PUC

ISSN 0103-9741

Monografias em Ciência da Computação

n° 05/10

On a Class of Stochastic Programs with Endogenous Uncertainty: Theory, Algorithm and Application

Bruno da Costa Flach

Marcus Vinicius Soledade Poggi de Aragão

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO – BRASIL

Monografias em Ciência da Computação, No. 05/10 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena April, 2010

On a Class of Stochastic Programs with Endogenous Uncertainty: Theory, Algorithm and Application

Bruno da Costa Flach§

Marcus Vinicius Soledade Poggi de Aragão

Abstract. In this work we study a class of stochastic programming problems with endogenous uncertainty – i.e., those in which the probability distribution of the random parameters is decision-dependent – which is formulated as a MINLP. Although discussed in the context of the humanitarian logistics problem, the proposed methodology and obtained results are also valid for a more general class of problems which comprehends a variety of applications. In particular, we propose (i) a convexification technique for polynomials of binary variables, (ii) an efficient cut-generation algorithm and (iii) the incorporation of importance sampling concepts into the stochastic programming framework so as to allow the solution of large instances of the problem.

Keywords. Stochastic programming; Endogenous uncertainty; Convexification; Importance sampling; Humanitarian logistics.

Resumo. Neste trabalho estudamos uma classe de problemas de otimização estocástica com incertezas endógenas – i.e., aqueles em que a distribuição de probabilidade dos parâmetros aleatórios depende das decisões tomadas – que é formulado como um MINLP. Apesar de discutido dentro do contexto do problema de logística humanitária, a metodologia proposta e os resutados obtidos são válidos para uma classe geral de problemas que agrega uma variedade de aplicações. Em particular, propõe-se (i) uma técnica de convexificação de polinômios de variáveis binárias, (ii) um algoritmo de geração de cortes e (iii) a incorporação dos conceitos de importance sampling dentro do contexto de otimização estocástica de modo a permitir a solução de grandes instâncias do problema.

Palavras-chave. Otimização estocástica; Incertezas endógenas; Convexificação; Logística Humanitária

§ Corresponding author: brunoflach@gmail.com

In charge of publications:
Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Decision under uncertainty

In a vast range of practical applications, the input data necessary for the solution of mathematical programs cannot be precisely determined beforehand. In general, that may happen either because data is inherently random or due to inevitable errors in measurement. In 1955, Dantzig [20] and Beale [6] first recognized that even a relatively small deviation from the values used as input data could compromise the quality of the optimal solution to a problem. Since then, two main methodologies have been developed with the aim of incorporating – into the modeling and solution procedures – the uncertainties which are part of a diverse set of problems: robust optimization and stochastic programming.

1.2 Robust Optimization

The field of robust optimization was founded in 1973 by Soyster's seminal work [54] which proposed the solution to a problem similar to that in standard form ($\min_{x \in X} c^T x \mid Ax \leq b$) with the additional requirement that the optimal solution should be feasible for all elements of the set $\mathcal{A} = \{A_j, \forall j \in J\}$ of technology matrices.

Following the notation of Bertsimas and Sim (2004) [15], let J_i denote the set of coefficients in row i of matrix A which are subject to uncertainty and each element $a_{ij}, (j \in J_i)$ be modeled as a symmetric and bounded random variable with support $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$. The formulation proposed by Soyster may be written as:

$$Min \quad c^T x \tag{1.1}$$

$$subject\ to: \quad \sum_j a_{ij} x_j + \sum_{j \in J_i} \hat{a}_{ij} y_j \leq b_i \qquad \forall i \tag{1.2}$$

$$-y_j \leq x_j \leq y_j \qquad \forall j \tag{1.3}$$

$$l \leq x \leq u \tag{1.4}$$

$$y \geq 0 \tag{1.5}$$

where l and u are vectors of appropriate dimension which represent, respectively, lower and upper bounds on variables x_j.

Such an approach is shown by Soyster to be equivalent to a worst-case scenario analysis. This extreme conservativeness leads the value of the objective function at the optimal solution to be usually significantly worse than that of the original (or nominal-value) problem and motivated the search for different approaches which could provide a balance between feasibility and optimality.

A quarter of a century after Soyster's work, Ben-Tal and Nemirovksi ([9], [10], [11] and [12]) and El-Ghaoui et al. ([22] and [23]) proposed an alternative way to model the uncertainty by defining "ellipsoidal regions of uncertainty" around the nominal values of the coefficients, inside which one admits that the realization of the unknown parameters will be. The proposed approach results in a modification of the original constraints of the problem which turns it into a second order conic program, thus requiring specific solution procedures (which are, in general, not guaranteed to find the global optimum solution to a problem):

$$Min \quad c^T x \tag{1.6}$$

$$subject\ to: \quad \sum_j a_{ij}x_j + \sum_{j \in J_i} \hat{a}_{ij}y_{ij} + \Omega_i \sqrt{\sum_{j \in J_i} \hat{a}_{ij}^2 z_{ij}^2} \leq b_i \qquad \forall i \tag{1.7}$$

(1.8)

$$-y_{ij} \leq x_j - z_{ij} \leq y_{ij} \qquad \forall j \tag{1.9}$$

$$l \leq x \leq u \tag{1.10}$$

$$y \geq 0 \tag{1.11}$$

where Ω_i is a user-defined parameter related to the probability of violation of each constraint – the authors prove that the probability of each constraint i being violated is less or equal to $\exp(-\Omega_i^2/2)$.

Robust optimization was again boosted in 2003 with the publication of [13], [14] and [15] by Bertsimas and Sim. The novel approach assumes a polyhedral uncertainty set and its major advantage is the fact that the formulation of the robust counterpart of a problem does not modify its structure, maintaining all the original properties such as linearity. In summary, the proposed approach introduces a parameter Γ_i that takes values in the interval $[0, |J_i|]$ and determines the maximum number of coefficients in row i which will be allowed to vary from their respective nominal values a_{ij}. The robust counterpart is initially formulated as:

$$\begin{array}{rlll}
Min & c^T x & & (1.12) \\
subject\ to: & \sum_j a_{ij}x_j + \beta_i(x,\Gamma_i) \leq b_i & \forall i & (1.13) \\
 & -y_j \leq x_j \leq y_j & \forall j & (1.14) \\
 & l \leq x \leq u & & (1.15) \\
 & y \geq 0 & & (1.16)
\end{array}$$

where:

$$\begin{array}{rlll}
\beta(x,\Gamma_i) = Max & \sum_{j \in J_i} \hat{a}_{ij}|x_j|z_{ij} & & (1.17) \\
subject\ to: & \sum_{j \in J_i} z_{ij} \leq \Gamma_i & \forall i & (1.18) \\
 & 0 \leq z_{ij} \leq 1 & \forall j \in J_i & (1.19)
\end{array}$$

As shown in [15], this is equivalent to the linear formulation presented below:

$$\begin{array}{rlll}
Min & c^T x & & (1.20) \\
subject\ to: & \sum_j a_{ij}x_j + z_i\Gamma_i + \sum_{j \in J_i} p_{ij} \leq b_i & \forall i & (1.21) \\
 & z_i + p_{ij} \geq \hat{a}_{ij}y_j & \forall i, j \in J_i & (1.22) \\
 & -y_j \leq x_j \leq y_j & \forall j & (1.23)
\end{array}$$

$$l \leq x \leq u \tag{1.24}$$

$$y, z, p \geq 0 \tag{1.25}$$

1.3 Stochastic programming

The stochastic programming approach relies on the assumption – which is perfectly reasonable in various settings – that one might be able to know or estimate the probability distribution of the unknown parameters. Generally speaking, the objective of stochastic programming models is to determine a solution that is feasible for all possible data realizations (or for a given percentage of them) and that minimizes the expected value of a function of the decision and random variables.

The objective of this Section is not to provide a comprehensive overview on the subject – which the interested reader may find in Birge and Loveaux (1997) [16], Kall and Wallace (1994) [40], Ruszczynski and Shapiro (2003) [52], Shapiro, Dentcheva and Ruszczynski (2009) [55] and Haneveld and van der Vlerk (2005) [33] – but to introduce the topic so that the reader may grasp the basic difference between standard stochastic programming models in the literature and the one studied in this work. In addition to the basic references just mentioned, the state-of-the-art in various applications may be found in Wallace and Fleten (2003) [67] (energy), Dupacova, Hurt and Stepan (2002) [21] (finance), Poojari, Lucas and Mitra (2006) [49] (supply chain and logistics) and Gaivoronski (2005) [26] (telecommunications).

The majority of research and applications of stochastic programming is done on the so-called two-stage stochastic programming linear models, although multistage stochastic programs are also the subject of great interest – a graphical depiction of the conceptual difference between two-stage and multistage models is presented in Figure 1-1. In the former case, one usually seeks to determine a first stage decision which is then succeeded by the realization of a random event that affects the outcome of the action taken. Recourse actions may then be taken in the second stage so as to compensate for potential damages caused by the realization of the random variable(s). While in the second stage there might be a different set of corrective decisions for each scenario, according the possible outcomes of the random event, first stage decisions for all scenarios are required to be the same – a condition usually referred to as non-anticipativity.

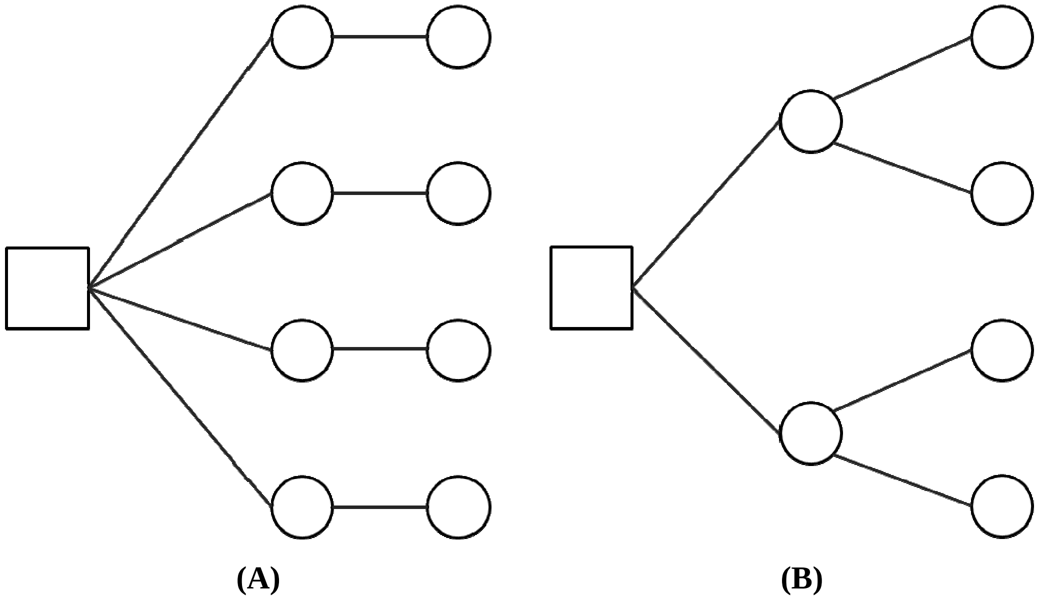


Figure 1-1 – Two-stage (A) and multistage (B) scenario-tree structure of stochastic programming models

The general formulation of a two-stage stochastic program is presented next:

$$Min \quad c^T x + \mathbb{E}\{Q(x,\xi)\} \tag{1.26}$$

$$subject\ to: \quad Ax \leq b \tag{1.27}$$

$$x \in X \tag{1.28}$$

where $Q(x,\xi)$ is defined as the value of the optimal solution of the second stage problem:

$$Min \quad q(\xi)^T y \tag{1.29}$$

$$subject\ to: \quad T(\xi)x + W(\xi)y \leq h(\xi) \tag{1.30}$$

$$y \in Y \tag{1.31}$$

The actions to be taken before the random parameters are known are determined by the vector of first stage decision variables x, whose feasible region is defined by the set of constraints $Ax \leq b$ and by the set X – which may include integrality constraints. The vector of second stage decision variables is denoted by y and the the vector of coefficients of the objective function q, technology matrices T and W and the right-hand side vector h may all depend on the vector of random variables ξ.

Difficulties in evaluating multi-dimensional integrals imply that the determination of a numerical solution to these problems usually require the enumeration of a finite number S of possible outcomes for the vector $\xi = \{\xi_1, \xi_2, \dots, \xi_S\}$. Each one of these outcomes is called a scenario, to which there must also be an associated probability of occurrence $p = \{p_1, p_2, \dots, p_S\}$. This discretization allows the expression for the expected value in equation (1.22) to be written as:

$$\mathbb{E}\{Q(x,\xi)\} = \sum_{s \in S} p_s \cdot Q(x, \xi_s) \tag{1.32}$$

Finally, problems (1.26) – (1.28) and (1.29) – (1.31) may now be jointly re-written as follows:

$$\begin{aligned}
Min \quad & c^T x + \textstyle\sum_{s \in S} p_s q_s y_s && && (1.33)\\
subject\ to: \quad & Ax \le b && && (1.34)\\
& T_s x + W_s y_s \le h_s && \forall s \in S && (1.35)\\
& x \in X, y \in Y && && (1.36)
\end{aligned}$$

1.4 Motivation and related bibliography

A common hypothesis concerning the two approaches discussed above is that the realization of the uncertain parameters is independent of the decision variables, a illustrated in . This conjecture is valid in a variety of applications, such as portfolio optimization, hydrothermal scheduling for electricity generation, communication network planning under demand uncertainty, etc. Not surprisingly, the vast majority of the body of work both in robust optimization and in stochastic programming deals with problems in which this hypothesis is satisfied and the uncertainty is said to be exogenous.

