,e}(i),\omega) + \sum_{e\in E}\sum_{i=1}^{n_e} d_e(\mathbf{z}_{T,e}(i),\omega) + \sum_{e\in E} q(w_{T,e},\omega) \\
& + \sum_{e\in E}\sum_{i=1}^{n_e} g_e(\mathbf{x}_{T,e}(i),\omega) \pm h(\omega) \Bigg) \Bigg] \dots \Bigg]
\end{aligned}
\tag{4}
$$

where i indexes the design and operational parameters being considered for each edge e, n_e is the number of design and operational parameters, and γ_t is the discount factor. To represent the limitations and special conditions associated with the management of flexibility and the optimal flow distribution inside the network, the following constraints are formulated:

First, the constraints for flow conservation between converging and diverging links at a node k are:

$$\sum_{v} w_{(k,v)} - \sum_{u} w_{(u,k)} = \phi_k(\omega) \qquad \forall u, v, k \in V, \tag{5}$$

where $\phi_k = 0$ if k is a transshipment node and $\phi_k \neq 0$ if k is either a supply or a demand node.

Second, the constraints for the minimum and maximum flow that can move through each link e are:

$$\ell_{lower,t,e} \leq w_{t,e} \leq \ell_{upper,t,e} \qquad t = 1 \dots T, \forall e \in E \tag{6}$$

where the lower and upper limits $\ell_{lower}, \ell_{upper}$ are part of the design and operational variables vector $\mathbf{x}_{t,e}$ and may have a flexibility value associated (can be adapted).

Third, the constraints that control the trigger of an adaptation process:

$$V(\mathbf{x}_{t,e}, \boldsymbol{\xi}_t) \geq \pi_{t,e} \qquad t = 1 \dots T, \forall e \in E \tag{7}$$

where the performance function V provides a quality measure of the design and operation variables $\mathbf{x}_{t,e}$ in comparison with the pressure caused by the random parameters $\boldsymbol{\xi}_t$. This set of constraints establishes that the performance measure must be larger than a certain value $\pi_{t,e}$ given by the management policy. In this way, the constraints defines the criteria to make a change. For example, it can define the maximum ratio flow/capacity in an edge before it has to be expanded.

Fourth, the constraints for the maximum planned adaptation in a design or operation variable in an edge are:

$$\sum_{j=1}^{t} \mathbf{y}_{j,e}(i) \leq \mathbf{f}_{0,e}(i) \qquad t = 1, \ldots, T, \quad i = 1, \ldots, n_e, \quad \forall e \in E \tag{8}$$

These constraints establish that the sum of the total planned changes for a design/operation variable i in an edge e must be, at most, the flexibility built for that variable i and edge e.

Fifth, the constraints to update the state variables of an edge in accordance with the available actions are:

$$\mathbf{x}_{t,e}(i) = \mathbf{x}_{t-1,e}(i) + \mathbf{y}_{t-1,e}(i) + \mathbf{z}_{t-1,e}(i) \quad t = 1, \ldots, T, \quad i = 1, \ldots, n_e, \quad \forall e \in E \tag{9}$$

Other constraints can be added depending on the nature of the network, or to properly characterize the adaptations (e.g. non-negativity constraints restrict adaptations to expansions; integrality changes the nature of the adaptations from modular to continuous). There are non-anticipativity constraints implicitly formulated in the other constraints to restrict the knowledge available to make a decision at each stage; this is, at each time t, the decisions can only depend on the information available up to t.

The decision stages t are selected considering the frequency of the decision process. This frequency may not coincide with the frequency of the random processes, in which case one decision stage may encompass many random process periods. For instance, it is not expected for an infrastructure network to be expanded every year, even if the demand is reported every month. In such a case, the decision stage t can consider the average, the maximum, or a quantile of the demand periods clustered in t. It is also not required for the decision stages $t_1, t_2, \ldots$ to be equal in size. Considerations regarding the life-cycle of the network and the reliability of the uncertain data may result in a higher frequency of decisions during the first years of the network. It must also be considered that increasing the frequency of the decision stages also increases the size of the problem.

In brief, it is a modeling decision that depends on the purpose of the model and the experience of the modeler.

5. Approximate Solution of MSPs

5.1. Criteria for approximating the solution

Solving a multistage stochastic program using the original distribution of the uncertain parameters $\boldsymbol{\xi}_t$ may not be feasible in many real applications because the set of possible paths (realizations of the demand) is infinite. For this reason, several methods have been developed to approximate the distribution of the random phenomena [52, 49, 64]. Overall, the available methods focus on: minimizing some measure of the distance between the original and the approximated distribution (e.g. Wasserstein distance) [52, 65], matching moments between the distributions [50], and generating samples (e.g. Monte-Carlo sampling [66]. As a result, the original distribution becomes a set of scenarios (each with an associated probability), usually organized in a tree structure. This scenario tree represents the increasing finite filtrations of available information, as shown in Figure 3.

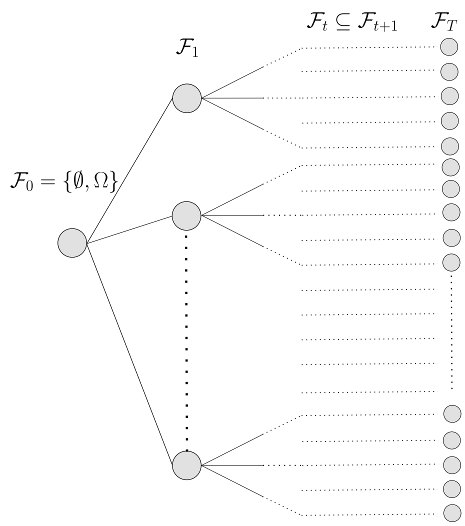


Fig. 3. Scenario tree representation of increasing finite filtrations

The accuracy of the stochastic program solution highly depends on the quality of the resulting discretization [67]. However, it is important to keep in mind that the main objective is to optimize the quality of the solution obtained by the stochastic program and not necessarily to produce an optimal discretization of the original distribution. This can be achieved by having stability and low bias in the solution [68]. Stability means that the variation in the optimal value is minimal among different scenario trees (obtained from the same distribution). Low bias refers to a small gap between the real (unobtainable) and the approximated solution. The most direct approach to achieve both stability and low bias is to simulate a large number of scenarios [69, 68]. A trial and error procedure is usually necessary to achieve these objectives with the smaller scenario tree as possible.

5.2. Discretization method by clustering

As explained before, it is generally not possible to solve an MSP using the real distribution of $\boldsymbol{\xi}_t$; instead, an approximation is required. Typical approximate distributions are represented in the form of scenario trees. In this paper, the discretization process is performed by combining Monte-Carlo sampling and a k-medoids clustering algorithm (see Figure 4).

The methodology requires listing the properties of the random process to be simulated, which can include values such as the mean and the standard deviation, or additional shape parameters. Then, the time mission is divided in a finite set of time steps (stages) $t = 1, \ldots, T$, and a large number N of random trajectories are generated (see Figure 4(a)). At the first time stage $t = 1$, the coefficient k_1 is defined to represent the number of clusters in which the trajectories will be aggregated; this coefficient is known as the *branching factor*. Then, the trajectories are combined into k_1 clusters using a k-medoids clustering algorithm. At the next time stage, the trajectories grouped inside each cluster are further divided into k_2 clusters using the same algorithm. This process is repeated until the end of the time mission when a total of $K = \prod_{t=1}^{T} k_t$ clusters are generated. This process results in a symmetric scenario tree.

This method requires a large number of trajectories per cluster to avoid clustering single trajectories. Considering that the number of scenarios grows exponentially with the number of stages, the number of required trajectories may be difficult to generate for large scale applications. While it works well for problems with a number of scenario paths of order 10^5 to be clustered in trees with a number of scenarios of order 10^3, problems of larger scale may require more complex approaches. For instance, a larger number of clusters can be used at the initial stages to obtain a finer discretization due to the importance of the prediction at these early stages (Figure 4(b)). Fortunately, in most cases, flexibility problems in infrastructure networks usually involve a small

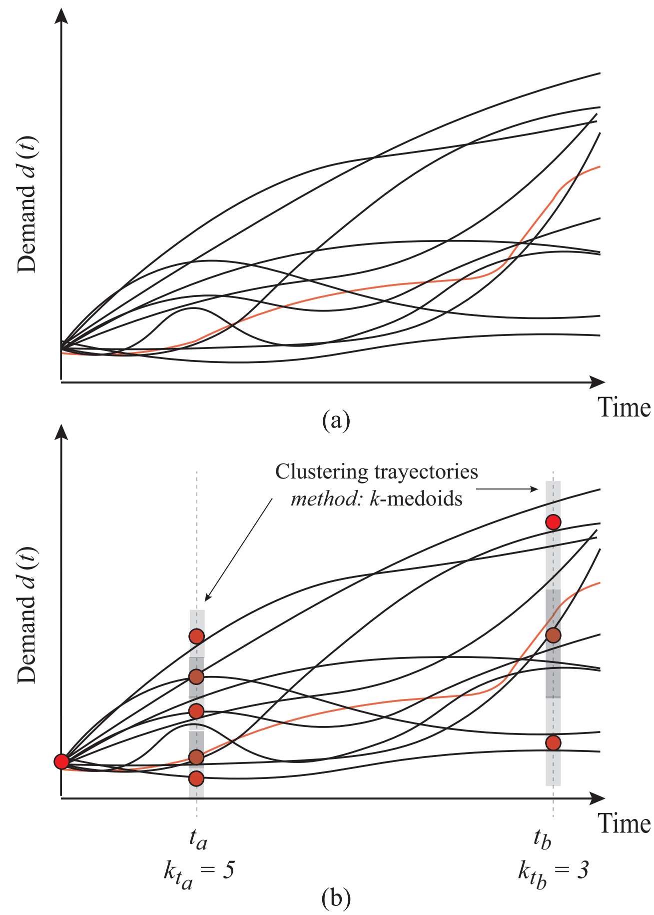


Fig. 4. (a) realization of system path trajectories; (b) clustering of sample paths at different stages.

number of variables and stages due to the limited number of characteristics that can be adapted and the low frequency of changes.