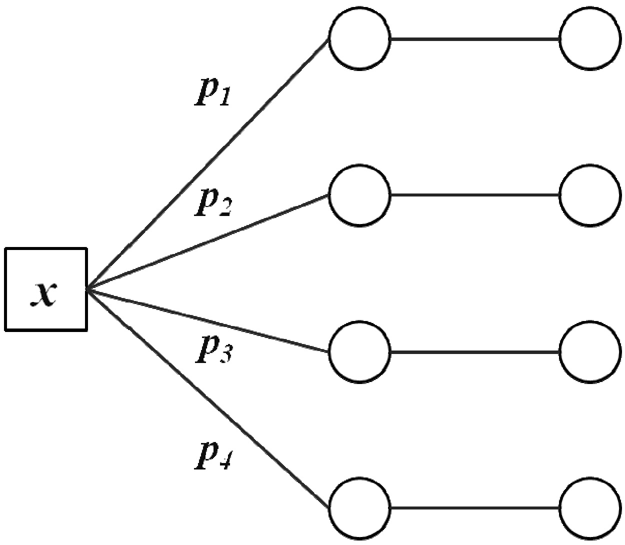


Figure 1-2 – Stochastic programming model with exogenous uncertainty – probabilities p_1, p_2, p_3 and p_4 are independent of decision x

On the other hand, the literature on problems where the knowledge of the probability of occurrence of random events depends on the decisions taken (i.e., when the uncertainty is said to be endogenous) is very limited. According to Goel and Grossmann (2006) [31], out of the 4300+ works in the Stochastic Programming Bibliography compiled by van der Vlerk [65], only 8 ([48], [66], [2], [39], [36], [30], [31] and [61]) involve the case of endogenous uncertainty (references [54] and [47] are other works on the subject, not yet included in the database).

The work on stochastic programs with endogenous uncertainty may be further sub-divided into two categories with respect to the particular way in which decisions affect the knowledge of the probability distributions.

The first group involves problems where the probability distribution of the random variables is not directly affected but, rather, uncertainty may be partially resolved depending on actions performed by the decision-maker. This is essentially related to the timing of information discovery and to an anticipation or delay of the moment at which more accurate information is revealed. Such situation is pictured in Figure 1-3 below, in which the dashed line represents a possible relaxation of non-anticipativity constraints between scenarios related to first-stage decisions.

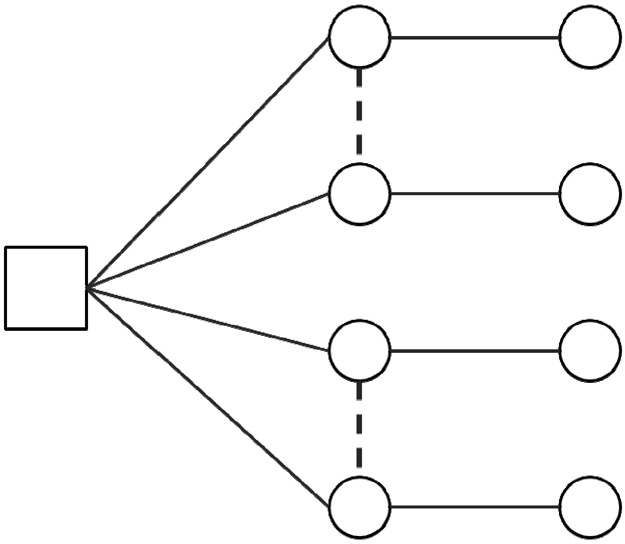


Figure 1-3 – Endogenous uncertainty related to the time of information discovery

This group includes the work of Jonsbraten (1998) [39], Goel and Grossmann (2004, 2006) [30][31], Held (2003) [36] and Senay (2007) [54]. The type of uncertainty dealt with in these works is exemplified by that studied in [39] and [30] where an oil and gas exploration company must choose among different testing and probing methods in order to try and find the size and quality of reserves – the installation of a facility does not change the likelihood of the company actually finding oil, but may provide evidence as to what are the most probable scenarios. Other examples lie in the areas of project management [54] and network interdiction.

Finally, the second group of stochastic programs with endogenous uncertainty refers to those in which decisions directly affect the probability distribution of the random parameters i.e., the actions performed at a given stage may change the probability of occurrence of future events – as conceptually illustrated in Figure 1-4.

Pflug (1990) [48] was the first to address this issue by discussing an application in stochastic queuing networks – decisions affect the arrival and service rates of each element in the queue – and proposing a stochastic quasigradient algorithm which requires repeated simulations of the system's functioning for each fixed first-stage solution. Talluri and Ryzin (2004) [61] worked on a revenue management problem from the point of view of an airline who must choose which combination of fares to offer at each moment in time preceding the departure of a flight. Under some assumptions regarding consumer behavior, they developed a dynamic programming algorithm to determine the pricing policy which results in the maximum total expected revenue. In 2000, Ahmed [2] presented some examples related to network design, server selection and facility location. These problems were formulated under a hyperbolic programming framework and a specialized algorithm was developed. An application to the stochastic PERT (Program Evaluation and Review Technique) problem is developed by Plambeck et al. in [47] where one seeks to minimize two conflicting objectives: a project's cost and its completion time. A sample-path algorithm is proposed and results are presented under the assumption of uniform distributions with a fixed spread around the mean. Viswanath et al. (2004) [66] studied the humanitarian logistics problem – briefly described in Section 1.5 below and then again discussed in Chapter 2 in a more detailed fashion – and proposed an approximation to the objective function which allows the simplification of the problem down to an ordinary knapsack problem.

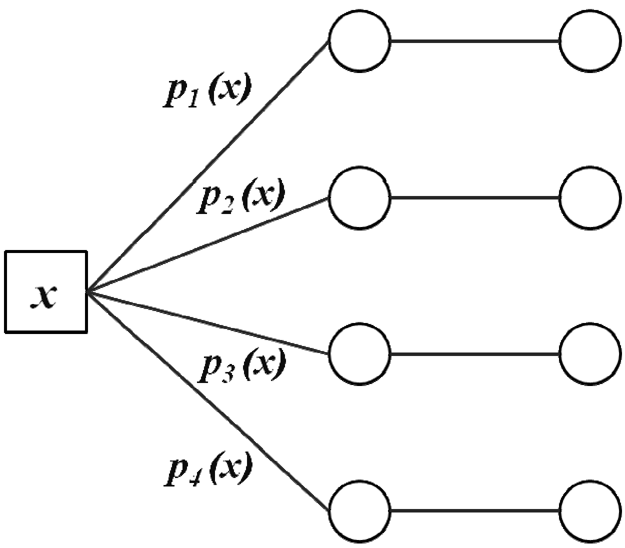


Figure 1-4 – Endogenous uncertainty and decision-dependent probabilities

Given the diminished amount of research on the topic, it is expected that there should be many questions to be answered. In the next section a brief description of the specific problem to be tackled is given, along with a characterization of a more general class of problems for which the results obtained in this work are also valid.

1.5 Objective and contributions

This work will focus on the second group of stochastic programs with endogenous uncertainty discussed above and, in this sense, the humanitarian logistics problem (as defined in Viswanath et al. [66]) will be used as the main motivating example.

A detailed description of the problem is provided in Chapter 2 but, essentially, it refers to the problem of determining the optimal set of investments on the reinforcement of the links of a network which are subject to random failures – the decision to reinforce a link increases the probability that it will be available afterwards.

The results presented here, although discussed in the context of the humanitarian logistics problem, should also hold for a more general class of problems, including some of those discussed above – namely the ones related to stochastic queuing networks, stochastic PERT and revenue management. The general formulation of such problem class is given by:

$$\begin{aligned} Min \quad & c^T x + \mathbb{E}_x\{Q(x, \xi(x))\} && (1.37) \\ subject\ to\text{:} \quad & Ax \leq b && (1.38) \\ & x \in X && (1.39) \end{aligned}$$

where the function $Q(x, \xi(x))$ is now defined as the optimal solution of the following second stage problem:

$$\begin{aligned} Min \quad & q(\xi(x))^T y && (1.40) \\ subject\ to\text{:} \quad & W(\xi(x))y \leq h(\xi(x)) && (1.41) \\ & y \in Y && (1.42) \end{aligned}$$

It is important to observe that the coupling between the first and second stages is not given by the existence of the term Tx as in the set of constraints

(1.35) of problem (1.33) – (1.36) but by the dependence of the probability distribution of the random variables with respect to first stage decision variables x – evidenced by the subscript x in the expression $\mathbb{E}_x\{Q(x,\xi(x))\}$.

The methodology proposed in this work will allow the determination of provably optimal solutions to instances of problems much larger than those currently solved in the literature. Specifically, the contributions are:

1) **Reformulation scheme** which avoids the non-linearities due to products of first and second stage variables and due to the calculation of scenarios probabilities.

2) **Provably finite cut generation algorithm** that overcomes a potential pitfall of the proposed linearization technique and allows the solution of moderately-sized instances for a given error tolerance level;

3) **Incorporation of importance sampling concepts** into the stochastic programming framework. This overcomes the problem of not knowing the probability distribution of the random variables beforehand and allows the solution of large sample-based instances of the problem.

1.6 Outline

The remainder of this work is organized as follows: Chapter 2 describes the humanitarian logistics problem in detail, with a special emphasis on the difficulties that arise out of its formulation; Chapter 3 presents the re-formulation scheme which solves the obstacles related to existing non-linearities; Chapter 4 introduces the approximation algorithm based on cut generation and Chapter 5 extends this algorithm into a statistical framework in order to consider instances of the problem that are not amenable to complete scenario enumeration; Chapter 6 presents computational results, Chapter 7 concludes and discusses future work alternatives and how the developments presented in the previous chapters may be extended to other contexts.

2 HUMANITARIAN LOGISTICS PROBLEM

2.1 Introduction

The impacts of natural or man-made disasters can be very significant in terms of death toll and damages to affected regions. Earthquakes, hurricanes and floods have recently proven their catastrophic potential and concerns over global warming and climate change worsen the perspective in years to come. Besides the immediate loss of lives and destruction of infra-structure, the effects of these calamities usually last long after the initial strike. When an earthquake strikes a city, for example, utility services such as water, electricity and gas may have to be interrupted for weeks before necessary repairs are carried out. On top of that, several roads and bridges are usually affected, rendering the transportation network severely impaired. It has been pointed out [64] that more casualties actually happen due to the isolation to which many residents are forcefully put to rather than by the event itself. This has also been the experience reported by humanitarian organizations in the aftermath of the recent earthquake in Haiti [28].

In face of that, regions that are prone to the occurrence of natural disasters must take preventive measures in order to mitigate potential damages, and devise emergency plans so that they are able to provide care for those affected by such events. It is clear that it is very important to assess the vulnerability of the transportation network and to take steps aimed at guaranteeing that it will be possible to either evacuate people to safe locations or to provide them with basic resources in post-disaster days.

The objective of the humanitarian logistics problem is to determine the optimal set of investments on the seismic retrofit of the links of a transportation network so as to minimize the sum of (deterministic) investment costs and expected (probabilistic) costs incurred when transporting people and/or resources after a catastrophic event. Investment in bridges and tunnels, for example, may increase their resilience so that an earthquake is less likely to render them unusable – Cooper et al. (1994) [19]. Such investments usually involve very large sums of money and a limited budget must thus be optimally allocated.

2.2 Literature review

The literature on the humanitarian logistics problem is very limited. To the best of our knowledge, there are only three papers that deal with the same (or a very similar) problem as the one studied in this work.

Viswanath, Peeta and Salman (2004) [66] were the first to state the problem, motivated by the risks of an earthquake hitting Istambul, the capital of Turkey. They limit the scope of their model to the case where one is interested in maintaining connectivity between origin (O) and destination (D) pairs. Their approach relies on the enumeration of the paths O-D (which, for practical purposes and due to computational difficulties is limited to listing a pre-defined number of paths by using a *k*-shortest path algorithm). Next, they propose an approximation of the objective function based on the first order terms of its Taylor series expansion. As they recognize in their article, the disadvantage of this approach is that by ignoring higher order terms they neglect the potential synergies of simultaneously investing in more than one link.

Liu, Fan and Ordonez (2006) [43] and Fan and Liu (2009) [24] also study the stochastic network protection problem. In the former, the problem follows the same outline as that described above [66] and they propose an extension of the L-Shaped method of Van Slyke and Wets by using generalized Benders decomposition. In the latter, the second-stage problem involves the determination of a Nash equilibrium by solving an MPEC (mathematical program with equilibrium constraints) which results from the consideration that users may choose their own best-perceived routes along the network. Their solution method relies on the application of the Progressive Hedging algorithm of Rockafellar and Wets (1991).

Both articles, however, make the explicit assumption that the decision to invest on the reinforcement of a link eliminates the probability that it might become unavailable after the disaster. They argue that it would be preferable and more realistic to maintain a probabilistic view on link failures but doing so would lead the problem to fall under the class of stochastic programming problems with decision-dependent uncertainties for which “mathematical analysis (…) is very sparse, and is only limited to convex problems of special structures” thus relying "heavily on heuristic methods to solve problems with realistic sizes due to computational difficulties".

Although not dealing with the same problem, there are some related works on the investment in links of a stochastic network. Wollmer (1980) [69] focused on a generalized multicommodity network in which links have random capacities. He formulated the problem as a two-stage stochastic program – where first-stage decisions are the amounts to be invested on the increase of link capacities, and second-stage variables represent the flows of each commodity through the links – and proposed a cutting plane technique that exploits network structure. In 1987, Wallace [67] studied the problem of investing in new links in a network where existing link capacities are random. He also formulated it as a standard two-stage stochastic program and suggested decomposition strategies to solve it. Again in 1991, Wollmer [70] worked on a problem in which one seeks to optimize the tradeoff between first-stage investment costs and second-stage expected maximum flow between a pair of nodes. The formulation follows the regular two-stage stochastic programming framework and was solved using an algorithm based on cutting planes.

Finally, there is also a significant body of work on the development of plans for disaster preparedness and response which adopt a different perspective from that of mathematical programming. Instead, these works usually take a somewhat heuristic view to determine critical links of a network based on a set of pre-defined criteria. Sohn et al. (2003) [59] and Sohn (2006) [58] study the prioritization of links which may become unavailable due to earthquakes in the Midwest states in the US or due to floods in Maryland, US. Based on a disaster scenario, they analyze the potential disruptions and their consequences with respect to travel delays, reconstruction costs and accessibility to affected cities/counties. This is also in line with the approach of Basoz and Kiremidjian (1995) [6] and Bana e Costa, Oliveira and Vieira (2008) [7] who use Palo Alto, CA and Lisbon, respectively, as case studies for their methodologies which consider the physical characteristics of bridges and the social and economical aspects which may be adversely affected by disasters.

2.3 Mathematical formulation

Mathematically, the problem is formulated by assuming we are given an undirected graph $G = (N, E)$ with node set N and edge set E. Nodes represent locations where survivors and/or resources may be located, and arcs represent the roads, bridges and tunnels which comprise the transportation network. For ease of presentation, a deterministic supply or demand h_i is associated with each node i. Edges have non-negative transportation costs c_e, capacity u_e and are assumed to be available after the occurrence of the disastrous event with probabilities q_e^C. As also stated in related works [66], it is assumed that each edge fails independently of the others – although this is not a necessary assumption for the methods proposed in this work. The survival probability of an arc may be increased to q_e^I if an amount r_e is invested in it. We associate the availability of an arc to the value of a random variable ξ_e, which is equal to 1 if the edge e is operational and 0 otherwise.

Assuming that we are able to enumerate all the possible scenarios S of network configuration, the problem may be formulated as follows:

$$(\boldsymbol{P}) \quad Min \quad \sum_{e\in E} r_e x_e + \sum_{s\in S} p_s \left(\sum_{e\in E} c_e y_{es} + \sum_{i\in N} d_i z_{is} \right) \tag{2.1}$$

$$subject\ to: \quad Ax \leq b \tag{2.2}$$

$$W_s y_s + z_s \leq h_s \qquad \forall s \in S \tag{2.3}$$

$$p_s = \prod_{e\in E} (\mathrm{p}_{es}^C + (\mathrm{p}_{es}^I - \mathrm{p}_{es}^C) \cdot x_e) \qquad \forall s \in S \tag{2.4}$$

$$y_{es} \leq u_e \xi_{es} \qquad \forall s \in S, \forall e \in E \tag{2.5}$$

$$x \in \{0,1\}^{|E|};\ y, z \in \mathbb{R}^+ \tag{2.6}$$

where:

ξ_{es}	realization of random variable ξ_e in scenario s
p_{es}^C	probability of the availability status of edge e in scenario s, given that no investment is made on it (i.e., $P(\xi_e = \xi_{es} \mid x_e = 0)$ or, alternatively, $q_e^C \cdot \xi_{es} + (1 - q_e^C) \cdot (1 - \xi_{es})$

p^I_{es}	probability of the availability status of edge e in scenario s, given that a reinforcement investment is made on it (i.e., $P(\xi_e = \xi_{es} \vert x_e = 1)$ or, alternatively, $q^I_e \cdot \xi_{es} + (1 - q^I_e) \cdot (1 - \xi_{es})$
d_i	penalty cost for the non-fulfillment of demand of node i
p_s	continuous variable equal to the probability of scenario s
x_e	binary variable which is equal to 1 if an investment is to be made on edge e, 0 otherwise
y_s	vector of continuous flow variables of scenario s
z_s	vector of continuous slack variables for the demand and supply of each node in scenario s

The objective function (2.1) to be minimized provides the sum of deterministic costs incurred in the first stage due to decisions of reinforcement investments and expected second-stage costs of routing commodities through the network and demand curtailment. Expressions (2.2) and (2.3) represent, respectively, the sets of first-stage constraints (such as budget limitations, minimum investment in each region, etc.) and second-stage constraints (such as mass-balance equations on the realized network configuration of each scenario). Expression (2.4) defines variables p_s as a function of investment decision variables x_e and constraint (2.5) determines the upper bound of the flow in edge e, according to the realization of the random variable ξ_e in scenario s.

Problem (2.1) – (2.6) is a mixed-integer nonlinear program for which solution methods are usually not guaranteed to find a global optimal solution. In particular, there are three main difficulties associated with this formulation that prevent existing algorithms to obtain global optimal solutions. These obstacles are briefly described below; following that, Chapter 3 presents a reformulation scheme that overcomes the first two difficulties and Chapter 5 proposes a solution to the third.

1) **Non-linearity due to product of first and second stage variables.** In standard stochastic programming problems the probability of a scenario is known and it thus usually becomes a coefficient of the objective function. In the case of the class of problems being studied in this work, the

expression for the expected value of second stage costs – $\sum_{s \in S} p_s (\sum_{e \in E} c_e y_{es} + \sum_{i \in N} d_i z_{is})$ – involves the product of first stage variables p_s – since, as described earlier, first stage decisions affect the probability of occurrence of each possible outcome – and second stage variables y_{es} and z_{is}.

2) **Non-linearity due to the expression for the scenarios' probabilities.** A second source of non-linearity arises from the expression that defines variables p_s themselves, which represent the probabity of occurrence of each possible network configuration after taking into account first stage investment decisions. In this case, the expression involves non-linear terms of order up to $|E|$ due to products of binary variables x_e : $p_s = \prod_{e \in E} (\mathrm{p}_{es}^{C} + (\mathrm{p}_{es}^{I} - \mathrm{p}_{es}^{C}) \cdot x_e)$. These non-linear terms arise from the product of the probability of occurrence of the outcome of each random variable that composes a scenario.

3) **Scenario generation.** As previously mentioned, most stochastic programming models deal with random variables whose probability distribution is independent of the decision variables. This *a priori* knowledge of the joint probability distribution allows one to obtain scenarios for the realization of the random variables and their respective probabilities of occurrence – either by sampling from it in a Monte Carlo fashion or by constructing them based on a given criteria (e.g., moment matching such as in Kaut and Wallace (2007) [41] and Kaut, Wallace and Hoyland (2003) [42] or minimization of distances between probability measures – Romisch (2009) [50], Heitsch and Romisch (2005) [35] and Hochreiter and Pflug (2007) [37]) – which may then be used to numerically compute the expectation of second stage costs, as described in Chapter 1. Since the probability distribution of the random variables is not known beforehand in the class of problems being studied in this work (i.e., it can only be computed after first stage decisions are determined), one cannot rely on existing scenario generation methods.

3 REFORMULATION SCHEME

In this Chapter, a reformulation scheme which overcomes the difficulties associated with the existence of non-linear terms in the problem formulation will be presented. Section 3.1 describes the argument which allows the elimination of the product between first and second stage variables while Section 3.2 proposes a linearization technique that eliminates the products among binary variables; Section 3.3 is dedicated to a discussion of the approximation error and of how it can be managed when solving a problem.

3.1 Separability of second stage problems

The product between variables p_s and y_{es} in the objective function may be removed by observing that the feasible regions of the second-stage problems – sets of constraints (2.3) and (2.5) – are decoupled from first-stage variables. The second-stage problem of each scenario may then be solved independently of the others:

$$\forall s \in S, g_s = \quad Min \quad \sum_{e \in E} c_e y_{es} + \sum_{i \in N} d_i z_{is} \tag{3.1}$$

$$subject\ to: \quad W_s y_s + z_s \leq h_s \tag{3.2}$$

$$y_{es} \leq u_e \xi_{es} \qquad \forall e \in E \tag{3.3}$$

$$y, z \in \mathbb{R}^+ \tag{3.4}$$

As shown above, we denote by g_s the value of the optimal solution of problem (3.1) – (3.4) for a given scenario s, which then allows us to re-write problem (2.1) – (2.6) as follows:

$$(\boldsymbol{P_1}) \qquad Min \quad \sum_{e \in E} r_e x_e + \sum_{s \in S} p_s g_s \tag{3.5}$$

$$subject\ to: \quad Ax \leq b \tag{3.6}$$

$$p_s = \prod_{e \in E} (\mathrm{p}_{es}^{C} + (\mathrm{p}_{es}^{I} - \mathrm{p}_{es}^{C}) \cdot x_e) \qquad \forall s \in S \qquad (3.7)$$

$$x \in \{0,1\}^{|E|} \qquad (3.8)$$

In the following we will assume that $g_s \geq 0, \forall s \in S$. However, this hypothesis comes without loss of generality, as shown in Annex A.

3.2 Polynomials in binary variables

A remaining difficulty in solving problem (3.5) – (3.8) lies on the product of binary variables x_e in the definition of variables p_s – each equation defined in the set of constraints (3.7) is a polynomial of order $|E|$.

There has been a significant amount of research on the linearization of the product of binary variables. Following the initial article of Glover in 1975 [29], there have been related works focused on quadratic functions – Hansen and Meyer (2009) [34], Balas and Mazzola (1984) in [4] and [5], Gueye and Michelon (2005) [32] – but some authors have also considered the case of cubic and higher-degree polynomials – c.f., Adams and Forrester (2005) [1], Chang (2000) [17], Chang and Chang (2000) [18], Oral and Ketani (1990 and 1992) [44] and [45]. Essentially, the proposed techniques resort to the addition of auxiliary variables and constraints to linearize each non-linear term in the problem. Since the definition of each variable p_s implies an exponential number of nonlinear terms ($\sum_{k=2}^{|E|} \binom{|E|}{k}$ or, equivalently, $2^{|E|} - |E| - 1$), these methods result impractical for the class of problems under consideration. The special structure of the polynomials defined in the set of constraints (3.7) – specifically, the fact that they may be written as the product of linear terms in the form $a \cdot x + b$, where $a > 0$ and $a + b > 0$ – allows for the straightforward application of the linearization technique proposed in this work, described below.

3.2.1 Proposed linearization technique

By relying on the fact that $a = b \cdot c \rightarrow a = \exp(\ln b + \ln c)$, each equation in (3.7) may be re-written as:

$$\mathrm{p}_s = \exp\left(\sum_{e\in E} \ln(\mathrm{p}^C_{es} + (\mathrm{p}^I_{es} - \mathrm{p}^C_{es}) \cdot x_e)\right) \tag{3.9}$$

Since x is a vector of binary variables, the expression within the summation operator may also be re-written in such a way that variables x_e are not part of the logarithmic expression. This is accomplished by observing that the argument of each logarithm is p^C_{es} if x_e is equal to 0 and p^I_{es} otherwise, leading to:

$$\mathrm{p}_s = \exp\left(\sum_{e\in E} \{\ln(\mathrm{p}^C_{es}) + [\ln(\mathrm{p}^I_{es}) - \ln(\mathrm{p}^C_{es})] \cdot x_e\}\right) \tag{3.10}$$

A continuous variable may be defined as the logarithm of the probability of each scenario, thus being an affine function of variables x_e (this auxiliary variable is introduced for ease of presentation but it is not strictly necessary):

$$w_s = \ln(\mathrm{p}_s) = \sum_{e\in E} \{\ln(\mathrm{p}^C_{es}) + [\ln(\mathrm{p}^I_{es}) - \ln(\mathrm{p}^C_{es})] \cdot x_e\} \tag{3.11}$$

Having the value of the natural logarithm of the probability of a scenario given by expression (3.11), the actual value of its probability (i.e., the value of p_s) may be obtained by a piecewise linear approximation of the exponential function. Since the optimization sense of the problem is to minimize and the exponential function is convex, this approximation may be represented by a set of linear constraints which can be incorporated into the problem.

Example. Let there be a network connecting cities A, B and C composed of two links (AB and BC), as shown in the Figure below. Suppose the current (i.e., pre-investment) survival probability of each link is given by $P_{AB} = 50\%$ and $P_{BC} = 60\%$. If a reinforcement investment is made on each link, these probabilities increase to $P_{AB} = 70\%$ and $P_{BC} = 90\%$, respectively. There are obviously four possible scenarios of network configuration and we will use the

ISSN: 0103–9741

one where both links are operational to illustrate the proposed linearization technique.

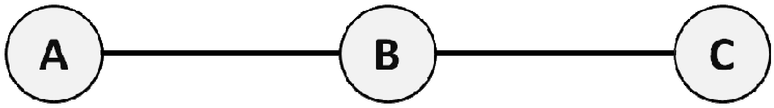


Figure 3-1 – Two-link network example

The probability of the scenario in which both links survive (s_1) is given by the following expression:

$$P_{s_1} = (50\% + 20\% \cdot x_{AB}) \times (60\% + 30\% \cdot x_{BC}) \tag{3.12}$$

The application of the logarithm to both sides of equation (3.12) results in:

$$\ln(P_{s_1}) = \ln(50\% + 20\% \cdot x_{AB}) + \ln(60\% + 30\% \cdot x_{BC}) \tag{3.13}$$

The first term of the right-hand side of expression (3.13) is equivalent to $\ln(50\%) + [\ln(70\%) - \ln(50\%)] \cdot x_{AB}$ (or, written in a slightly different form, $\ln(50\%) \cdot (1 - x_{AB}) + \ln(70\%) \cdot x_{AB}$). An analogous transformation may be applied to the second term of the right-hand side of expression (3.13), resulting in:

$$\begin{aligned} \ln(P_{s_1}) = w_{s_1} = \ln(50\%) + [\ln(70\%) - \ln(50\%)] \cdot x_{AB} + \\ \ln(60\%) + [\ln(90\%) - \ln(60\%)] \cdot x_{BC} \end{aligned} \tag{3.14}$$

The values of the logarithm of the probability of occurrence of scenario s_1 (i.e., the possible values of variable w_{s_1} defined above) are given in the Table below for all possible values of the investment decision variables x_{AB} and x_{BC}.