5.3. Approximated flexibility MSP model

The approximation of the distribution of $\boldsymbol{\xi}_t$ using a scenario tree can be combined with the original flexibility MS approach ((4)) to produce an approximated MSP that can be solved with numerical solvers. By replacing the probability model with an approximate, simpler, model with finite support [65], the original filtration becomes a finite filtration with a tree structure (scenario tree). This approximation leads to the formulation of an equivalent deterministic program, as shown in Equation 10:

$$
\begin{aligned}
\min_{\mathbf{x}_0,\mathbf{f}_0} \quad & \sum_{e\in E}\sum_{i=1}^{n_e} a_e(\mathbf{x}_{0,e}(i)) + \sum_{e\in E}\sum_{i=1}^{n_e} b_e(\mathbf{f}_{0,e}(i)) \\
& + \underbrace{\sum_{k=1}^{s} p_k}_{\text{Scenario probability}} \left[\gamma_1 \left(\sum_{e\in E}\sum_{i=1}^{n_e} c_{e,k}(\mathbf{y}_{1,e,k}(i)) + \sum_{e\in E}\sum_{i=1}^{n_e} d_{e,k}(\mathbf{z}_{1,e,k}(i)) + \sum_{e\in E} q_k(w_{1,k}) \right.\right. \\
& \left. + \sum_{e\in E}\sum_{i=1}^{n_e} g_{e,k}(\mathbf{x}_{1,e,k}(i)) \pm h_{1,k} \right) + \dots \\
& + \gamma_T \left(\sum_{e\in E}\sum_{i=1}^{n_e} c_{e,k}(\mathbf{y}_{T,e,k}(i)) + \sum_{e\in E}\sum_{i=1}^{n_e} d_{e,k}(\mathbf{z}_{T,e,k}(i)) + \sum_{e\in E} q_k(w_{T,k}) \right. \\
& \left.\left. + \sum_{e\in E}\sum_{i=1}^{n_e} g_{e,k}(\mathbf{x}_{T,e,k}(i)) \pm h_{T,k} \right)\right]
\end{aligned}
\tag{10}
$$

where s indicates the number of scenarios and p_k is the individual probability of each scenario. Scenarios are understood as a complete path from the root node to a leaf node in the scenario tree. This formulation of the objective function clearly shows that a particular solution is obtained for each scenario k except for the initial stage variables ($\mathbf{x}_0$ and $\mathbf{f}_0$) which are the same for all scenarios.

Formulating the MSP in this form requires to explicitly develop the non-anticipativity constraints. The objective of these constraints is to ensure that scenarios with the same history up to stage t are indistinguishable [47].

$$
\begin{aligned}
& \mathbf{x}_{t,e,j}(i) = \mathbf{x}_{t,e,k}(i) \\
& \mathbf{y}_{t,e,j}(i) = \mathbf{y}_{t,e,k}(i) \\
& \mathbf{z}_{t,e,j}(i) = \mathbf{z}_{t,e,k}(i) \qquad \forall\, j,k \text{ for which } \boldsymbol{\xi}_{t,j} = \boldsymbol{\xi}_{t,k},\ t = 1,\dots,T, \quad i = 1,\dots,n_e, \quad \forall e \in E
\end{aligned}
\tag{11}
$$

To illustrate the non-anticipativity conditions, Figure 5 shows a fragment of a scenario tree. The figure shows that Scenario s_{1,t_r} and Scenario s_{2,t_r} are identical up to stage t_q, same as Scenario s_{3,t_r} and Scenario s_{4,t_r}. Furthermore, the four scenarios are indistinguishable up to stage t_p. The restrictions in Equation 11 ensure that the decision variables comply with the non-anticipativity conditions given by the scenario tree of the random process.

The procedures described in Sections 5.1-5.3 to formulate and solve the proposed MSP are summarized in the chart presented in Figure 6.

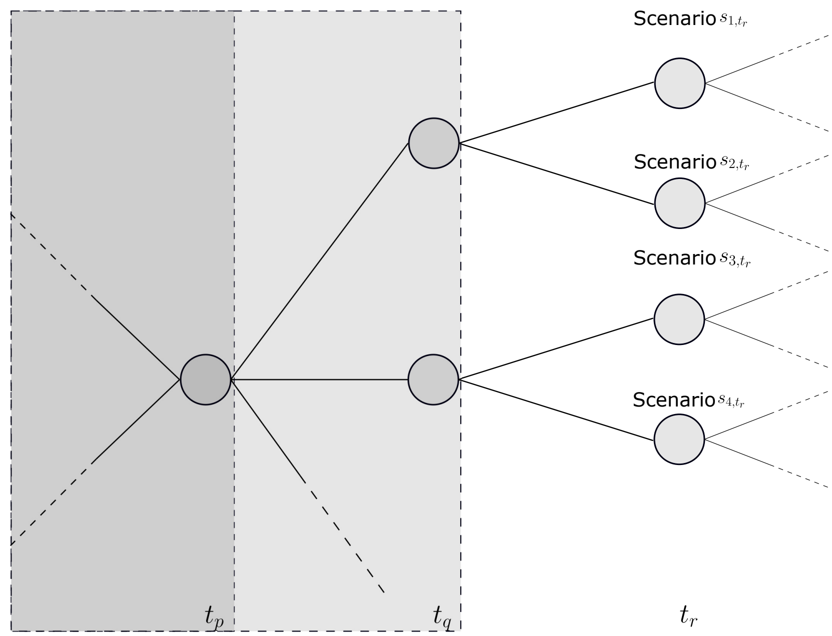


Fig. 5. Non-anticipativity conditions in scenario tree

5.4. Model scalability

The scalability of the proposed model can be analyzed for two elements: the scenario tree generation algorithm and the solution method. First, the scenario generation algorithm is based on the general approach described by Dupačová et al. [63] where a set of scenario paths are generated according to a probabilistic model (based on, for instance, historical data series), a scenario tree structure is predefined according to some heuristic (e.g. detailed branching for early stages and coarse branching for later stages), and a clustering algorithm is applied to the scenario paths based on some dissimilarity measure. The scenario path generation process, even for the multivariate case, scales linearly in the worst case, and it is never regarded in the literature as a worrisome source of computational complexity. The clustering algorithms, however, do not scale as well due to the required comparisons between all the data points (distance or dissimilarity measures).

The k-medoid algorithms used in this paper are the Partitioning Around Medoids (PAM) and the Clustering LARge Applications (CLARA) implementations in Matlab©. The CLARA algorithm is used to cluster the first 2-3 stages of the scenario tree where the number of data points is large (> 5000). This algorithm is of order $O(k^3)$ [70] and can be used for large data sets to be bundled in

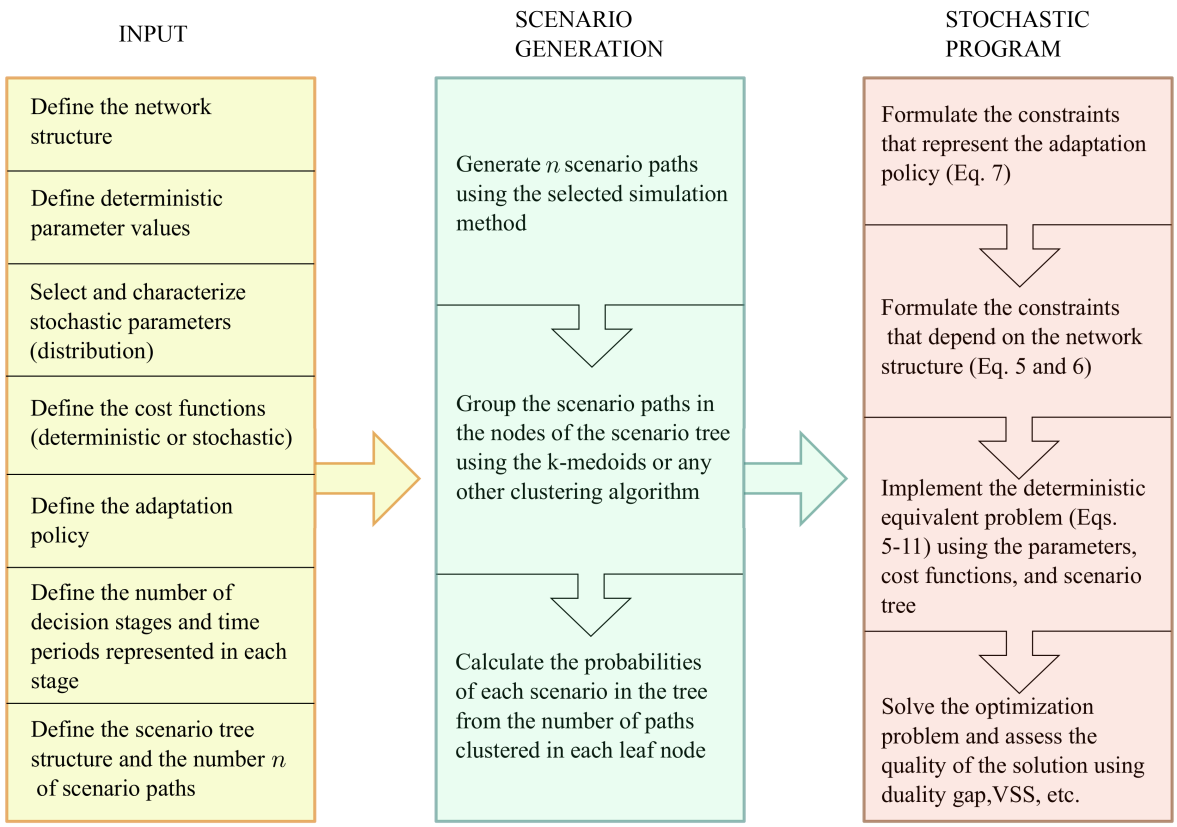


Fig. 6. General procedure to formulate and solve proposed MSP model

a small number of clusters. The subsequent stages require clustering progressively smaller number of points in an also small number of clusters, which can be done using the PAM algorithm of order $O(k(n-k)^2)$. While these clustering algorithms work well for a wide range of data set sizes and a limited number of branching factors, the structure of the scenario tree requires that the algorithms have to be executed an increasing number of times at each stage. This is, the number of calls for the clustering algorithms grows exponentially with the number of stages. Therefore, the scalability of the scenario generation procedure depends on both the scalability of the clustering algorithms and the size of the scenario tree.