$\boldsymbol{x_{AB}}$	$\boldsymbol{x_{BC}}$	$\boldsymbol{ln(P_{AB})}$	$\boldsymbol{ln(P_{BC})}$	$\boldsymbol{ln(P_{s_1})}$
0	0	$\ln(50\%)$	$\ln(60\%)$	$\ln(30\%) = -1.204$
0	1	$\ln(50\%)$	$\ln(90\%)$	$\ln(45\%) = -0.799$
1	0	$\ln(70\%)$	$\ln(60\%)$	$\ln(42\%) = -0.868$

x_{AB}	x_{BC}	$ln(P_{AB})$	$ln(P_{BC})$	$ln(P_{s_1})$
1	1	$\ln(70\%)$	$\ln(90\%)$	$\ln(63\%) = -0.462$

Table 3-1 – Probability of occurrence of scenario s_1 according to the investment decisions

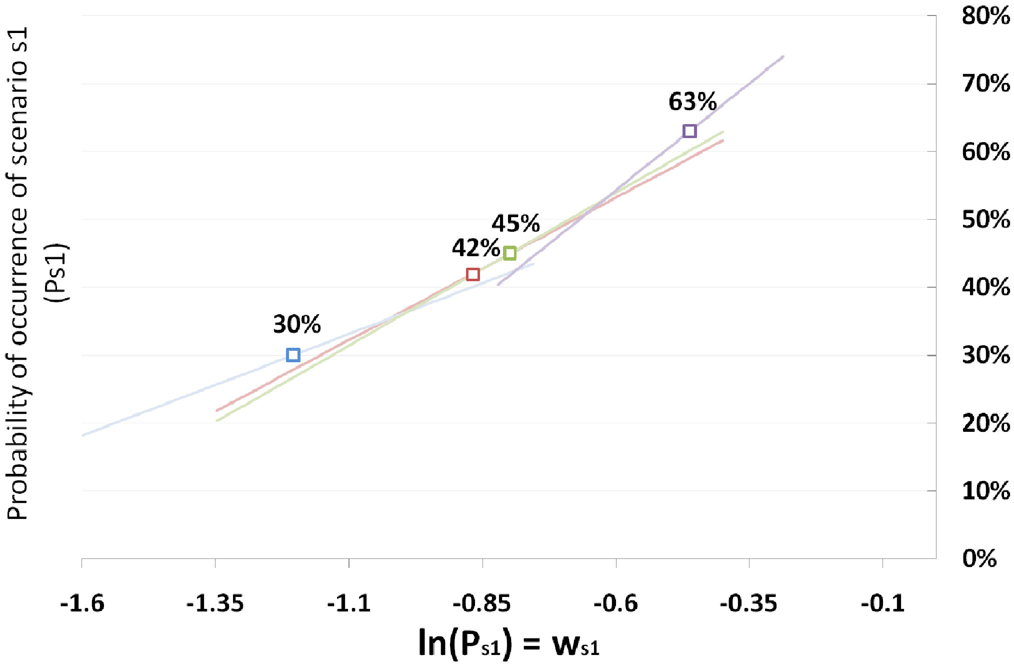
The scenario's actual probability of occurrence, represented by variable $\hat{p}_{s_1}$, may be obtained by adding to the problem the inequalities corresponding to the first order terms of the Taylor series expansion of the exponential function around the possible values of variable w_{s_1} – specified below and represented by the linear segments depicted in the following Figure:

$$\hat{p}_{s_1} \geq 30\% + 30\% \cdot (w_{s_1} - \ln(30\%)) \quad (3.15)$$

$$\hat{p}_{s_1} \geq 42\% + 42\% \cdot (w_{s_1} - \ln(42\%)) \quad (3.16)$$

$$\hat{p}_{s_1} \geq 45\% + 45\% \cdot (w_{s_1} - \ln(45\%)) \quad (3.17)$$

$$\hat{p}_{s_1} \geq 63\% + 63\% \cdot (w_{s_1} - \ln(63\%)) \quad (3.18)$$



ISSN: 0103–9741

Figure 3-2 – Inequalities that provide a piecewise linear approximation to the exponential function

This would ensure that for every possible combination of the values of variables x_{AB} and x_{BC}, the value of $\hat{p}_{s_1}$ would be exactly equal to the probability of occurrence of scenario s_1.

3.2.2 Reformulation

Following the linearization technique proposed in the previous Section, we re-write problem (3.5) – (3.8), eliminating the non-linearities:

$$(\boldsymbol{P_2}) \quad Min \quad \sum_{e \in E} r_e x_e + \sum_{s \in S} g_s \hat{p}_s \tag{3.19}$$

$$subject\ to: \quad Ax \leq b \tag{3.20}$$

$$w_s = \sum_{e \in E} \{\ln(\mathrm{p}_{es}^{C}) + [\ln(\mathrm{p}_{es}^{I}) - \ln(\mathrm{p}_{es}^{C})] \cdot x_e\} \qquad \forall s \in S \tag{3.21}$$

$$\hat{p}_s \geq \alpha_k + \beta_k \cdot w_s \qquad \forall s \in S, \forall k \in K \tag{3.22}$$

$$\hat{p} \in \mathbb{R}^+, w \in \mathbb{R} \tag{3.23}$$

$$x \in \{0,1\}^{|E|} \tag{3.24}$$

where:

K	set of linear constraints that approximate the exponential function
α_k, β_k	coefficients of the k-th segment used to approximate the exponential function
w_s	continuous variable equal to the natural logarithm of the probability of scenario s
$\hat{p}_s$	continuous variable equal to the approximation of the probability of scenario s

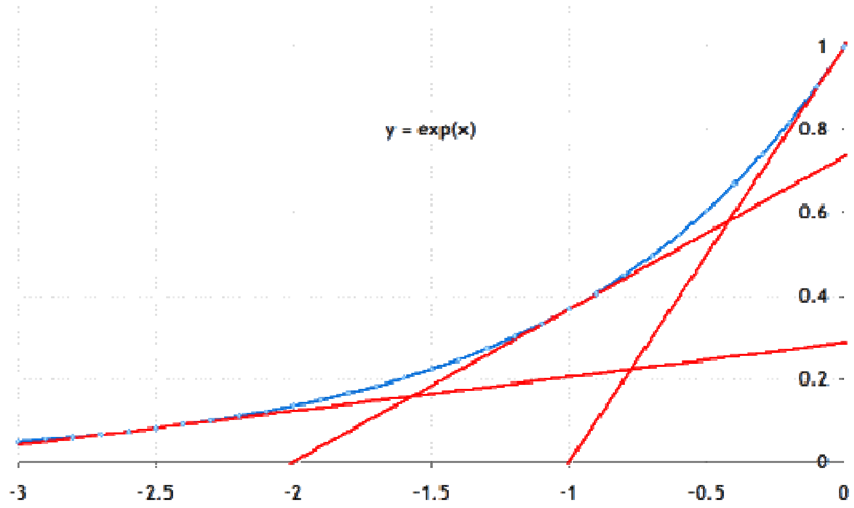


Figure 3-3 – Piecewise linear approximation of the exponential function

Given an approximation to the exponential function (i.e., given a set of cuts in the form $y \geq \exp(w_0) + \exp(w_0) \cdot (w - w_0)$ that provide a piecewise linear approximation to the exponential function) and assuming it is computationally feasible to enumerate and solve the second stage problems for all possible network configurations, one is able to solve problem (3.19) – (3.24) using commercially available solvers. The following sub-sections discuss the necessary number of additional constraints for an exact solution to the problem and the generation of constraints for a given error tolerance level ε.

3.2.3 Additional constraints

According to the set of constraints (3.21), each variable w_s is equal to the sum of the logarithm of the probability of the availability status of each edge e in scenario s. Each of these logarithms may take one out of two possible values depending on whether an investment is made on the corresponding edge and, consequently, variables w_s may potentially assume $2^{|E|}$ different values. In order to guarantee that the optimal solution to the reformulated problem corresponds to the global optimum of the original problem, there must be a cut to approximate the exponential function centered on each one of these values and, therefore, each equation defined in (3.21) requires the addition of $2^{|E|}$ constraints to the problem.

ISSN: 0103–9741

3.3 Approximation

As described above, the exact representation of the nonlinear terms of the problem requires an exponential number of additional constraints and this may cause the problem to grow prohibitively large even for medium-sized instances. In this section we discuss an approximation to the problem which allows it to be solved for larger instances whilst maintaining the approximation error bounded.

3.3.1 Generation of cuts for an error tolerance threshold ε

The solution of any two-stage stochastic program is essentially related to the determination of the optimal trade-off between deterministic first-stage costs and expected (probabilistic) second-stage costs. Therefore, the quality of the optimal solution of problem (3.19) – (3.24), which results from the application of the proposed linearization technique, relies on the quality of the piecewise linear approximation of the exponential function. Given a set K of linear constraints and a solution to the corresponding problem, the absolute error (i.e., the difference between the true value of the second-stage cost function and its approximation) is equal to:

$$\sum_{s \in S} g_s(\exp(w_s) - \hat{p}_s) \tag{3.25}$$

The percentage error is obtained by dividing the absolute error by the true value of the second-stage function at a solution:

$$\frac{\sum_{s \in S} g_s \cdot (\exp(w_s) - \hat{p}_s)}{\sum_{s \in S} g_s \cdot \exp(w_s)} \tag{3.26}$$

An approximation which guarantees the maximum percentage error to be below a given tolerance level ε may be constructed based on the following proposition:

Proposition 1. Let F be the set of elements $\{a_i/b_i\}_{i=1}^{N}$ and $\varepsilon^{MAX} = \max_i\{a_i/b_i\}$, then

$$\frac{\sum_{i=1}^{N} a_i}{\sum_{i=1}^{N} b_i} \leq \varepsilon^{MAX} \tag{3.27}$$

Proof.

$$\frac{\sum_{i=1}^{N} a_i}{\sum_{i=1}^{N} b_i} \leq \frac{\sum_{i=1}^{N} \epsilon^{MAX} b_i}{\sum_{i=1}^{N} b_i} = \epsilon^{MAX} \cdot \frac{\sum_{i=1}^{N} b_i}{\sum_{i=1}^{N} b_i} = \varepsilon^{MAX} \tag{3.28}$$

■

This result ensures that provided $(\exp(w_s) - \hat{p}_s)/\exp(w_s) \leq \varepsilon, \forall s \in S$ (i.e., the percentage error of the piecewise linear approximation is less or equal to ε for each scenario) for all scenarios and possible values of w_s, then the percentage error of the approximation to the second stage cost function is also not greater than ε.

For a given set of cuts K one can easily verify in $O(|K|)$ whether the condition is satisfied (since the largest error between two adjancent cuts occurs at the point where they intersect) which allows for various heuristic/iterative methods for generating a piecewise linear approximation that guarantees that a maximum percentage error threshold is not violated. In the next sub-section, the minimum number of cuts necessary for an ε-approximation of the second stage cost function along with a method for generating them will be shown.

3.3.2 Minimum number of cuts

The following proposition establishes the minimum number of cuts necessary for an approximation of the second stage cost function whose percentage error is not greater than ε.

Proposition 2. Let $q_e^C \xi_{es} + (1 - q_e^C) \cdot (1 - \xi_{es})$ and $q_e^I \xi_{es} + (1 - q_e^I) \cdot (1 - \xi_{es})$ be the two possible values for the probability of the availability status of edge e in scenario s and $W_s = \left\{\ln(p_s^i)\right\}_{i=1}^{2^{|E|}}$ be the set of all possible values that the logarithm of the probability of scenario s may assume (given by all combinations of the product of the edges' probabilities). Also, let η^+ and η^- be, respectively, the positive and negative roots of the equation $1 - \exp(\eta) + \eta \exp(\eta) = \varepsilon$. Then, the minimum number of additional constraints necessary for an approximation which guarantees the percentage error to be less or equal to ε is given by $\sum_{s \in S} \varphi_s$, where φ_s is the optimal value of the following optimization problem:

$$\varphi_s = \quad Min \quad \sum_{j=1}^{2^{|E|}} z_j \tag{3.29}$$

$$subject\ to: \quad y_j \geq M_1(1 - z_j) - M_1 z_j \qquad \forall j = 1, \dots, 2^{|E|} \tag{3.30}$$

$$\ln(p_s^i) \geq y_j - \eta^+ - M_2(1 - x_{ij}) \qquad \forall i, j = 1, \dots, 2^{|E|} \tag{3.31}$$

$$\ln(p_s^i) \leq y_j - \eta^- + M_3(1 - x_{ij}) \qquad \forall i, j = 1, \dots, 2^{|E|} \tag{3.32}$$

$$x_{ij} \leq z_j \qquad \forall i, j = 1, \dots, 2^{|E|} \tag{3.33}$$

$$\sum_{j=1}^{2^{|E|}} x_{ij} \geq 1 \qquad \forall i = 1, \dots, 2^{|E|} \tag{3.34}$$

$$z, x \in \{0, 1\} \tag{3.35}$$

$$y \in \mathbb{R} \tag{3.36}$$

Proof.

The percentage error of the approximation provided by a cut centered at point w_0 is given by the following expression:

$$\frac{\exp(w) - (\exp(w_0) + \exp(w_0) \cdot (w - w_0))}{\exp\,(w)} \tag{3.37}$$

where $\exp(w)$ is the true value of the exponential function (i.e., the true value of the probability of occurrence of a scenario) and $(\exp(w_0) + \exp(w_0) \cdot (w - w_0))$ is the approximation provided by a cut centered on w_0 as discussed in Section 3.2.1.

By rearranging the terms, this expression may be re-written as:

$$1 - \exp(w_0 - w) + (w_0 - w) \cdot \exp(w_0 - w) \tag{3.38}$$

ISSN: 0103–9741

which is a strictly concave function and analogous to the equation defined in the Proposition by defining $\eta = w_0 - w$. Observe that the percentage error depends not on specific values of w or w_0 individually, but solely on the difference between the point in question w and the point at which the approximation is centered on, w_0. Consequently, the percentage error within the interval $\{w_0 - \eta^+, w_0 - \eta^-\}$ resulting from an approximation centered on any given point w_0 is less or equal to ε. This is illustrated in the Figures 3.4 and 3.5.

Regarding the optimization problem (3.29) – (3.36) corresponding to a given scenario s, binary variables z_j indicate the addition of a cut centered on the value of continuous variable y_j; variables x_{ij} indicate that a given point $\ln(p_s^i)$ is assigned to the cut centered on y_j, which – according to constraints (3.31) and (3.32) – can only occur if $\ln(p_s^i)$ is within the interval $\{y_j - \eta^+, y_j - \eta^-\}$ (i.e., if the approximation error at point $\ln(p_s^i)$ provided by the cut centered on y_j is less or equal to ε); M_1, M_2 and M_3 are sufficiently large positive numbers. Objective function (3.29) represents the number of cuts which are effectively needed to ensure the approximation error for all elements of the set W_s is no larger than ε. Contraint (3.30) ensures that a cut can only provide a useful approximation if the the corresponding variable z_j is properly set to 1; constraint (3.33)[1] determines that a given point $\ln(p_s^i)$ may only be assigned to a valid cut and constraint (3.34) requires each element of the set W_s to be assigned to at least one cut.

The solution of such problem determines not only the number of necessary cuts that provide an approximation for which the error is not larger than ε ($\sum_{j=1,\dots,2^{|E|}} z_j$) but also the exact points at which they should be centered on ($\{y_j | z_j = 1\}_{j=1}^{2^{|E|}}$).

■

[1] Constraint (3.26) is actually redundant, given the set of constraints (3.23) to (3.25).

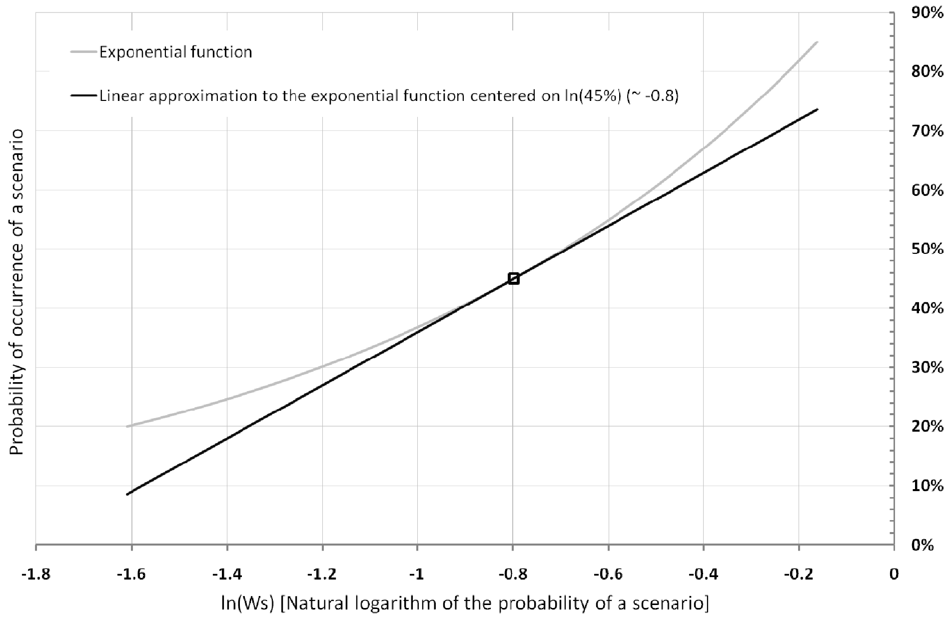


Figure 3-4 – Linear approximation to the exponential function provided by a cut centered on ln(45%)

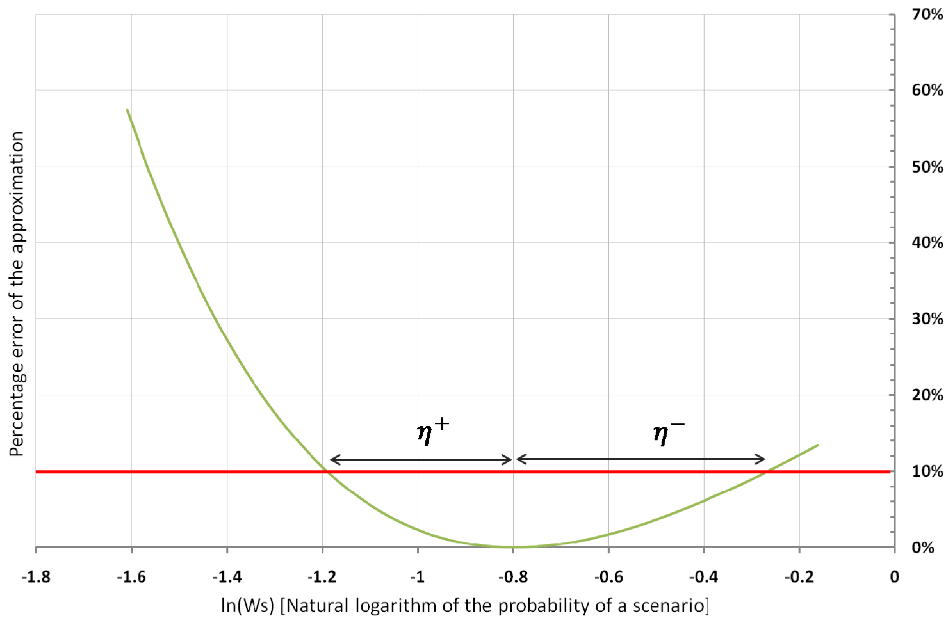


Figure 3-5 – Percentage error provided by a linear approximation centered on ln(45%) and illustration of η^+ and η^- for $\varepsilon = 10\%$

The size of each optimization problem defined in the previous Proposition, including the number of binary variables, grows exponentially with the number of edges of the graph corresponding to the transportation network of a given instance of the humanitarian logistics problem. This may cause the computational burden to be excessively large and ultimately render its solution to optimality very unlikely. Next, we discuss a relatively simpler approach to determining a set of cuts that provide an approximation that does not violate the bound on the maximum error and is much easier to compute since it does not require the full enumeration of the elements of the sets W_s.

3.3.3 An easier way to generate the cuts

Since the condition of each edge (i.e., whether each edge is active or failed) is known for each scenario, the feasible interval for each variable w_s is given by:

$$\left\{\ln\Big(\prod_{e\in E}\lambda_e^s\Big), \ln\Big(\prod_{e\in E}\mu_e^s\Big)\right\} \tag{3.39}$$

where $\lambda_e^s = \min\ \{q_e^C\xi_{es} + (1-q_e^C)\cdot(1-\xi_{es}),\ q_e^I\xi_{es} + (1-q_e^I)\cdot(1-\xi_{es})\}$ and $\mu_e^s = \max\ \{q_e^C\xi_{es} + (1-q_e^C)\cdot(1-\xi_{es}),\ q_e^I\xi_{es} + (1-q_e^I)\cdot(1-\xi_{es})\}$. The interval defined in expression (3.32) thus contains all the possible values of a given variable w_s and an approximation that ensures that the percentage error is not violated at any point within this range may be easily computed by adding the cuts corresponding to the first order Taylor's expansion of the exponential function around the points $\{\ln(\prod_{e\in E}\lambda_e^s) + \eta^+ + k\cdot(\eta^+ - \eta^-)\}_{k=0}^{\tau_s-1}$ (or, alternatively, $\{\ln(\prod_{e\in E}\mu_e^s) + \eta^- - k\cdot(\eta^+ - \eta^-)\}_{k=0}^{\tau_s-1}$), where τ_s is defined as follows:

$$\tau_s = \left\lceil\frac{(\ln(\prod_{e\in E}\mu_e^s) - \ln(\prod_{e\in E}\lambda_e^s))}{(\eta^+ - \eta^-)}\right\rceil \tag{3.40}$$

For each scenario, this procedure results in a number of inequalities which is, obviously, an upper bound to the optimal solution of the optimization problem defined in Proposition 2 and in a total number of cuts equal to:

ISSN: 0103–9741

$$\sum_{s \in S}\left(\left\lceil\frac{(\ln(\prod_{e \in E} \mu_e^s) - \ln(\prod_{e \in E} \lambda_e^s))}{(\eta^+ - \eta^-)}\right\rceil\right) \tag{3.41}$$

4 CUT GENERATION ALGORITHM

4.1 Active cuts at the optimal solution

Depending on the percentage error threshold (and on the minimum and maximum scenarios' probabilities should the approximation cuts be determined as discussed at the end of Chapter 3) the number of necessary cuts may grow to be very large, leading to computational difficulties and slower performance of solution algorithms.

However, the observation that only a small fraction of these cuts will be active at the optimal solution of problem (3.12) – (3.17) – only $|S|$ cuts represented in the set of constraints (3.15) will be actually binding – naturally points towards the design of an algorithm that dynamically generates the cuts to construct the piecewise linear approximation to the exponential function.

Next, we follow the notation and terminology of Geoffrion (1972) [27]: the value of the objective function at the optimal solution of an optimization problem $(\cdot)$ is denoted by $v(\cdot)$ and its set of feasible solutions by $F(\cdot)$. Additionally, $\cdot^*$ denotes the value of variable $\cdot$ at the optimal solution.

4.2 Solution properties

The original problem (2.1) – (2.6) and its re-formulated linear counterpart (3.12) – (3.17) have exactly the same set of feasible solutions (or, more precisely, any feasible solution to one may be mapped into the feasible solution space of the other), which may be expressed by $F(P) = F(P_2)$. In addition, if we denote the true second-stage cost function by $Q(x, \xi(x))$ and its piecewise linear approximation by $\hat{Q}(x, \xi(x))$ then, by construction, the following relation holds for all feasible x:

$$\hat{Q}\big(x, \xi(x)\big) \leq Q\big(x, \xi(x)\big), \tag{4.1}$$

ISSN: 0103–9741

Consequently, the value of the optimal solution of problem (2.1) – (2.6) will always be greater or equal to the optimal value of problem (3.12) – (3.17), i.e.:

$$v(P_2) \leq v(P) \tag{4.2}$$

4.3 Approximation of the second-stage cost function

Based on the previous remarks, the following algorithm (ALG1) may be used in order to obtain a solution to the problem for which the percentage error of the approximation of the second-stage cost function is less or equal to ε:

1 Initialize the set of cuts $K = \emptyset$, the lower bound $LB = -inf$, upper bound $UB = +inf$ and define the maximum percentage error ε
2 While $|(UB - LB)/UB| > \varepsilon$
3 Solve problem P_2 defined by (3.12) – (3.17) with the currently defined set of cuts K
4 Set $LB = v(P_2) - \sum_{e \in E} r_e x_e^* \, (= \sum_{s \in S} g_s \cdot \hat{p}_s^*)$
5 Set $UB = \sum_{s \in S} g_s \cdot \exp(w_s^*)$
6 For each scenario $s \in S$
7 Add the cut defined by $\alpha_k = \exp(w_s^*) \cdot (1 - w_s^*)$ and $\beta_k = \exp(w_s^*)$ to the cut set K
8 End For
9 End While

The algorithm works by gradually constructing a better approximation of the second stage cost function through the addition of cuts around the optimal values of variables w_s^* found at each iteration. Following the discussion in Chapter 3, the addition of a cut centered on a specific value w_s^* provides an approximation that may also be useful (i.e., for which the percentage approximation error is smaller than ε) for other possible values of the same variable which may be part of the optimal solution found in subsequent iterations.

4.3.1 Convergence analysis

The following proposition determines the maximum number of iterations of the algorithm needed in order to obtain a solution for which the percentage error of the approximated second-stage cost function relative to the true function is no larger than ε.

Proposition 3. Let η^+, λ_s^e and μ_s^e be defined as in Chapter 3, then algorithm ALG1 converges to a solution of problem $\boldsymbol{P}$ for which the percentage gap of the approximated second-stage cost function relative to its exact counterpart is less or equal to ε in a number of iterations not larger than:

$$\sum_{s \in S} \left(\left\lceil \frac{(\ln(\prod_{e \in E} \mu_e^s) - \ln(\prod_{e \in E} \lambda_e^s))}{\eta^+} \right\rceil \right) \tag{4.3}$$

Proof. As per the result of Proposition 1, if the convergence criterium of the algorithm has not been met at a given iteration i, it means that there exists at least one $s \in S$ for which $\frac{(\exp(w_s^*) - \hat{p}_s^*)}{\exp(w_s^*)} > \varepsilon$. Since it can be verified that $|\eta^+| < |\eta^-|$, this implies the fact that there exists at least one variable w_s $(s \in S)$ which satisfies the relation:

$$\left| w_s^{*(i)} - w_s^{*(j)} \right| > \eta^+, \forall j < i \tag{4.4}$$

where $w_s^{*(i)}$ denotes the value of variable w_s at the optimal solution of problem $\boldsymbol{P_2}$ solved at iteration i ($w_s^{*(j)}$ are thus the points around which the piecewise linear approximation to the exponential function has been built in previous iterations).