The second element that affects the scalability of the model is the solution method. As explained in Section 3.3, solving an MSP is challenging due to the considerable number of variables, the interdependence between them, and possible non-linear constraints. While special methods have been developed to solve MSPs, in some cases commercial solvers are enough to obtain a solution in reasonable times for problems with hundreds of thousands of variables. One of these cases is when the problem has a linear (or non-linear) objective and linear constraints [63]. By introducing the scenario tree representation to formulate the deterministic equivalent problem, the result is a large scale linear (or non-linear) optimization problem that can be solved efficiently by commercial solvers. The model presented in Section 5.3 has linear constraints and the cost functions can be properly represented as linear or quadratic functions for many applications. Therefore, commercial solvers can be used to solve the proposed model, as is shown in the example in Section 6.

Nonetheless, the exponential growth in the number of variables as the number of stages increases poses a significant limitation to the frequency of the decision process. This limitation can be circumvented as the nature of infrastructure systems makes monthly or even annual adaptation plans unrealistic due to the time and resources required to complete an adaptation. The decision stages for infrastructure adaptation problems can encompass years and, in consequence, it is possible to model the complete or a considerable portion of the system's lifetime with a limited number of stages.

In addition to the linearity of the constraints and the control to the scenario tree explosion by limiting the number of stages, the proposed model has two additional properties that reduce its computational complexity. The first property is the diagonal structure in the constraint matrices. Limiting the dependence of the decision variables to their immediate predecessor by adding state variables results in sparse matrices that are much faster to solve [46]. The second property is block-separable recourse. This property allows transforming an MSP into a two-stage problem by separating the aggregate level decision variables from the planning level variables. The aggregate level decisions can represent capacity expansion decisions while the planning level decisions can refer to the use of this capacity [48]. If the aggregate level decisions do not depend on the planning level decisions, then the former can be moved to the first stage while the latter are placed on the second stage. Specifically, block-separable recourse exists if i) at stage t the cost functions can be written as:

$$f_t(\mathbf{u}_t) + g_t(\mathbf{v}_t) \tag{12}$$

where $\mathbf{u}_t$ represents the aggregated level decisions, $\mathbf{v}_t$ the planning level decisions, and f_t and g_t are the respective cost functions. And ii) if the constraints can be written as:

$$\begin{pmatrix} T_t & 0 \\ S_t & 0 \end{pmatrix} \begin{bmatrix} \mathbf{u}_{t-1} \\ \mathbf{v}_{t-1} \end{bmatrix} + \begin{pmatrix} W_t & 0 \\ 0 & D_t \end{pmatrix} \begin{bmatrix} \mathbf{u}_t \\ \mathbf{v}_t \end{bmatrix} = \begin{bmatrix} \mathbf{a}_t \\ \mathbf{b}_t \end{bmatrix} \tag{13}$$

where T_t and S_t are submatrices of the technology matrix, W_t and D_t are submatrices of the recourse matrix, and $\mathbf{a}_t$ and $\mathbf{b}_t$ are subvectors of the right-hand side vector. This equation clearly shows that both current aggregate and current planning level decisions depend only on previous aggregate level decisions.

The model proposed in Equations 4-11 is block-separable for the following reasons: i) the cost functions are clearly segregated between the aggregate level variables ($\mathbf{x}_t$, $\mathbf{y}_t$, $\mathbf{z}_t$, and $\mathbf{f}_0$) and the planning variables $\mathbf{w}_t$; ii) the constraints in Equations 5-9 and 11 are not enforcing any dependence between the set of aggregate decisions $\mathbf{x}_t$, $\mathbf{y}_t$, $\mathbf{z}_t$, and $\mathbf{f}_0$ and the decisions $\mathbf{w}_{t-1}$. Constraints 6 and 7 do generate a dependence between $\mathbf{w}_t$ and $\mathbf{x}_{t-1}$ but this relationship is included in the definition of block-separability.

In summary, the proposed model has advantageous characteristics in its structure that can be exploited to vastly reduce its computational complexity and allow the use of commercial solvers to find a solution. The main bottleneck happens in the scenario tree generation procedure due to the combinatorial nature of the process. Despite these advantages, the number of variables in the problem still grow exponentially with the number of stages, and even if the MSP can be transformed into a large scale linear problem and a simple binary tree structure is used, having more than ~ 22 stages implies hundreds of millions of variables.

6. Numerical Example

6.1. Case description

This section presents an example of modeling flexibility in network design and operation using the proposed MSP. The purpose of this example is threefold: i) to show how the model can be used to make decisions when managing flexible networks; ii) to study how the presence of flexibility affects the cost performance of the network; and iii) to identify the elements in the model that have the highest impact in the assessment of the value of flexibility.

The network used in the example is shown in Figure 7; it is a generic network that may describe telecommunication, energy, water distribution, or a transportation network. This example only considers one element in the random parameters vector $\boldsymbol{\xi}_t = \delta_t$ to represent the demand that enters the network through node A and propagates throughout the network until it reaches node F, where it exits. This demand is modeled as a stochastic process defined by:

$$\delta(t) = \beta_1 t + e^{\beta_2 t}\ \beta_3 t\ \sin(\beta_4 t) + B(t) \tag{14}$$

where $\beta_1, \beta_2, \beta_3, \beta_4$ are normally distributed random variables and $B(t)$ is a Wiener process. Table 1 shows the values of the parameters used in the example. Figure 8 shows five realizations of this process.

Table 1. Random demand process parameters

Parameter	Distribution(μ, σ^2)
β_1	$\mathcal{N}(450,\ 1.26 \times 10^4)$
β_2	$\mathcal{N}(-0.02,\ 2.5 \times 10^{-5})$
β_3	$\mathcal{N}(100,\ 625)$
β_4	$\mathcal{N}(0.9,\ 5.06 \times 10^{-2})$

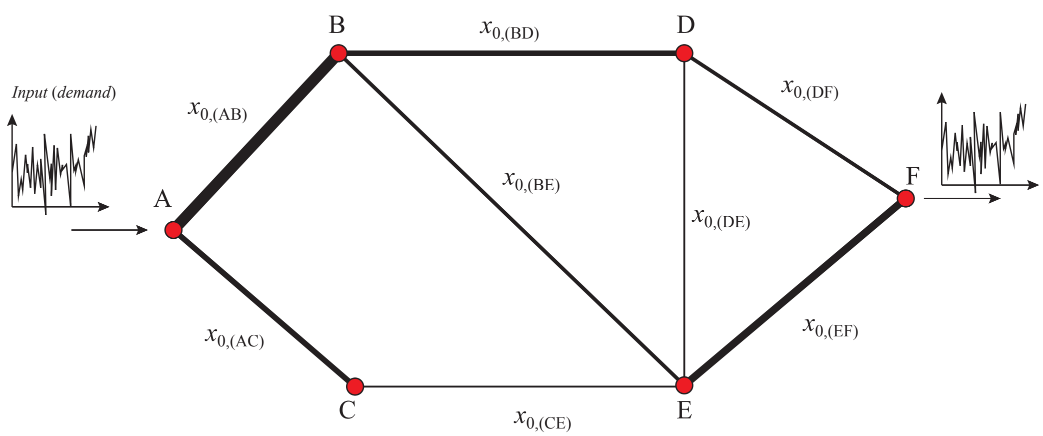


Fig. 7. Network used in the numerical example.

The design and operation variables vector $\mathbf{x}_{t,e}$ of each link in the network only contains the link capacity $x_{t,e}$. The initial capacity is notated as $x_{0,e}$, and together with the flexibility range $f_{0,e}$ define the initial stage decisions. The maximum capacity that can be reached by flexibility for each link at any time is given by $x_{max,e} = x_{0,e} + f_{0,e}$. However, the system can be modified beyond

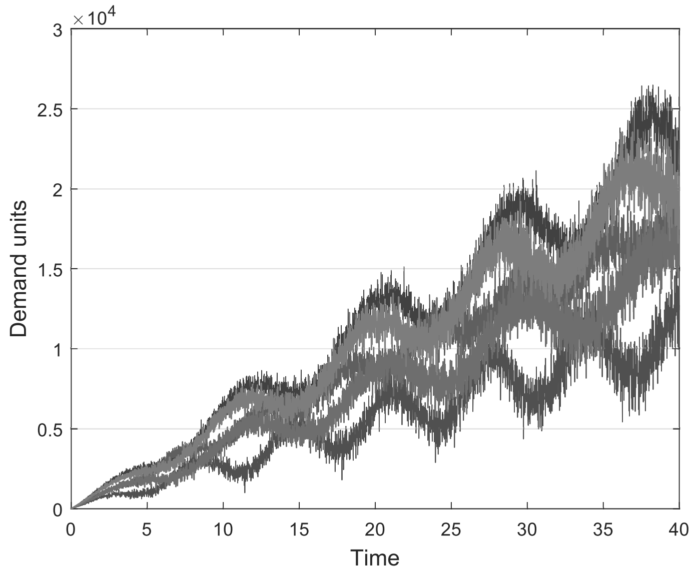


Fig. 8. Realizations of the stochastic process

that value but at a larger cost. This may happen, for instance, due to larger times required to modify the system, larger material and labor costs that result from last-minute contracts, and the related disturbances and damages caused by the unexpected partial or total cease of operation. In the example, only capacity expansions are considered.

The time mission of the network is $T = 40$ years starting from the date of commissioning. To reduce the size of the problem, it is assumed that adaptations can only occur at specific times: $[6, 12, 18, 24, 32, 40]$ (except for the sensitivity analysis in Section 7.1.2). This limitation allows downsizing the scenario tree, which highly impacts the computational performance. The reduction can be justified by the fact that infrastructure networks, and in general infrastructure systems, cannot be changed frequently in most of their dimensions without incurring high costs.

The model uses two families of cost functions. The initial design cost functions a_e, flexible adaptation cost functions c_e, and the unplanned adaptation cost functions d_e (see Equation 10) are assumed linear of the form $a_e = \alpha_0\ \mathbf{x}_{0,e}$, $c_e = \alpha_{fa}\ \mathbf{y}_{t,e}$, and $d_e = \alpha_{ia}\ \mathbf{z}_{t,e}$, equal for all the edges. The linear cost assumption reduces the comparison to a straightforward relationship between the cost coefficients. Quadratic functions are used to represent the costs of introducing flexibility of the form $b_e = \alpha_f\ f_{0,e}^2$. This non-linear increasing function is used to consider the effect of dis-economies of scale that may exist in systems that cannot be expanded indefinitely without restrictions, such as most of infrastructure networks.