Let $s \in S$ be a scenario for which relation (4.4) holds and let δ be the total length of the region(s) within the feasible interval of variable w_s for which the current approximation violates the maximum percentage error threshold. The addition of a cut around the value $w_s^{*(i)}$ reduces δ by at least η^+ (and, potentially, by $\eta^+ - \eta^-$). As discussed in Chapter 3, and repeated here for convenience, the

ISSN: 0103–9741

condition of each edge is known for each scenario, thus allowing us to determine the feasible interval for each variable w_s as:

$$\left\{ \ln\left(\prod_{e \in E} \lambda_e^s\right), \ln\left(\prod_{e \in E} \mu_e^s\right) \right\} \tag{4.5}$$

At different iterations, each variable w_s ($s \in S$) may satisfy condition (4.4) at most $\lceil (\ln(\prod_{e \in E} \mu_e^s) - \ln(\prod_{e \in E} \lambda_e^s))/\eta^+ \rceil$ times – since, after that, the approximation of the exponential function over all its feasible region will be so that the maximum percentage error is less or equal to ε. The result on the maximum number of iterations of the algorithm follows naturally.

■

4.4 An algorithm considering the gap to the global optimal solution

The approximation of the second-stage cost function at the solution obtained by the algorithm presented in the previous Section is ensured to be within ε percentage points of the true function. However, the gap between the solution returned by the algorithm and the global optimal solution to the problem may be different since it depends on the first-stage cost function as well.

A slight modification to the algorithm may be introduced in order to account for the percentage gap between the solution of the problem solved using the approximation to the second-stage cost function and the global optimum, as shown below (ALG2):

1 Initialize the set of cuts $K = \emptyset$, the lower bound $LB = -inf$, upper bound $UB = +inf$ and define the maximum percentage error ε
2 While $|(UB - LB)/UB| > \varepsilon$
3   Solve problem P_2 defined by (3.12) – (3.17) with the currently defined set of cuts K
4   Set $LB = v(P_2)$
5   Set $UB_{aux} = \sum_{e \in E} r_e x_e^* + \sum_{s \in S} g_s \cdot \exp(w_s^*)$
6   If $UB_{aux} < UB$, set $UB = UB_{aux}$

For each scenario $s \in S$

Add the cut defined by $\alpha_k = \exp(w_s^*) \cdot (1 - w_s^*)$ and $\beta_k = \exp(w_s^*)$ to the cut set K

End For

End While

The algorithm above works by (*i*) obtaining a series of feasible solutions for the original problem and (*ii*) progressively perfecting the approximation of the second stage cost function at each iteration, as in ALG1.

On the one hand, the series of feasible solutions provide a monotonically decreasing sequence of upper bounds. On the other hand, the series of values of the objective function at the optimal solution of the approximated problem solved at each iteration constitutes a monotonically increasing sequence of lower bounds, since $\hat{Q}_{i+1}\big(x, \xi(x)\big) \geq \hat{Q}_i\big(x, \xi(x)\big)$ for all feasible x (where $\hat{Q}_i\big(x, \xi(x)\big)$ denotes the piecewise linear approximation of the second stage cost function at iteration i).

In this case, a simple upper bound on the number of iterations until the convergence of the algorithm is given by $|S| \cdot 2^{|E|}$, which would correspond to a complete enumeration of the linear constraints that provide an exact representation of the exponential function at all possible values of each variable w_s.

ISSN: 0103–9741

5 SCENARIO GENERATION

5.1 Difficulty in scenario generation

While the number of possible network realizations is computationally tractable, the algorithm presented in Chapter 4 may be used in order to obtain a solution which is within a tolerance level ε from the global optimum of the original problem. However, if one wants to be able to solve large-scale problems, it becomes imperative to have an estimate of the expected value of the second stage cost function which is not based on the complete enumeration of all possible network configurations.

Standard two-stage stochastic programming models usually resort to scenario generation to allow for the evaluation of these multi-dimensional integrals. However, unlike the vast majority of problems studied in the literature, in the humanitarian logistics problem – and, more generally, in the class of problems presented in Section 1.5 – the probability distribution of the random variables is not known before first-stage decisions are determined.

As already pointed out in Section 2.3, this makes it impossible to utilize traditional scenario generation methods such as Monte Carlo sampling, moment matching or minimization of distances between probability measures. In this work, we propose to overcome this obstacle by merging the concepts from importance sampling into a stochastic programming framework, as presented next.

5.2 Importance sampling

In statistics, importance sampling is a technique used to estimate the properties of a certain distribution while only having samples drawn from a different one. In the context of simulation studies, importance sampling is usually employed as a variance reduction technique used in conjunction with the Monte Carlo method. The basic idea is that certain values of the random variable may have a stronger effect upon the parameter being estimated than others, so it might be interesting to sample these values more frequently than what would otherwise be expected based on the original probability distribution.

As detailed in Rubinstein (1981) [51], the method relies on a simple observation to compute the expected value of a random variable $X{\sim}F_1(x)$ based on samples from another distribution $F_2(x)$:

$$\mathbb{E}_{f_1}\{x\} = \int_x xf_1(x)dx = \int_x x\frac{f_1(x)}{f_2(x)}f_2(x)dx = \mathbb{E}_{f_2}\left\{x\frac{f_1(x)}{f_2(x)}\right\} \tag{5.1}$$

For a given set of samples x_i $(i = 1, \dots, N)$ drawn according to a probability density function $f_2(X)$, the importance sampling estimator of the mean of distribution $f_1(X)$ is then defined as:

$$\hat{\mu}_X^{IS} = \frac{1}{N}\sum_{i=1}^{N} x_i \cdot \frac{f_1(x_i)}{f_2(x_i)} \tag{5.2}$$

Following expression (5.1), each sample is weighted differently based on the likelihood ratio, i.e. the ratio between the probability of occurrence of that sample under the distribution of interest and the one from which the samples were drawn.

Again according to [51], this estimator is proved to be consistent – it converges to μ_X with probability 1 as the sample size grows to infinity – and unbiased – its expected value is μ_X, whatever the sample size. In the next section, this technique is incorporated into the optimization problem so as to allow for the estimation of the second stage cost function based on scenarios.

5.3 Reformulation

Although the final (*post*-investment) probability distribution of the availability of the edges is not known *a priori*, the initial distribution (i.e., the one which does not consider any reinforcement investments) may be used to generate scenarios of network configuration, for which the probability of occurrence may be easily calculated. This is also the case of the more general class of stochastic programming problems with endogenous uncertainty defined in Chapter 2: the initial probability distribution of the random variables is always known, even though it might change after first-stage decisions are determined.

Additionally, since the linearization technique proposed in Chapter 3 makes it possible to compute the probability of occurrence of any scenario given the first-stage investment decisions (or, at least, an approximation to its value), we may join these pieces of information in order to compute the importance sampling estimator of the expected value of the second stage cost function.

By examining expression (5.2) for the importance sampling estimator, we may identify the corresponding elements of the optimization problem being studied: $f_1(x)$ and $f_2(x)$ are, respectively, the final and initial probability density functions of the scenarios, N is obviously the number of sampled scenarios and the samples x_i represent the values of the scenario-specific second-stage problems which are solved separately, as discussed in Chapter 3. Once again, it is important to stress that the scenarios of network realization are to be sampled according to the initial probability distribution of the edges' availabilities.

This analogy allows us to reformulate problem (3.12) – (3.17) in a way which does not require the full enumeration of all possible network configurations but relies on a smaller subset of randomly generated scenarios, as shown below:

$$(\boldsymbol{P_3}) \quad Min \quad \sum_{e \in E} r_e x_e + \frac{1}{|S|} \sum_{s \in S} g_s\left(\frac{\hat{p}_s}{p_s^{INI}}\right) \tag{5.3}$$

$$subject\ to: \quad Ax \leq b \tag{5.4}$$

$$w_s = \sum_{e \in E} \{\ln(\mathrm{p}_{es}^C) + [\ln(\mathrm{p}_{es}^I) - \ln(\mathrm{p}_{es}^C)] \cdot x_e\} \qquad \forall s \in S \tag{5.5}$$

$$\hat{p}_s \geq \alpha_k + \beta_k \cdot w_s \qquad \forall s \in S, \forall k \in K \tag{5.6}$$

$$\hat{p} \in \mathbb{R}^+, w \in \mathbb{R} \tag{5.7}$$

$$x \in \{0,1\}^{|E|} \tag{5.8}$$

where:

p_s^{INI} — probability of sampled scenario s, calculated based on the initial probability distribution of the availability of each edge, i.e. $p_s^{INI} = \prod_{e \in E} p_{es}^C$

ISSN: 0103–9741

Based on a set of scenarios of network realizations, sampled according to the initial probability distribution of the edges' availabilities, a solution to problem (5.3) – (5.8) may be found using the algorithm outlined in Chapter 4.

5.4 Solution robustness

As with any two-stage stochastic program, the solution to these problems depends, essentially, on balancing the trade-off between deterministic first-stage costs and the expected value of probabilistic second-stage costs. It is thus imperative that we have a reasonable estimate of second stage costs in order to be able to have confidence in the quality of the solution obtained.

On the one hand, the larger the set of sampled scenarios, the better the estimate of second-stage expected costs will be. On the other hand, having fewer scenarios makes the problem smaller and solution times are usually faster. Anyhow, once a solution is found for a given set of scenarios, a Monte Carlo simulation – in which the probability distribution of the edges' availabilities takes into account the determined first-stage decisions – may then provide a confidence interval against which the estimate of the expected costs of the second-stage provided at the solution of the problem can be compared in order to assess the need for a larger number of samples. This is discussed in Annex B, where an algorithm for determining an adequate number of scenarios is described.

6 COMPUTATIONAL RESULTS

Computational tests were perfomed to analyze the performance of the proposed reformulation schemes and solution algorithm. All testes were conducted on a computer with processor Pentium 4, 3.00 GHz and 2 GB of RAM. The models and algorithms were implemented using the modeling language MOSEL and solved by XPRESS 19.00.04.

The first results are those obtained for the set of instances described in Viswanath et al. [66]. These are all small-size problems which served as a “proof of correctness” for the proposed methodology. Since no other work in the literature deals with the problem in its original form (remember that [43] dismisses the probabilistic nature of the problem by assuming that investment on an edge completely eliminates the probability of that edge failing afterwards), several other instances were created in order to assess the performance of the methodology for medium and large-size instances of the problem.

The remainder of this Chapter is organized as follows: Section (6.1) presents the results for the instances provided in [66], Section (6.2) describes how the medium and large-size instances were generated and presents results for the former while Section (6.3) discusses the results for the latter.

6.1 Instances from the literature

All the instances solved in [66] refer to a graph which contains 4 nodes and 5 edges, as depicted in Figure 6.1. There is a total of 28 instances which are detailed in Table 6.1: they differ from each other in the investment and transportation costs associated with each edge (columns *InvCost* and *TranspCost*, respectively), maximum budget (column *Budget*), penalty for not fulfilling the demand associated to a node (column *Penalty*) and initial and final survival probabilities (initial survival probability is equal to 70% for all edges in instances 1 through 14 and equal to 60% in instances 15 through 28 and column *SurvProbInv*). Nodes *O* and *D* are the origin and destination for a unit commodity that must flow through the network.

ISSN: 0103–9741

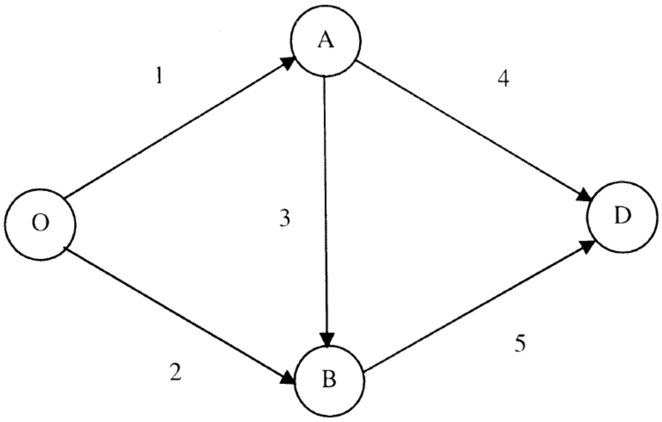


Figure 6-1 – Graph corresponding to the instances solved in Viswanath et al. [66]

For all these instances, there are 32 (2^5) scenarios of network configuration – given by all the possible combinations of the availability of the edges – and the first step of the proposed methodology determines that the minimum cost network flow problem corresponding to each one of these configurations must be solved independently. For this set of instances, total solution time of the network flow problems for all scenarios is minuscule.

Once these optimal values are known, they are used as coefficients in the objective function of the main problem, which is then solved by the algorithm outlined in Chapter 4. All instances were solved to optimality in less than 1.0 second and average solution time was 0.313 second. Details are provided in Table 6.2 where the column *Id* indicates the instance identification, column *OptVal* presents the value of the optimal solution , column *# Iter* indicates the number of iterations of the algorithm until convergence was achieved and column *TotalTime* the time it took for the algorithm to complete.