Coefficient α_0 is equal to 5 cost units per capacity unit. Coefficients α_{fa} and α_{ia} vary for the analyses shown in Section 7.2. Operation and maintenance costs (O&M) g_e is assumed equal to 2% of α_0 per unit of capacity installed, except for the last analysis. Flow costs q_e are function of O&M costs g_e as shown in Table 2.

Table 2. Flow costs summary

Link	Flow Cost
A-B	$0.05g_e$
A-C	$1.2q_{AB}$
B-D	$1.1q_{AB}$
B-E	q_{AB}
C-E	$1.2q_{AB}$
D-E	$1.3q_{AB}$
D-F	$1.1q_{AB}$
E-F	$1.2q_{AB}$

The flow costs vary between links to represent an existing minimal cost route and increase the stability of the solution by avoiding a random allocation of initial capacity and flexibility.

6.2. Discretization of the demand space

The first task is the discretization of the demand space using the sampling and clustering algorithm presented in Section 5.2. This requires defining first the scenario tree branching factors which vary between [4 4 2 2 2 2] (256 scenarios) and [4 4 4 4 2 2] (1024 scenarios). Then, the algorithm generates between 15000 and 25000 demand trajectories according to Equation 14 depending on the total number of scenarios. The clustering algorithm processes these inputs to generate a scenario tree whose nodes contain the demand values at the decision stages $[6, 12, 18, 24, 32, 40]$ years, with an associated probability that depends on the number of trajectories clustered in each node.

6.3. MSP solution

To obtain the optimal network initial conditions, $x^*_{0,e}$ and $f^*_{0,e}$ for every edge e in the network, it is necessary to solve the multistage stochastic problem described by Equation 10, subject to the restrictions presented in Equations 5 to 11. The model assumes that every time a flexible adaptation $y_{t,e}$ occurs, the flexibility built into the system at the initial stage $f_{0,e}$ is reduced until its depletion. Once this point is reached only unplanned adaptations $z_{t,(u,v)}$ can be deployed. Finally, adaptations (i.e., increments incapacity) can only occur at $t \in \{0, 6, 12, 18, 24, 32, 40\}$.

The evolution of the network capacity is controlled by a policy $\pi(\xi_t, x_{t,e})$ based on a safety criteria that establishes that the capacity of every edge at time t should be such that $x_{t,e} \geq SF\ w_{t,e}$, with $w_{t,e}$ the flow moving through link e, and SF some safety factor. This formulation keeps the constraints in Equation 7 linear. Additional flow conservation constraints are formulated for each network node according to the topology shown in Figure 7.

The numerical results were obtained using the CPLEX 12.8 API for Matlab® 2017b and the YALMIP toolbox as algebraic interpreter [71] with an Intel®Core™ i5-5200 2.20GHz processor. An average run for 1000 scenarios takes approximately 100 seconds of processing time, where 95% of the time is spent in the scenario tree generation algorithm and less than 5% in solving the optimization program.

The solver used (CPLEX) automatically selects the barrier optimizer due to the large size and sparsity of the problem. This algorithm stops iterating when the primal and dual solution are complementary, i.e., when a sum of products between the primal and dual solutions is smaller than a predefined tolerance. The tolerance of the complementary used in this numerical example is defined as 1^{-8}.

7. Network Design for Flexibility: Results

Solving the MSP defined by Equations 5 to 11 a results in an optimal (minimal) value of the total present costs C^*_{total}, an optimal value of the initial capacity $x^*_{0,e}$ for each link e, an optimal value of the flexibility $f^*_{0,e}$ for each link e, and optimal values each variable $y^*_{t,e,k}$ and $z^*_{t,e,k}$, that represent the evolution process of each network link, for every stage t and scenario k. Figure 9 shows three realizations of the demand process and the respective evolution of the whole network capacity.

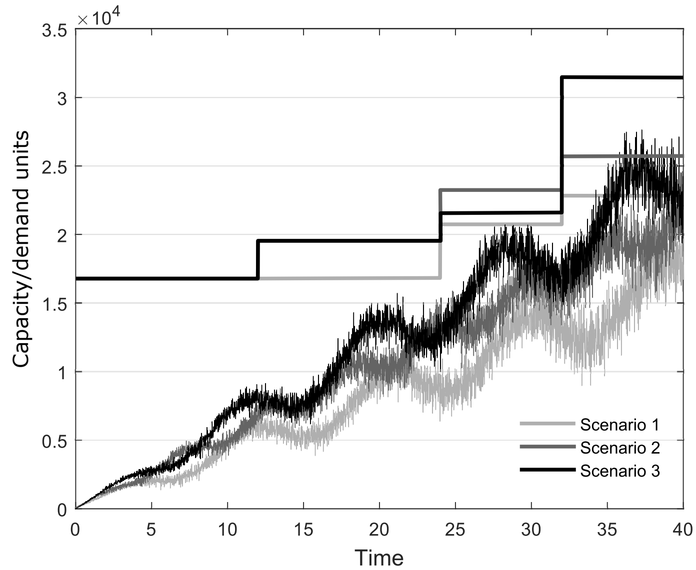


Fig. 9. Network capacity evolution for three demand scenarios

This figure illustrates one of the main properties of MSPs: The optimal solution for the initial capacity, i.e. the initial stage decision, is the same for all scenarios while the recourse decisions that expand the system are particular for each scenario. By averaging the optimal costs of all the system's future responses, the model makes an initial decision that is optimal in average for all scenarios.

Considering that the solutions of the recourse decisions $y^*_{t,e,k}$ and $z^*_{t,e,k}$ are generated for each scenario, they are not suited for the comparisons intended in the forthcoming analyses. Only the initial solutions $x^*_{0,e}$ and $f^*_{0,e}$ are the same for all scenarios. For this reason, only these solutions, together with the optimal costs C^*_{total}, will be considered in the analysis of results.

7.1. Numerical stability of the model

7.1.1. Stability verification of the demand tree generation

Stability in the generation of the demand tree is essential for the accuracy of the model. If the program is not stable, the solution can vary considerably between scenario trees. To verify the stability of the solution, three scenario tree structures were generated using different branching factors. These factors determine how many nodes grow from each node at each time stage and, ultimately, the number of scenarios. Furthermore, the demand values in the nodes of the scenario tree vary due to the Monte-Carlo generation procedure. To address these sources of variability, the MSP was solved in batches of increasing size (from 2 solutions up to 40 solutions) and the resulting optimal costs were averaged for each batch. Figure 10 presents the coefficient of variation, CV, of the optimal costs for each batch size, and for three sets of branching factors (scenario tree structures).

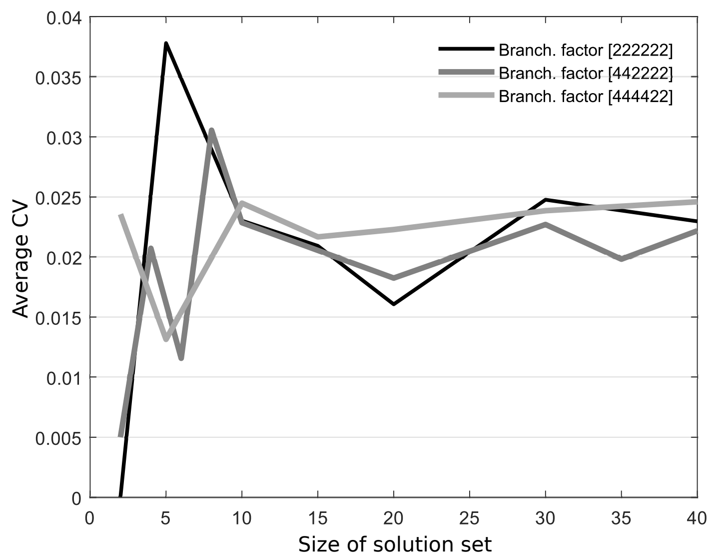


Fig. 10. Solution variability as function of number of scenarios and runs

The scenario tree structures compared in Figure 10 were selected starting by the smallest possible symmetrical tree [2 2 2 2 2 2]. Additional nodes were increasingly added to the initial stages to improve the quality of the short-term predictions. The largest scenario tree considered, [4 4 4 4 2 2], showed similar behavior with larger trees but with considerably lower computational times.

Figure 10 shows that there is variability due to the number of solutions being considered and due to the instability in the solution. As the size of the batch increases, the first source decreases and only instability remains. It is observed that the optimal costs are relatively stable with a $CV = 2.5\%$. It is also observed that a larger number of scenarios reduces the number of solutions required to reach a stable CV (branching factor [4 4 4 4 2 2]). This result is expected as a larger number of scenarios represent better the underlying distribution, achieving higher stability as a

Table 3. Sensitivity analysis results

Decision stages	Optimal value	$f^*_{0,AB}$	$x^*_{0,AB}$	Processing time(s)
4	361619	2546	14787	45
5	340954	2657	13341	62
6	334664	2614	12798	85
7	339480	2661	12894	95
8	329768	2793	11175	92
9	341655	2664	12618	90
10	341035	2693	12961	110

result. Generally, only the optimal costs are compared in stability analyses and not the optimal solutions due to the typical flat objective functions found in MSPs [72].

7.1.2. Planning horizon discretization into decision stages

The results from the previous section assumed that the planning horizon is divided into 6 decision stages. This section presents a sensitivity analysis to determine how this modeling decision affects the MSP results.

To this end, the planning horizon is divided into $4, 5, \ldots, 10$ decision stages, and the resulting MSP is solved. In the cases when the planning horizon is not a multiple of the number of stages, the initial stages encompass fewer years than later stages, but the purpose is to cover the entirety of the planning horizon as uniformly as possible. The scenario trees are designed to have the same number of scenarios (2^{10}) which requires using variable branching factors. Therefore, the discretization of the probability space for the intermediate stages will not be equal for all cases, with a difference of a factor of 2 at most in the number of scenario tree nodes for similar time instants. For instance, for 4 decision stages, there are 64 nodes representing year 20, while for 5 stages there are 32 nodes at year 16. Table 3 shows the optimal value, the optimal flexibility solution $f^*_{0,AB}$ for link AB, the optimal initial capacity solution $x^*_{0,AB}$, and the processing time obtained for each number of decision stages.