ISSN: 0103–9741

Id	***SurvProbInv***	***InvCost***	***Budget***	***TranspCost***	***Penalty***
1 / 15	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	2	{10, 10, 10, 10, 10}	31
2 / 16	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 10, 10, 10, 10}	31
3 / 17	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 10, 15, 30, 10}	41
4 / 18	{80%, 80%, 90%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 10, 15, 30, 10}	41
5 / 19	{80%, 80%, 80%, 80%, 80%}	{2, 1, 1, 1, 1}	3	{10, 10, 15, 30, 10}	41
6 / 20	{80%, 80%, 80%, 80%, 80%}	{1, 2, 1, 1, 1}	3	{10, 10, 15, 30, 10}	41
7 / 21	{80%, 80%, 80%, 80%, 80%}	{1, 1, 2, 1, 1}	3	{10, 10, 15, 30, 10}	41
8 / 22	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 2, 1}	3	{10, 10, 15, 30, 10}	41
9 / 23	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 2}	3	{10, 10, 15, 30, 10}	41
10 / 24	{80%, 80%, 80%, 80%, 80%}	{2, 1, 1, 1, 1}	3	{10, 20, 10, 15, 10}	31
11 / 25	{80%, 80%, 80%, 80%, 80%}	{2, 1, 1, 1, 1}	3	{10, 20, 10, 15, 10}	43.9
12 / 26	{80%, 80%, 80%, 80%, 80%}	{2, 1, 1, 1, 1}	3	{10, 20, 10, 15, 10}	57.3
13 / 27	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 15, 5, 15, 10}	26
14 / 28	{80%, 80%, 80%, 80%, 80%}	{1, 1, 1, 1, 1}	3	{10, 15, 1, 15, 10}	26

Table 6-1 – Description of the instances provided in Viswanath et al. [66]

ISSN: 0103–9741

Id	*OptVal*	*# Iter*	*TotalTime*
1	21.9961	3	0.137
2	21.7155	2	0.082
3	26.8835	3	0.123
4	26.8494	4	0.225
5	26.9087	3	0.120
6	26.9681	3	0.106
7	26.8835	3	0.136
8	26.8835	3	0.122
9	26.9681	3	0.129
10	26.9601	4	0.257
11	29.0251	3	0.144
12	31.0963	3	0.297
13	25.1315	5	1.000
14	23.0995	3	0.359
15	22.5114	3	0.302
16	22.0285	2	0.187
17	26.9725	3	0.359
18	26.9638	3	0.531
19	27.0157	3	0.359
20	27.1194	3	0.375
21	26.9725	3	0.360
22	26.9725	3	0.421
23	27.1194	3	0.359
24	27.0074	3	0.422
25	28.8943	3	0.421
26	32.0447	3	0.375
27	25.1565	3	0.625
28	23.1405	2	0.235

Table 6-2 – Results of the instances provided in Viswanath et al. [66]

ISSN: 0103–9741

6.2 Medium-size instances

Given the lack of additional instances of the problem available in the literature, we developed an instance generator which was then used to test the proposed methodology.

The instances were created by randomly selecting the location of a given number of nodes within a region defined by minimum and maximum values for the x and y coordinates. Next, a predefined number of edges connecting the nodes was created (the resulting graph was checked for connectedness in order to avoid trivial and meaningless solutions) and the Euclidean distance between the corresponding nodes was assigned as the transportation cost of each edge. Pre- and post-investment survival probabilities were assigned to each edge and, for the large instances presented in Section 6.3, scenarios of network configuration were generated based on the initial survival probability of each edge.

Next, in Tables 6.3 and 6.4, we present the results for a total of 30 instances which were all solved by the algorithm designed in Chapter 4 (ALG2) with full scenario enumeration and tolerance level set to no more than 1%. The table provides the following information: column *Id* identifies the instance, columns *# Nodes* and *# Edges* indicates, respectively, the number of nodes and edges of the graph, column *# Scen* provides the number of scenarios of network configuration used in each problem; column *UB* reports the value of the best solution found while column *LB* indicates the value of the solution to the last approximated problem (i.e., the one which is solved by considering the set of cuts that approximate the exponential function), column *% Gap* presents the percentage gap between the upper and lower bounds and column *ErrTol* contains the maximum acceptable error, which is the stopping criterium for the algorithm; column *# Iter* indicates the number of iterations of the algorithm needed to reach the final solution, and column *MainTime* report the total time for the convergence of the algorithm (the time needed for the solution of the independent scenario-specific network flow problems is not reported but they are usually orders of magnitude smaller than the time it takes for the algorithm to converge which thus represents the bottleneck of the methodology).

Id	*# Nodes*	*# Edges*	*# Scen*	*UB*	*LB*	*% Gap*	*ErrTol*	*# Iter*	*MainTime*
v5e6A	5	6	64	219.423	219.423	0.00%	0.1%	5	0.437
v5e6B	5	6	64	50.6467	50.6467	0.00%	0.1%	3	0.140
v5e6C	5	6	64	143.923	143.923	0.00%	0.1%	4	0.281
v5e6D	5	6	64	107.918	107.918	0.00%	0.1%	4	0.250
v5e6E	5	6	64	277.904	277.904	0.00%	0.1%	4	0.281
v6e8A	6	8	256	361.265	361.265	0.00%	0.1%	6	1.796
v6e8B	6	8	256	45.9159	45.9159	0.00%	0.1%	5	1.328
v6e8C	6	8	256	350.268	350.268	0.00%	0.1%	5	1.594
v6e8D	6	8	256	110.915	110.915	0.00%	0.1%	5	1.062
v6e8E	6	8	256	65.4201	65.4201	0.00%	0.1%	3	0.328
v7e10A	7	10	1024	122.857	122.851	0.0049%	0.1%	4	3.735
v7e10B	7	10	1024	201.934	201.926	0.0040%	0.1%	5	4.468
v7e10C	7	10	1024	104.863	104.857	0.0057%	0.1%	4	3.063
v7e10D	7	10	1024	158.868	158.861	0.0044%	0.1%	5	4.063
v7e10E	7	10	1024	75.5659	75.5619	0.0053%	0.1%	5	5.704

Table 6-3 – Results for the medium-size instances

Id	*# Nodes*	*# Edges*	*# Scen*	*UB*	*LB*	*% Gap*	*ErrTol*	*# Iter*	*MainTime*
v7e11A	7	11	2048	306.692	306.672	0.0065%	0.1%	8	79.143
v7e11B	7	11	2048	252.026	251.985	0.0163%	0.1%	6	39.344
v7e11C	7	11	2048	66.7778	66.7561	0.0325%	0.1%	5	15.593
v7e11D	7	11	2048	312.985	312.959	0.0083%	0.1%	6	38.875
v7e11E	7	11	2048	58.0173	57.9977	0.0338%	0.1%	5	27.859
v8e12A	8	12	4096	31.9086	31.5927	0.99%	1%	11	405.395
v8e12B	8	12	4096	141.750	141.515	0.1658%	1%	6	242.878
v8e12C	8	12	4096	97.1507	97.0042	0.1508%	1%	4	40.219
v8e12D	8	12	4096	49.6668	49.4192	0.4985%	1%	7	286.253
v8e12E	8	12	4096	155.492	155.381	0.0714%	1%	4	71.297
v8e12b6A	8	12	4096	40.8165	40.6271	0.4640%	1%	6	247.645
v8e12b6B	8	12	4096	28.5302	28.2736	0.8994%	1%	8	831.795
v8e12b6C	8	12	4096	22.3931	22.1859	0.9253%	1%	7	449.365
v8e12b6D	8	12	4096	66.6392	66.3808	0.3878%	1%	8	992.796
v8e12b6E	8	12	4096	105.355	104.637	0.6815%	1%	5	113.814

Table 6-4 – Results for the medium-size instances

6.3 Large-size instances

The set of instances in Section 6.2 involved graphs with a maximum of twelve edges and 4096 possible scenarios of network configuration. The total time required to solve these problems clearly shows how the computational effort increased very rapidly with respect to the number of edges – just as an illustration of this fact, the average time needed to solve the instances with 11 edges was 40.2 seconds, while the average time consumed by the algorithm in solving the instances with 12 edges was 368.1 seconds.

A critical example is provided by an instance of the problem with 10 nodes and 15 edges (and, consequently, 32768 possible scenarios of network configuration) which was solved by full scenario enumeration. The Figure below presents the performance of the algorithm – data points represent the upper and lower bounds obtained at each iteration:

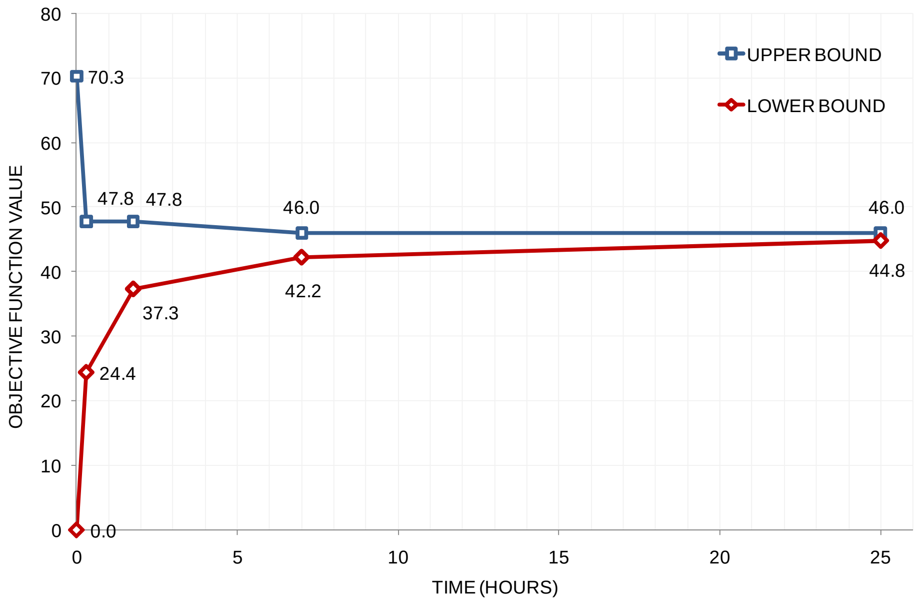


Figure 6-2 – Algorithm perfomance on an 15-edge instance with full scenario enumeration

While the previous instances converged to solutions with gaps not larger than 1% after no more than 17 minutes, in the case of the 15-edge instance it took a total of 25 hours for the algorithm to narrow the gap down to 2.57%. This clearly leads to the conclusion that full scenario enumeration is currently not a viable option when one tries to solve large scale problems and a sample-based version of the problem – such as that suggested in Chapter 5 – becomes a necessity.

In Table 6.5, we present results for 18 instances with the number of edges ranging from 15 to 40, also constructed according to the description given in the previous Section. All instances were solved to a maximum gap of 0.87%. Compared to Table 6.4, there is an additional column *# TotScen* where the number of possible scenarios of network configuration is reported – column *# Scen* indicates the number of scenarios actually used when solving the problem.

The instances with 15 edges (*v10e15_1*, *v10e15_2* and *v10e15_3*) all refer to the same graph of the example for which the convergence of the algorithm was shown in Figure 6.2. Each one of them was solved using a different set of 500 scenarios (out of the 32768 possible network configurations), sampled according to the initial probability distribution of the edges' availabilities. It is interesting to observe that even though the number of scenarios used in these instances is significantly smaller than the total number of possible scenarios, the solutions found for these problems in under 60 seconds have an objective function value which is close to that found after 25 hours in the case of full scenario enumeration.

A significant increase in computational times was observed when solving the instances with 20 edges (*v10e20_1* through *v10e20_5*) and 500 scenarios. All instances refer to the same graph and were solved using different sets of scenarios to a maximum gap of 0.868%, including one instance which was solved to optimality. The difference between the minimum and maximum optimal values obtained for all five problems was 5.49%, which seems like a reasonable compromise considering that the number of scenarios actually used to solve the problems represents a very small fraction (namely, 0.048%) of all possible network configurations.

As the number of binary variables increase so does the computational effort required to solve the problems at each iteration, leading to a compromise between network size (which dictates the number of binary variables) and number of sampled scenarios. This was done when solving the instances with 25, 30 and 40 edges: the number of scenarios utilized was reduced to 300 in the former and 200 in the last two cases.

ISSN: 0103–9741

Id	*# Nodes*	*# Edges*	*# Scen*	*# TotScen*	*UB*	*LB*	*% Gap*	*ErrTol*	*# Iter*	*MainTime*
v10e15_1	10	15	500	3.28E+4	44.5735	44.4068	0.374%	1%	6	41.032
v10e15_2	10	15	500	3.28E+4	44.8405	44.8405	0%	1%	6	27.562
v10e15_3	10	15	500	3.28E+4	47.0452	46.9256	0.254%	1%	6	27.578
v10e20_1	10	20	500	1.05E+6	81.8326	81.7279	0.128%	1%	8	1169.703
v10e20_2	10	20	500	1.05E+6	81.4761	80.7692	0.868%	1%	9	2725.251
v10e20_3	10	20	500	1.05E+6	81.6981	81.3323	0.448%	1%	10	1713.125
v10e20_4	10	20	500	1.05E+6	78.4703	78.1662	0.388%	1%	10	3164.391
v10e20_5	10	20	500	1.05E+6	77.3371	77.3371	0%	1%	10	4028.000
v12e25_A	12	25	300	3.36E+7	75.5378	74.9683	0.754%	1%	8	2528.266
v12e25_B	12	25	300	3.36E+7	52.4450	52.1022	0.654%	1%	12	3133.67
v12e25_C	12	25	300	3.36E+7	70.3214	70.1937	0.182%	1%	11	1882.297
v12e25_D	12	25	300	3.36E+7	43.1088	42.931	0.412%	1%	8	810.234
v13e30_1	13	30	200	1.07E+9	32.4429	32.3264	0.359%	1%	9	516.454
v13e30_2	13	30	200	1.07E+9	38.8679	38.6852	0.470%	1%	11	6332.030
v13e30_3	13	30	200	1.07E+9	32.4177	32.4131	0.014%	1%	7	1086.485
v13e30_4	13	30	200	1.07E+9	33.1784	32.9004	0.838%	1%	9	1095.703
v13e30_5	13	30	200	1.07E+9	34.4604	34.1707	0.841%	1%	9	3457.325
v16e40_1	16	40	200	1.10E+12	19.5682	19.4753	0.475%	1%	7	4367.515

Table 6-5 – Results for the large-size instances

7 CONCLUSIONS

While there has been a huge amount of work on stochastic programs with exogenous uncertainty, the same is not true for the case where the underlying stochastic processes depend on the decisions taken.