The results in Table 3 show that when the number of stages that divide the planning horizon is 5 or larger, only the processing times are affected. For 4 decision stages, the model solutions seem to be significantly different, beyond what could be explained by the random variation present in the solutions for the remaining number of stages. This tendency was found in multiple executions. This difference could be explained by the relationship between the frequency of the required adaptations and the frequency of the decision stages. As shown in Figure 9, the time between adaptations can vary between 8 and 12 years on average, with longer times during the first half of the planning horizon. This means that during the system's lifetime only 3 or 4 adaptations are expected. Therefore, when the planning horizon has 4 decision stages or fewer, it can be expected that an adaptation will take place at every stage and that the magnitude of these adaptations (and the initial capacity) will be larger to accommodate the changes in the demand during such long time periods.

In summary, the discretization of the planning horizon into decision stages can affect the solution when there is insufficient opportunity to use the recourse to respond to the changes in the external conditions.

7.1.3. Out-of-sample behavior of the model

The verification of the out-of-sample behavior consists in solving the optimization program with a particular tree and using the optimal solution, i.e., $x^*_{0,e}$ and $f^*_{0,e}$, as the initial solution for MSPs with different scenario trees of the same size and generated from the same stochastic process. Figure 11 compares the optimal costs obtained in the baseline program (horizontal line), with the optimal costs found in 10 out-of-sample programs. This figure shows that 70% of the out-of-sample costs have a difference of 1.2% or less with the base case, while the remaining 30% has, at most, a difference of 3.3%. The results from the stability analysis and the out-of-sample analysis suggest that even if there is some instability because the scenario tree is not infinite, it is under control and the program exhibits both *in* and *out* sample stability, which is a requirement for low bias and general stability.

7.1.4. Value of the stochastic solution

A second key element in the analysis of MSPs is the value of the stochastic solution (VSS). This metric compares how much is gained by solving the stochastic program instead of solving a deterministic program where the scenario tree is replaced with the expected value of the random process. To calculate the VSS, first, the expected value problem is solved and the optimal solution is used as the initial solution for an MSP. The optimal costs obtained for this new program are

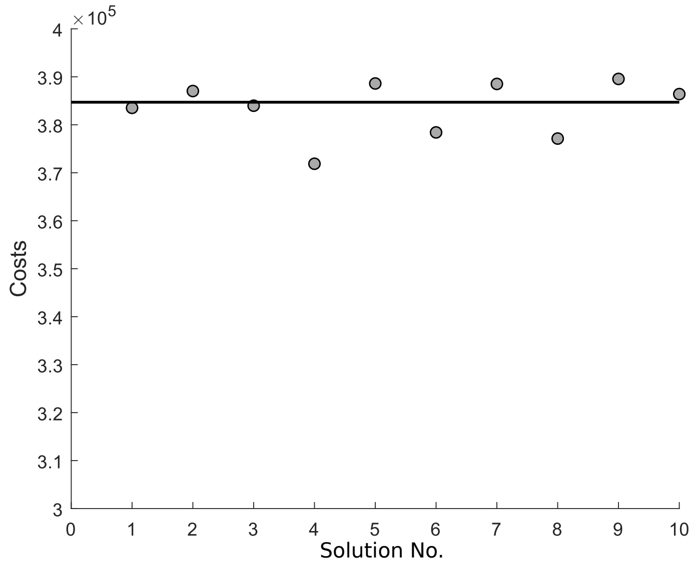


Fig. 11. Out-of-sample solution comparison

known as the *expected value* solution (EV). The VSS is obtained as the difference between the original MSP optimal costs and the EV.

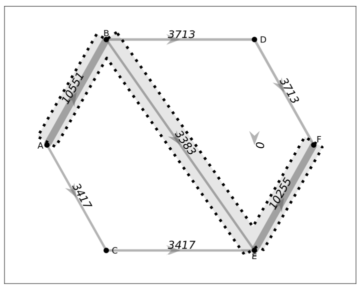
Using the scenario tree structure regarded as the most stable from the analysis in Section 7.1.1, [4 4 4 4 2 2], generating 50 scenario trees and averaging the VSS, it was obtained a value that is 16% of the average optimal costs. This result is not negligible and suggests that the VSS is large enough to justify solving a stochastic program instead of the much simpler expected value problem.

7.2. The value of flexibility

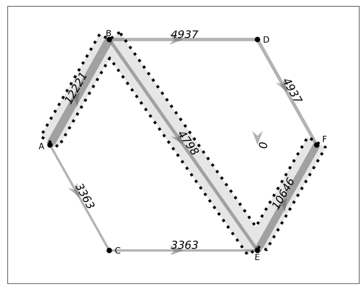
The third element of the analysis focuses on the effects of the cost functions on the value of flexibility. Different relationships between α_0, α_{fa}, and α_{ia} are considered to determine the effect on: i) the optimal initial capacity $x^*_{0,e}$, ii) the optimal flexibility $f^*_{0,e}$, and iii) the optimal costs.

Figures 12 and 13 show, respectively, the evolution of the optimal solution for the capacity $\mathbf{x}^*_{0,e}$ and the flexibility $\mathbf{f}^*_{0,e}$ in each link for different ratios α_{fa}/α_0. The critical route is highlighted with the dotted lines and light shadow. As expected, the larger the cost of future adaptations α_{fa} compared with the cost of building the required capacity from the beginning α_0, the less flexibility is introduced, and the system relies more on the initial capacity to serve the demand. This means that reducing α_0 or increasing α_{fa} decreases the value associated with flexibility, and the system manager is willing to pay less to have it in the network.

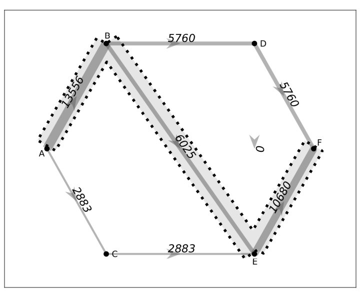
It is interesting to observe in Figures 12 and 13 that the optimal solution for the initial capacity $\mathbf{x}^*_{0,e}$ results in larger values for the links in the minimum cost route, while the optimal solution for flexibility $\mathbf{f}^*_{0,e}$ is similar for all the links in the network, excluding DE. This suggests that the short and medium-term demand is mostly met by the minimum cost route until a point where it is less expensive to use the flexibility built in other links to expand them and reroute part of the demand. Besides, the quadratic cost function used for the introduction of flexibility discourages the approach of adding too much flexibility into one link.



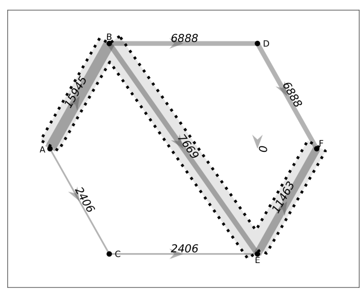
(a) $\alpha_{fa}/\alpha_0 = 0.5$



(b) $\alpha_{fa}/\alpha_0 = 1$



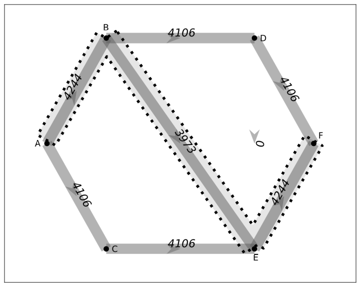
(c) $\alpha_{fa}/\alpha_0 = 1.4$



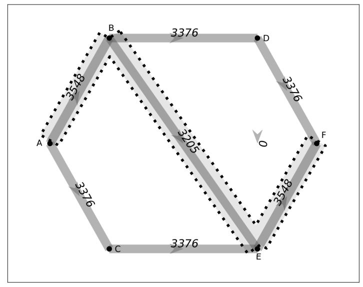
(d) $\alpha_{fa}/\alpha_0 = 2$

Fig. 12. Initial capacity evolution for different ratios α_{fa}/α_0

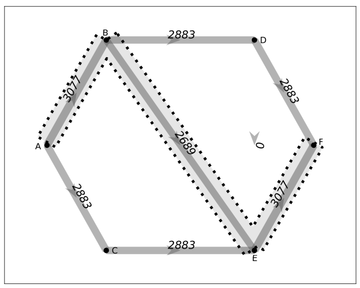
Figure 14 compares the optimal costs obtained for different ratios α_{fa}/α_0. The figure includes in the comparison the flexibility vectors **fv** (see Equation 1) for each link of the network. The horizontal component of the vector **fv** depends on the ratio between planned and unplanned adap-



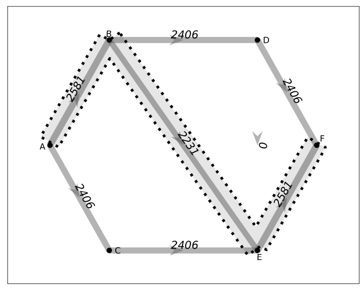
(a) $\alpha_{fa}/\alpha_0 = 0.5$



(b) $\alpha_{fa}/\alpha_0 = 1$



(c) $\alpha_{fa}/\alpha_0 = 1.4$



(d) $\alpha_{fa}/\alpha_0 = 2$

Fig. 13. Initial flexibility for different ratios α_{fa}/α_0

tations, while the vertical component depends on the ratio between the flexibility $\mathbf{f}_{0,e}$ and the initial design/operation conditions $\mathbf{x}_{0,e}$. It is first observed the increment in optimal costs by increasing the ratio α_{fa}/α_0; expectedly, increasing the flexible adaptation costs increases the overall costs. As shown in Figures 12 and 13, the model reacts by the decreasing the flexibility $\mathbf{f}_{0,e}$ and increasing the initial capacity $\mathbf{x}_{0,e}$; nonetheless, the model cannot completely offset the increments in costs because flexibility is still a cost-efficient strategy. This behavior is replicated in the flexibility vectors, whose vertical component decreases as α_{fa}/α_0 increases (the horizontal component decreases by definition).

It is interesting to observe that the decrement is not uniform for all links. For $\alpha_{fa}/\alpha_0 = 0.5$, the vectors are divided into two groups (vector for link DE has a zero vertical component): i) links AB, EF, which are part of the critical route, and ii) the remaining. Links AB, EF are key for the entrance and exit of the demand to the network, and the model privileges initial capacity, which results in **fv** with a lower vertical component. Because the demand is free to flow through all the remaining links, the program can assign more flexibility. As α_{fa}/α_0 increase to 1 and 1.4, the vertical component of the **fv** keeps decreasing non-uniformly, resulting now in three distinct groups: i) key input and output links of the critical route AB, EF, with the lowest vertical component; ii) internal transit links BD, BE, EF whose vertical component keeps decreasing as flexibility becomes less attractive and it is replaced by adding more initial capacity; and iii) the less used links AC, AE, which start transporting flow later on the network life-cycle and can keep relying on flexibility. This tendency is clearly observed when $\alpha_{fa}/\alpha_0 = 2$ and the vertical component of **fv** decreases considerably for all links except AC, AE.

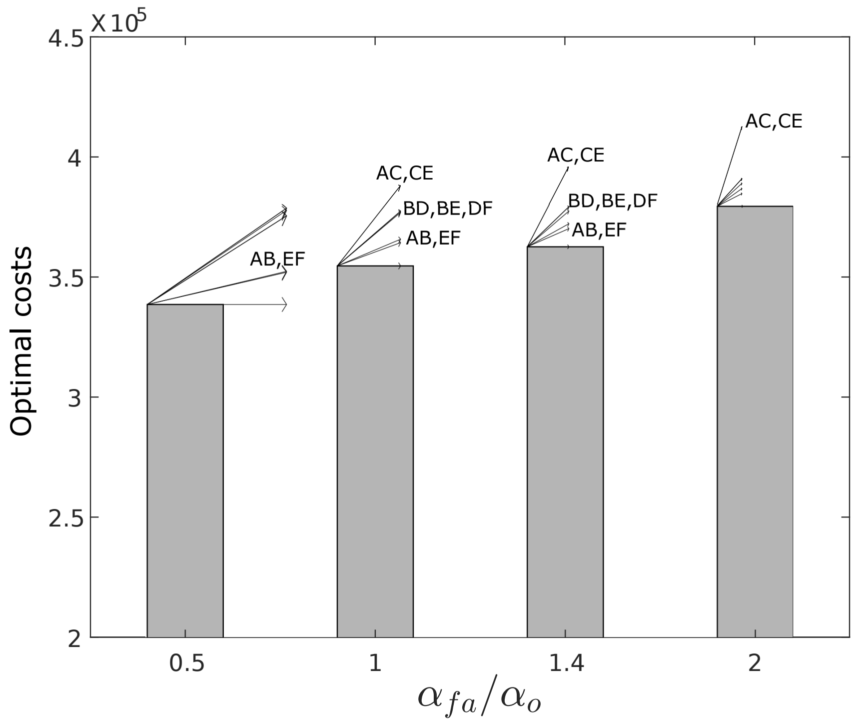


Fig. 14. Optimal costs and flexibility vectors **fv** for different ratios α_{fa}/α_0

7.3. Effect of the discount rate

A similar analysis was conducted to examine the effect of the discount rate on the optimal values $x^*_{0,e}$ and $f^*_{0,e}$. Figure 15 compares the capacity and flexibility of link AB (other links follow a similar tendency) for three discount rates: $dr = \{3\%, 5\%, 8\%\}$ as a function of α_{fa}/α_0. While the values of flexibility are mostly unaffected by the change in the discount rate dr, the initial capacity presents a clear decreasing tendency with an increment in the rate. In this case, larger discount rates make future adaptations (inside or beyond the flexible range) more attractive in the medium and long-term.

The decision of adding flexibility depends both on the cost of adding flexibility, which is incurred at the initial stage and it is not affected by the discount rate, and the costs of performing adaptations, which are discounted. Increasing the discount rate reduces the present value of all future cost making both planned and unplanned adaptations more attractive, or at least moves forward the time stage where they are the optimal option. As a result, the short and medium-term demand is served using planned adaptations (flexibility) and long-term demand is served with unplanned adaptations, in detriment of the initial capacity. Therefore, it is expected that for typical ranges of the discount rate ($\sim 3-10\%$) flexibility will be considered valuable. However, it could be argued that for unrealistically large discount rates the model would rely entirely on unplanned adaptations, except for the initial demand, abandoning completely the option of flexibility.

7.4. Costs of adaptation beyond the flexible range

In this analysis, the effect of the cost of unplanned adaptations α_{ia} is determined by using different ratios α_{fa}/α_{ia} and α_0/α_{ia} to solve the MSP. Figure 16 shows a similar tendency as Figure 15, with the optimal initial capacity and flexibility decreasing as the cost of unplanned adaptations α_{ia} decreases, and the initial capacity being affected by the increasing discount rate, which further bolsters the cost-effectiveness of unplanned adaptations. However, in this case, a reversing tendency is detected for the 8% discount rate. Upon closer inspection, it is revealed that there is a point where the short-term demand must be met before unplanned adaptation can take place. At this point it does not matter how much further α_{ia} can be reduced; there is a minimum demand that must be met.

For lower discount rates it was cost-effective to use more flexibility to meet this short-term demand; however, for large discount rates, the initial cost of flexibility plus the adaptation cost

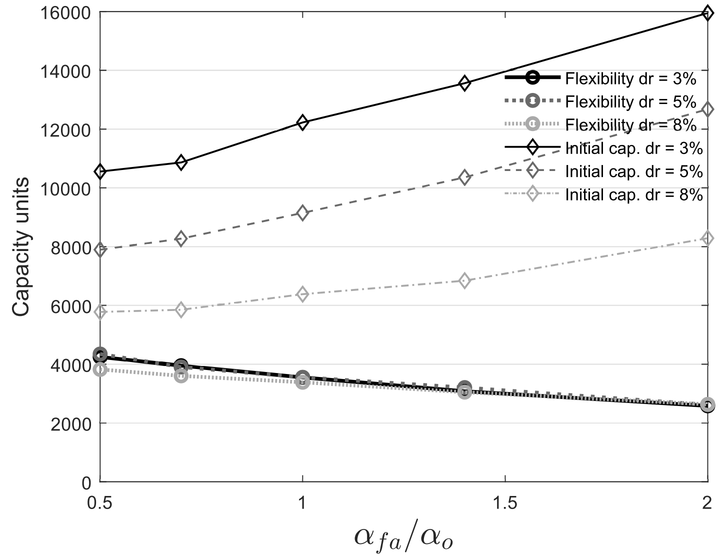


Fig. 15. Installed capacity and flexibility in link AB as function of ratio α_{fa}/α_0 and discount rate

is not as cost-effective as just relying on the initial capacity for the short-term and unplanned adaptations for the medium and long-term. This could explain the monotonic decreasing tendency for flexibility while the initial capacity behaves asymptotically. This means that reducing the cost of unplanned adaptations can only reduce the dependence on initial capacity up to the point where there is enough capacity to attend the short-term demand that could not be served in any other way. This point is dictated by the initial demand and its growth during the first decision period.

A final analysis was conducted to examine the effect of the unitary O&M costs on the optimal initial capacity and flexibility. By increasing the unitary O&M costs from 2% of α_0 to 10% of α_0, the optimal costs increase approximately 20%. The optimal initial capacity in all links is reduced by approximately 12% which is compensated by a similar increment in the optimal installed flexibility. If the O&M costs are further increased to 16% of α_0, the optimal initial capacity decreases another 9% with a similar increment in the optimal flexibility. Increasing the O&M costs discourage large system configurations in the short and medium-term, which makes flexibility more valuable.

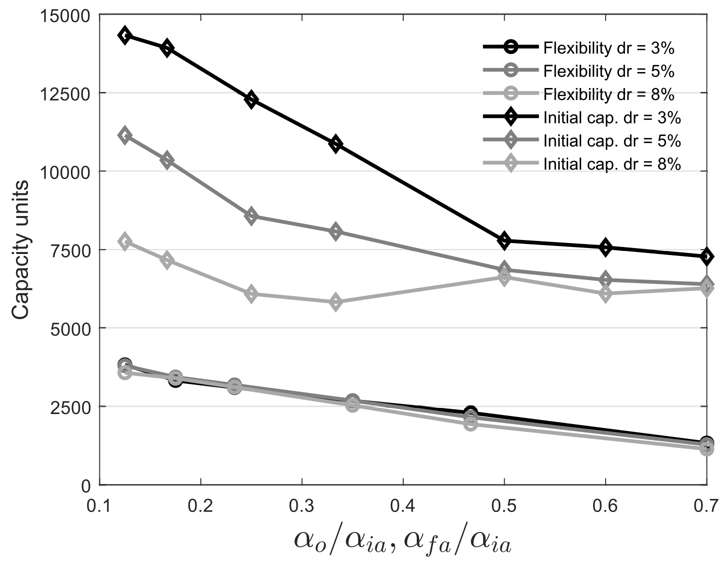


Fig. 16. Installed capacity and flexibility in link AB as function of ratios α_{fa}/α_{ia}, α_0/α_{ia} and discount rate

8. Summary and Conclusions

This paper presented a multistage stochastic programming model for the design and management of flexible infrastructure networks. The model considers the initial design characteristics of each network link for different design elements as an initial decision that can be modified in future stages by implementing planned and unplanned adaptations to face random external conditions. By modeling the flexibility built at the initial stage as a decision variable that has an associate cost and that limits future planned adaptations, the program can determine the optimal configuration that minimizes costs. This formulation also provides an indirect measure of the value of flexibility by finding the optimal *amount* of flexibility that is cost-effective to introduce.

An approximate solution method was implemented using Monte-Carlo simulation and k-medoids clustering to generate the discrete representation of the stochastic process (scenario tree). The deterministic equivalent program was formulated and tested for a generic network with one input and one output node for a stochastic demand. The numerical solution showed stability for reasonably sized scenario trees and a not negligible VSS of around 16%. Furthermore, the results showed how the value of flexibility is affected by the discount rate and the ratios between all the costs involved, not just the cost of deploying planned adaptations. If the ratios α_{fa}/α_0, and α_{ia}/α_0 are too large, the system will prefer to meet the short and medium-term demand with the initial capacity, lowering the value of flexibility. As these ratios increase, the adaptations are further pushed into the future, especially for low discount rates. In the case where the ratio α_{fa}/α_{ia} is large, unplanned adaptations become cost-effective, also decreasing the value of flexibility. In contrast, for a given α_{fa}, as the ratio α_{ia}/α_0 grows, flexibility becomes more desirable. These examples show how the value of flexibility constantly evolves due to complex decisions that happen at different time instants.

The proposed model and the procedure implemented in the numerical example assumed that all decision and state variables are continuous. A mixed-integer formulation may require specific algorithms (e.g. primal decomposition) to find a solution efficiently. Furthermore, all cost functions were assumed convex, which is a reasonable assumption for most applications, but if concave functions are required, the solution procedure may not efficient nor may guarantee to find a solution. The scenario generation procedure can be extended to the multivariate case (for independent variables) but the combinatorial nature of the problem restricts the number of decision stages than can be modeled. The formulation and solution of the MSP is also restricted by the exponential scalability. The expectation functional used in the model (Equation 4) enforces a risk-neutral approach that may not be appropriate for some applications. Further works should consider a formulation of the model with risk-averse functionals.

9. Acknowledgements

The financial support from the Vice-rectory for Research at Universidad de los Andes and from the Regional Council of ‘Pays de la Loire’ within the framework of the BUENO 2018-2021 research program (Durable Concrete for Offshore Wind Turbines) is gratefully acknowledged.

List of symbols and abbreviations

a_e Cost function of building edge e with intial parameters $\mathbf{x}_{0,e}$

α_0 Unitary cost of building initial design $\mathbf{x}_{0,e}$

α_{fa} Unitary cost of performing flexible adaptation $\mathbf{y}_{t,e}$

α_{ia} Unitary cost of performing unplanned adaptation $\mathbf{z}_{t,e}$

b_e Cost function of adding $\mathbf{f}_{0,e}$ flexibility to edge e

c_e Cost function of changing edge e by $\mathbf{y}_{t,e}$

CV Coefficient of variation

d_e Cost function of changing edge e by $\mathbf{z}_{t,e}$

γ_t Discount factor at time t

G Directed graph

dr Discount rate

e Edge/link in directed graph G

$\mathbf{f}_{0,e}$ Flexibility range of design/operation parameters for edge e

$w_{t,e}$ Flow thought edge e at time t

fv Flexibility vector

g_e Operation and maintenance cost function for edge e

$h(\boldsymbol{\xi}_t)$ Revenue function

MDP Markov decision process

MS multistage stochastic

MSP multistage stochastic program

O&M Operation and Maintenance

p_k Proability of scenario k

π Policy function

q_e Flow cost function for edge e

$\boldsymbol{\xi}_t$ Vector of random, external parameters at time t

ROA real options analysis

VSS Value of stochastic solution

$\mathbf{x}_{t,e}$ Vector of design/operation parameters at time t for edge e

$\mathbf{x}_{0,e}$ Vector of initial design/operation parameters for edge e

$\mathbf{y}_{t,e}$ Vector of flexible adaptations at time t for edge e

$\mathbf{z}_{t,e}$ Vector of unplanned adaptations at time t for edge e

10. References

[1] T. R. Browning, E. C. Honour, Measuring the Life-Cycle Value of Enduring Systems, Systems Engineering 11 (3) (2008) 187–202, doi:\bibinfo{doi}{10.1002/sys.20094}.

[2] J. Vieira, M. Cunha, Systemic approach for the capacity expansion of multisource water-supply

systems under uncertainty, Journal of Water Resources Planning and Management 142 (10), doi:\bibinfo{doi}{10.1061/(ASCE)WR.1943-5452.0000668}.

[3] X. Zhang, S. Mahadevan, S. Sankararaman, K. Goebel, Resilience-based network design under uncertainty , Reliability Engineering and System Safety 169 (2020) 364–379, doi: \bibinfo{doi}{10.1016/j.ress.2017.09.009}.

[4] M. Sánchez-Silva, Managing Infrastructure Systems through Changeability, Journal of Infrastructure Systems 25 (1) (2019) 04018040, doi:\bibinfo{doi}{10.1061/(ASCE)IS.1943-555X.0000467}.

[5] J.-M. Lozano, S. Zuluaga, M. Sánchez-Silva, Developing flexible management strategies in infrastructure: The sequential expansion problem for infrastructure analysis (SEPIA), Reliability Engineering and System Safety 200 (2020) 106951, doi:\bibinfo{doi}{10.1016/j.ress.2020.106951}.

[6] I. Roefs, B. Meulman, J. H. Vreeburg, M. Spiller, Centralised, decentralised or hybrid sanitation systems? Economic evaluation under urban development uncertainty and phased expansion, Water Research 109 (2017) 274–286, doi:\bibinfo{doi}{10.1016/j.watres.2016.11.051}.

[7] M.-A. Cardin, M. Ranjbar-Bourani, R. De Neufville, Improving the Lifecycle Performance of Engineering Projects with Flexible Strategies: Example of On-Shore LNG Production Design, Systems Engineering 18 (3) (2015) 253–268, doi:\bibinfo{doi}{10.1002/sys.21301}.

[8] R. De Neufville, S. Scholtes, Flexibility in Engineering Design, 2011.

[9] M. Mortazavi-Naeini, G. Kuczera, L. Cui, Application of multiobjective optimization to scheduling capacity expansion of urban water resource systems, Water Resources Research (50) (2014) 4624–4642, doi:\bibinfo{doi}{10.1002/2013WR014569}.

[10] C. C. Fraga, J. Medellín-Azuara, G. F. Marques, Planning for infrastructure capacity expansion of urban water supply portfolios with an integrated simulation-optimization approach, Sustainable Cities and Society 29 (2017) 247–256, doi:\bibinfo{doi}{10.1016/j.scs.2016.11.003}.

[11] T. Erfani, K. Pachos, J. J. Harou, Real-Options Water Supply Planning: Multistage Scenario Trees for Adaptive and Flexible Capacity Expansion Under Probabilistic Climate

Change Uncertainty, Water Resources Research 54 (2018) 5069–5087, doi:\bibinfo{doi}{10.1029/2017WR021803}.

[12] M. V. Chester, B. Allenby, Toward adaptive infrastructure: flexibility and agility in a non-stationarity age, Sustainable and Resilient Infrastructure 4 (4) (2019) 173–191, doi:\bibinfo{doi}{10.1080/23789689.2017.1416846}.

[13] A. M. Ross, D. H. Rhodes, D. E. Hastings, Defining Changeability : Reconciling Flexibility , Adaptability , Scalability , Modifiability , and Robustness for Maintaining System Lifecycle Value, Systems Engineering 11 (3) (2008) 246–262, doi:\bibinfo{doi}{10.1002/sys}.

[14] J. H. Saleh, G. Mark, N. C. Jordan, Flexibility : a multi-disciplinary literature review and a research, Journal of Engineering Design 20 (3) (2009) 307–323, doi:\bibinfo{doi}{10.1080/09544820701870813}.

[15] O. Špačková, D. Straub, Long-term adaption decisions via fully and partially observable Markov decision processes, Sustain Resilient Infrastruct 2 (1) (2017) 37–58, doi:\bibinfo{doi}{10.1080/23789689.2017.1278995}.

[16] S. Torres-Rincón, D. F. Villarraga, M. Sánchez-Silva, Conceptual and Numerical Analysis of Flexibility in Infrastructure Systems, Journal of Infrastructure Systems 26 (2) (2020) 04020012, doi:\bibinfo{doi}{10.1061/(ASCE)IS.1943-555X.0000546}.

[17] A. Angelus, E. Porteus, S. Wood, Optimal Sizing and Timing of Modular Capacity Expansions, 2000.

[18] H. Luss, Operations Research and Capacity Expansion Problems: A Survey, Operations Research 30 (5) (1982) 907–947, doi:\bibinfo{doi}{10.1287/opre.30.5.907}.

[19] J. A. Van Mieghem, Capacity Management, Investment, and Hedging: Review and Recent Developments, Manufacturing & Service Operations Management 5 (4) (2003) 269–302, doi:\bibinfo{doi}{10.1287/msom.5.4.269.24882}.

[20] C. Martínez-Costa, M. Mas-Machuca, E. Benedito, A. Corominas, A review of mathematical programming models for strategic capacity planning in manufacturing, International Journal of Production Economics 153 (2014) 66–85, doi:\bibinfo{doi}{10.1016/j.ijpe.2014.03.011}.

[21] A. S. Manne, Capacity Expansion and Probabilistic Growth, Econometica 29 (4) (1961) 632–649, doi:\bibinfo{doi}{10.2307/1911809}.

[22] S. Rajagopalan, Capacity expansion with alternative technology choices, European Journal of Operational Research 77 (1994) 392–403, doi:\bibinfo{doi}{10.1016/0377-2217(94)90405-7}.

[23] S. Ahmed, N. V. Sahinidis, An Approximation Scheme for Stochastic Integer Programs Arising in Capacity Expansion, Operations Research 51 (3) (2003) 461–471, doi:\bibinfo{doi}{10.1287/opre.51.3.461.14960}.

[24] K. J. Singh, A. B. Philpott, R. Kevin Wood, Dantzig-Wolfe Decomposition for Solving Multistage Stochastic Capacity-Planning Problems, Tech. Rep. 5, doi:\bibinfo{doi}{10.1287/opre.1080.0678}, 2005.

[25] K. Huang, S. Ahmed, The Value of Multistage Stochastic Programming in Capacity Planning Under Uncertainty, Operations Research 57 (4) (2009) 893–904, doi:\bibinfo{doi}{10.1287/opre.1080.0623}.

[26] Á. Marín, P. Jaramillo, Urban rapid transit network capacity expansion, European Journal of Operational Research 191 (2008) 45–60, doi:\bibinfo{doi}{10.1016/j.ejor.2007.08.010}.

[27] A. Karoonsoontawong, S. T. Waller, Integrated Network Capacity Expansion and Traffic Signal Optimization Problem: Robust Bi-level Dynamic Formulation, Networks and Spatial Economics 10 (4) (2010) 525–550, doi:\bibinfo{doi}{10.1007/s11067-008-9071-x}.

[28] L. Gao, C. Xie, Z. Zhang, S. T. Waller, Integrated maintenance and expansion planning for transportation network infrastructure, Transportation Research Record: Journal of the Transportation Research Board (2225) (2011) 56–64, ISSN 03611981, doi:\bibinfo{doi}{10.3141/2225-07}.

[29] Y. Saif, A. Almansoori, A capacity expansion planning model for integrated water desalination and power supply chain problem, Energy Conversion and Management 122 (2016) 462–476, doi:\bibinfo{doi}{10.1016/j.enconman.2016.06.011}.

[30] M. V. Loureiro, J. Claro, P. J. Pereira, Capacity expansion in transmission networks using portfolios of real options, International Journal of Electrical Power and Energy Systems 64 (2015) 439–446, doi:\bibinfo{doi}{10.1016/j.ijepes.2014.07.047}.

[31] M. A. Cardin, S. Zhang, W. J. Nuttall, Strategic real option and flexibility analysis for nuclear power plants considering uncertainty in electricity demand and public acceptance, Energy Economics 64 (2017) 226–237, doi:\bibinfo{doi}{10.1016/j.eneco.2017.03.023}.

[32] M. Taghavi, K. Huang, A Multi-Stage Stochastic Programming Approach for Network Capacity Expansion with Multiple Sources of Capacity, Naval Research Logistics 63 (2017) 600–614, doi:\bibinfo{doi}{10.1002/nav.21726}.

[33] O. Berman, Z. Ganz, J. M. Wagner, A stochastic optimization model for planning capacity expansion in a service industry under uncertain demand, Naval Research Logistics (NRL) 41 (4) (1994) 545–564, doi:\bibinfo{doi}{10.1002/1520-6750(199406)41:4<545::AID-NAV3220410407>3.0.CO;2-Z}.

[34] S. M. Ryan, Capacity Expansion for Random Exponential Demand Growth with Lead Times, Management Science 50 (6) (2004) 740–748, doi:\bibinfo{doi}{10.1287/mnsc.1030.0187}.

[35] A. Garcia, Z. Shen, Equilibrium capacity expansion under stochastic demand growth 58 (1) (2010) 30–42, doi:\bibinfo{doi}{10.1287/opre.1090.0719}.

[36] L. Trigeorgis, Real Options and Interactions with Financial Flexibility, Financial Management 22 (3) (1993) 202–224, doi:\bibinfo{doi}{10.2307/3665939}.

[37] R. DeNeufville, Real Options: Dealing With Uncertainty in Systems Planning and Design, Integrated Assessment 4 (1) (2003) 26–34, doi:\bibinfo{doi}{10.1076/iaij.4.1.26.16461}.

[38] R. E. Swaney, I. E. Grossmann, An Index for Operational Flexibility in Chemical Process Design. Part 2: Computational Algorithms, AIChE Journal 31 (4) (1985) 631–641, doi:\bibinfo{doi}{10.1002/aic.690310413}.

[39] I. E. Grossmann, B. A. Calfa, P. Garcia-Herreros, Evolution of concepts and models for quantifying resiliency and flexibility of chemical processes, Computers & Chemical Engineering 70 (2014) 22 − 34, ISSN 0098-1354, doi:\bibinfo{doi}{https://doi.org/10.1016/j.compchemeng.2013.12.013}, manfred Morari Special Issue.

[40] Q. Zhang, I. E. Grossmann, R. M. Lima, On the relation between flexibility analysis and robust optimization for linear systems, AIChE Journal 62 (9) (2016) 3109–3123, doi:\bibinfo{doi}{https://doi.org/10.1002/aic.15221}.

[41] M. Fitzgerald, Managing Uncertainty in Systems with a Valuation Approach for Strategic Changeability, 2012.

[42] O. Špačková, B. Dittes, D. Straub, Risk based Optimization of Adaptable Protection Measures Against Natural Hazards, in: 12th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP12, 2015.

[43] W. B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality, Wiley, 2nd edn., ISBN 978-0-470-60445-8, 2011.

[44] S. Zhao, W. B. Haskell, M. A. Cardin, Decision rule-based method for flexible multi-facility capacity expansion problem, IISE Transactions 50 (7) (2018) 553–569, doi:\bibinfo{doi}{10.1080/24725854.2018.1426135}.

[45] M. J. Kochenderfer, C. Amato, G. Chowdhary, J. P. How, H. J. D. Reynolds, J. R. Thornton, P. A. Torres-Carrasquillo, N. K. Üre, J. Vian, Decision Making Under Uncertainty: Theory and Application, The MIT Press, 1st edn., ISBN 0262029251, 2015.

[46] P. Brandimarte, Numerical Methods in Finance in Economics, Wiley, 2 edn., ISBN 9780471745037, 2006.

[47] A. Shapiro, D. Dentcheva, A. Ruszczyński, Lectures on Stochastic Programming, Society for Industrial and Applied Mathematics, doi:\bibinfo{doi}{10.1137/1.9780898718751}, 2009.

[48] J. R. Birge, F. Louveaux, Introduction to Stochastic Programming, Springer Science+Business Media, 2 edn., ISBN 978-1-4614-0237-4, doi:\bibinfo{doi}{10.1007/978-1-4614-0237-4}, 2011.

[49] N. Gröwe-Kuska, H. Heitsch, W. Römisch, in: 2003 IEEE Bologna PowerTech - Conference Proceedings, doi:\bibinfo{doi}{10.1109/PTC.2003.1304379}, 2003.

[50] K. Høyland, S. Wallace, Generating Scenario Trees for Multistage Decision Problems, Management Science 47 (2) (2001) 295–307, doi:\bibinfo{doi}{10.1287/mnsc.48.11.1512.261}.

[51] D. Xu, Z. Chen, L. Yang, Scenario tree generation approaches using K-means and LP moment matching methods, Journal of Computational and Applied Mathematics 236 (17) (2012) 4561 – 4579, ISSN 0377-0427, doi:\bibinfo{doi}{https://doi.org/10.1016/j.cam.2012.05.020}.

[52] H. Heitsch, W. Römisch, Scenario tree modeling for multistage stochastic programs, Mathematical Programming 118 (2) (2009) 371–406, doi:\bibinfo{doi}{10.1007/s10107-007-0197-2}.

[53] G. C. Pflug, A. Pichler, Dynamic generation of scenario trees, Computational Optimization and Applications 62 (3) (2015) 641–668, ISSN 1573-2894, doi:\bibinfo{doi}{10.1007/s10589-015-9758-0}.

[54] C. H. Rosa, A. Ruszczyński, On augmented Lagrangian decomposition methods for multistage stochastic programs, Annals of Operations Research 64 (1) (1996) 289–309, ISSN 1572-9338, doi:\bibinfo{doi}{10.1007/BF02187650}.

[55] A. Ruszczyński, Decomposition Methods, in: Stochastic Programming, vol. 10 of *Handbooks in Operations Research and Management Science*, Elsevier, 141 – 211, doi:\bibinfo{doi}{https://doi.org/10.1016/S0927-0507(03)10003-5}, 2003.

[56] H. Bakker, F. Dunke, S. Nickel, A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice, Omega 96 (2020) 102080, ISSN 0305-0483, doi:\bibinfo{doi}{https://doi.org/10.1016/j.omega.2019.06.006}.

[57] J. R. Birge, Decomposition and Partitioning Methods for Multistage Stochastic Linear Programs, Operations Research 33 (5) (1985) 989–1007, ISSN 0030364X, 15265463.

[58] A. Ruszczyński, A regularized decomposition method for minimizing a sum of polyhedral functions, Mathematical Programming 35 (3) (1986) 309–333, ISSN 1436-4646, doi:\bibinfo{doi}{10.1007/BF01580883}.

[59] A. Ruszczyński, Regularized Decomposition of Stochastic Programs: Algorithmic Techniques and Numerical Results. .

[60] R. T. Rockafellar, R. J.-B. Wets, Scenarios and Policy Aggregation in Optimization Under Uncertainty, Mathematics of Operations Research 16 (1) (1991) 119–147, doi:\bibinfo{doi}{10.1287/moor.16.1.119}.

[61] J. M. Mulvey, A. Ruszczyński, A diagonal quadratic approximation method for linear multistage stochastic programming problems, in: L. D. Davisson, A. G. J. MacFarlane, H. Kwakernaak, J. L. Massey, Y. Z. Tsypkin, A. J. Viterbi, P. Kall (Eds.), System Modelling and

Optimization, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-47220-9, 588–597, 1992.

[62] G. Lulli, S. Sen, A Branch-and-Price Algorithm for Multistage Stochastic Integer Programming with Application to Stochastic Batch-Sizing Problems, Management Science 50 (6) (2004) 786–796, doi:\bibinfo{doi}{10.1287/mnsc.1030.0164}.

[63] J. Dupačová, G. Consigli, S. W. Wallace, Scenarios for Multistage Stochastic Programs, Annals of Operations Research 100 (1) (2000) 25–53, ISSN 1572-9338, doi:\bibinfo{doi}{10.1023/A:1019206915174}.

[64] N. Löhndorf, An empirical analysis of scenario generation methods for stochastic optimization, European Journal of Operational Research 255 (2016) 121–132, doi:\bibinfo{doi}{10.1016/j.ejor.2016.05.021}.

[65] G. C. Pflug, A. Pichler, Multistage Stochastic Optimization, Springer International Publishing Switzerland, ISBN 9783319088426, doi:\bibinfo{doi}{10.1007/978-3-319-08843-3}, 2014.

[66] A. Shapiro, Monte Carlo Sampling Methods, in: Stochastic Programming, vol. 10 of *Handbooks in Operations Research and Management Science*, Elsevier, 353 – 425, doi:\bibinfo{doi}{https://doi.org/10.1016/S0927-0507(03)10006-0}, 2003.

[67] G. Guastaroba, R. Mansini, M. G. Speranza, On the effectiveness of scenario generation techniques in single-period portfolio optimization, European Journal of Operational Research 192 (2009) 500–511, doi:\bibinfo{doi}{10.1016/j.ejor.2007.09.042}.

[68] M. Kaut, S. W. Wallace, Evaluation of scenario-generation methods for stochastic programming, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Mathematik, doi:\bibinfo{doi}{10.18452/8296}, 2003.

[69] P. Beraldi, F. De Simone, A. Violi, Generating scenario trees: A parallel integrated simulation-optimization approach, Journal of Computational and Applied Mathematics 233 (2010) 2322–2331, doi:\bibinfo{doi}{10.1016/j.cam.2009.10.017}.

[70] E. Schubert, P. J. Rousseeuw, Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms, Lecture Notes in Computer Science (2019) 171–187ISSN 1611-3349, doi:\bibinfo{doi}{10.1007/978-3-030-32047-8_16}.

[71] J. Löfberg, YALMIP : A Toolbox for Modeling and Optimization in MATLAB, in: In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.

[72] M. C. Cario, B. L. Nelson, Modeling and Generating Random Vectors with Arbitrary Marginal Distributions and Correlation Matrix, Tech. Rep., 1997.