This works aims at contributing to fill this gap by studying a problem in the area of humanitarian logistics. The proposed re-formulation scheme overcomes the non-linearities that arise in the original formulation presented in the literature and the incorporation of the importance sampling concepts allows us to solve large instances of the problem – which would otherwise be untractable – by using sample scenarios even though the final probability distribution of the random variables is not known *a priori*.

The proposed approach was able to solve all the instances available in the literature in very short time. Additionally, larger instances of the problem were created – and will be made publicly available – in order to assess the performance of the developed algorithms. Medium-size instances were solved within reasonable times (each one of them was solved in under 17 minutes) and solutions provably within 1% of the global optimal have been found. Large-size instances were solved using samples of scenarios of network configuration and solutions were also within 1% of the global optimal. Considering that the only article that deals with the exact same problem studied in this work used instances with only 5 edges, an eight-fold improvement has been obtained.

7.1 Future work and extensions

Regarding improvements on the speficic problem discussed in this work, there are some issues that can probably be dealt with more efficiently:

a) The problem solved at each iteration is very similar to that of the previous one – the only difference being the |S| cuts added to improve the approximation of the exponential function. Instead of solving each problem from scratch, the previously obtained solution may provide useful information which may then be used to reduce computational times.

b) Valid cuts and/or tailor-made heuristics may be identified/developed to decrease the computational effort related to the number of binary variables.

A decrease in the computational time required to solve the problem of each iteration may allow for an increase in the number of scenario samples used to solve large-scale problems, thus providing better-quality solutions.

Additionally, it would also be of interest to carry out both in-sample and out-of-sample (following the algorithm suggested in Annex B) analyses of the problem in order to empirically determine the adequate number of sample scenarios in each case. In this sense, the incorporation of the importance sampling concepts could be further explored by using scenario samples drawn from probability distributions other than the original ones and adapting the problem accordingly.

Five main future(/current) work alternatives lie ahead:

a) Extend the methodology to consider the coupling between first and second stage variables to be given not only by the probability of occurrence of scenarios but also by the feasible region of second stage problems.

b) Adapt the methodology to consider the existence of a correlation structure between the random variables.

c) Incorporation of risk measures and probabilistic constraints (as in the context of chance-constrained programming).

d) Explore the proposed linearization technique in the context of the unconstrained quadratic 0–1 minimization problem.

e) Study the applicability and performance of the proposed methodology to other problems of the same nature, such as those described in Chapter 1: stochastic queuing networks, stochastic PERT and revenue management.

ANNEX A: Generalization with respect to second-stage problems

An essential part of the methodology proposed in this work – namely the linearization of the product of binary variables using the properties of the logarithm – lies on the piecewise linear approximation of the exponential function which may be represented in the optimization problem as linear constraints. However, if the objective function value of the optimal solution of a second stage problem is negative, the first stage problem clearly becomes unbounded and this would represent a limitation to the applicability of the concepts discussed afterwards.

This Annex describes how to address this situation so that the methodology remains valid, regardless of the values of the optimal solutions of the second-stage problems. In summary, the next two sections discuss that a constant term may be added to the value of the optimal solution of all second stage problems without affecting the solution of the original problem.

A.1 Problems solved with full scenario enumeration

In the case where one is able to enumerate all the possible scenarios of network configuration, the objective function (3.12) may be re-written as follows:

$$\sum_{e \in E} r_e x_e + \sum_{s \in S} \bar{g}_s \hat{p}_s - \bar{g} \qquad (9.1)$$

where $\bar{g}_s = g_s + \bar{g}$ and $\bar{g}$ is such that $\bar{g}_s > 0, \forall s \in S$. Expression (9.1) is clearly equivalent to:

$$\sum_{e \in E} r_e x_e + \sum_{s \in S} g_s \hat{p}_s + \sum_{s \in S} \bar{g} \cdot \hat{p}_s - \bar{g} \qquad (9.2)$$

On the one hand, the algebraic sum of the third and fourth terms of expression (9.2) will always amount to zero since $\sum_{s \in S} \hat{p}_s = 1$. On the other hand, the sum of the first and second terms above is the exact expression of the original objective function. The alternate objective function (9.1) assumes the exact same values as the original one (3.12) for all feasible values of the decision variables and, consequently, problem (3.12) – (3.17) is equivalent to problem (9.1) – (3.13) – (3.17).

A.2 Problems solved with a sample of scenarios

Following the proposed methodology, large-scale problems are solved using the formulation (5.3) – (5.8). Using the same rationale as above, the objective function (5.3) may be re-written as:

$$\sum_{e \in E} r_e x_e + \frac{1}{|S|} \sum_{s \in S} \bar{g}_s \left(\frac{\hat{p}_s}{p_s^{INI}} \right) - \bar{g} \tag{9.3}$$

which is equivalent to:

$$\sum_{e \in E} r_e x_e + \frac{1}{|S|} \sum_{s \in S} g_s \left(\frac{\hat{p}_s}{p_s^{INI}} \right) + \frac{1}{|S|} \sum_{s \in S} \bar{g} \left(\frac{\hat{p}_s}{p_s^{INI}} \right) - \bar{g} \tag{9.4}$$

If we denote by D the set of distinct scenarios in the sample and by n_d the number of occurrences of each one of them, the third term may be written as:

$$\bar{g} \sum_{d \in D} \frac{n_d}{|S|} \left(\frac{\hat{p}_d}{p_d^{INI}} \right) \tag{9.5}$$

As $|S| \to \infty$, $\frac{n_d}{|S|} \to p_d^{INI}$ and expression (9.5) converges to $\bar{g} \cdot \sum_{d \in D} \hat{p}_d$. Since $\sum_{d \in D} \hat{p}_d \to 1$, the result is analogous to that obtained in the previous Section.

ISSN: 0103–9741

ANNEX B: Solution robustness

The solution of two-stage stochastic programs depends, essentially, on balancing the trade-off between deterministic first-stage costs and the expected value of probabilistic second-stage costs. It is thus imperative that we have a reasonable estimate of second stage costs in order to be able to have confidence in the quality of the solution obtained.

On the one hand, the larger the set of sampled scenarios, the better the estimate of second stage costs will be. On the other hand, having fewer scenarios makes the problem smaller and solution times are usually faster. Anyhow, once a solution is found for a given set of scenarios, a Monte Carlo simulation – in which the probability distribution of the edges' availabilities takes into account the determined first-stage decisions – may then provide a confidence interval against which the estimate of the expected costs of the second-stage can be compared in order to assess the need for a larger number of samples. This suggests the following algorithm, detailed below:

1	Initialize the set of cuts $K = \emptyset$, define the maximum percentage error ε and the confidence level for the estimator of the mean second stage costs (expressed in terms of the number of standard deviations θ)
2	Initialize the lower bound $LB = -inf$, upper bound $UB = +inf$
3	While $\hat{\mu}_S \notin \{\hat{\mu}_M - \theta \cdot \sigma_{\hat{\mu}_M}, \hat{\mu}_M + \theta \cdot \sigma_{\hat{\mu}_M}\}$
	Generate a new sample S_{aux} of network configuration scenarios based on the initial probability distribution of the edges' availabilities
	For each scenario $s \in S_{aux}$
	Solve problem (3.1) – (3.4) and obtain the corresponding value g_s
	End For
4	Set $S = S \cup S_{aux}$
5	While $\lvert (UB - LB)/UB \rvert > \varepsilon$
6	Solve problem P_3 defined by (5.3) – (5.18) with the currently defined set of cuts K and scenario sample set S

ISSN: 0103–9741

Set $LB = v(P_3)$

Set $UB_{aux} = \sum_{e \in E} r_e x_e^* + \frac{1}{|S|} \sum_{s \in S} g_s \left(\frac{\exp(w_s^*)}{p_s^{INI}} \right)$

If $UB_{aux} < UB$, set $UB = UB_{aux}$

For each scenario $s \in S$

Add the cut defined by $\alpha_k = \exp(w_s^*) \cdot (1 - w_s^*)$ and $\beta_k = \exp(w_s^*)$ to the cut set K

End For

End While

Compute $\hat{\mu}_S = \frac{1}{|S|} \sum_{s \in S} g_s \left(\frac{\exp(w_s^*)}{p_s^{INI}} \right)$

Generate a sample M ($|M| \gg |S|$) of network configuration scenarios based on the probability distribution of the edges' availabilities which results from the determined first stage decisions

For each scenario $m \in M$

Solve problem (3.1) – (3.4) and obtain the corresponding value g_m

End For

Obtain the estimator of the mean of second stage costs and its standard deviation: $\hat{\mu}_M$ and $\sigma_{\hat{\mu}_M}$, respectively

End While

REFERENCES

[1] Adams, W.P., Forrester, R.J. (2005) A simple recipe for concise mixed 0-1 linearizations. Operations Research Letters 33 (1), pp. 55–61.

[2] Ahmed, S. (2000) Strategic planning under uncertainty: Stochastic integer programming approaches. PhD thesis, University of Illinois at Urbana-Champaign.

[3] Al-qurashi, F. (2004) New vision of emergency response planning. Process Safety Progress 23 (1), 56–61.

[4] Balas, E., Mazzola, J.B. (1984) Nonlinear 0–1 programming: 1. linearization techniques. Mathematical Programming 30, pp. 1–21.

[5] Balas, E., Mazzola, J.B. (1984) Nonlinear 0–1 programming: 2. dominance relations and algorithms. Mathematical Programming 30, pp. 22–45.

[6] Bana e Costa, C., Oliveira, C., Vieira, V. (2008) Prioritization of bridges and tunnels in earthquake risk mitigation using multicriteria decision analysis – application to Lisbon. Omega, Special Issue on Multiple Criteria Decision Making for Engineering, Vol. 36, Issue 3, pp. 442-450.

[7] Basoz, N., Kiremidjian, A. (1995) Prioritization of Bridges for Seismic Retrofitting. Buffalo, NY, U.S. National Center for Earthquake Engineering Research, Technical Report NCEER, 95-0007, 150 p.

[8] Beale, E. M. L. (1955) On minimizing a convex function subject to linear inequalities. Journal of the Royal Statistics Society Series B 17, pp. 173–184.

[9] Ben-Tal, A., A. Nemirovski. (1998) Robust convex optimization. Mathematics of Operations Research 23, pp. 769–805.

[10] Ben-Tal, A. and Nemirovski, A. (1998): On the quality of SDP approximations of uncertain SDP programs, Research Report #4/98, Optimization Laboratory, Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, Israel.

[11] Ben-Tal, A. and Nemirovski, A. (1999): Robust solutions to uncertain programs, Oper. Res. Letters, 25, 1-13.

[12] Ben-Tal, A. and Nemirovski, A. (2000): Robust solutions of Linear Programming problems contaminated with uncertain data, Math. Program., 88, 411-424.

[13] Bertsimas, D., M. Sim. (2003) Robust discrete optimization and network flows. Mathematical Programming 98, pp. 49–71.

[14] Bertsimas, D., Sim, M., Pachamanova, D. (2004) Robust Linear Optimization under General Norms, Operations Research Letters, 32, 510-516.

[15] Bertsimas, D., Sim, M. (2004) The price of Robustness, Operations Research, 52, 1, pp. 35-53.

[16] Birge, J., Louveaux, F. (1997) Introduction to Stochastic Programming. Springer Series in Operations Research and Financial Engineering.

[17] Chang, C.-T. (2000) Efficient linearization approach for mixed-integer problems. European Journal of Operational Research 123 (3), pp. 652–659.

[18] Chang, C.-T., Chang, C.-C. (2000) A linearization method for mixed 0-1 polynomial programs. Computers & Operations Research 27 (10), pp. 1005–1016.

[19] Cooper, J.D., Friedland, I.M., Buckle, I.G., Nimis, R.B. and Bobb, N.M. (1994). The Northridge Earthquake: Progress Made, Lessons Learned and Seismic-Resistant Bridge Design, Public Roads 58 (1).

[20] Dantzig, G. B. (1955) Linear programming under uncertainty. Management Science 1, pp. 197–206.

[21] Dupacová, J., Hurt, J. and Stepán, J. "Stochastic Modeling in Economics and Finance" Applied Optimization Vol. 75, Kluwer, 2002.

[22] El-Ghaoui, L. and Lebret, H. (1997): Robust solutions to least-square problems to uncertain data matrices, SIAM J. Matrix Anal. Appl., 18, pp. 1035-1064.

[23] El-Ghaoui, L., Oustry, F. and Lebret, H. (1998) Robust solutions to uncertain semidefinite programs, SIAM Journal of Optimization 9, pp. 33-52.

[24] Fan, Y., Liu, C. (2009) Solving Stochastic Transportation Network Protection Problems Using the Progressive Hedging-based Method. @JNSE – COMPLETAR REFERENCIA

[25] Fiedrich, F., Gehbauer, F., Rickers, U. (2000) Optimized resource allocation for emergency response after earthquake disasters. Safety Science 35 (1), pp. 41–57.

[26] Gaivoronski, A.A.: “Stochastic Optimization Problems in Telecommunications”. “Applications of Stochastic Programming”, MPS-SIAM Book Series on Optimization 5, Chapter 32, Edited by S.W.Wallace and W.T.Ziemba, 2005.

[27] Geoffrion, A.M. (1974) Lagrangean Relaxation for Integer Programming. Mathematical Programming Study 2, pp. 82-114.

[28] Globo Newspaper. Haitianos revoltados com demora na entrega de ajuda bloqueiam estradas com corpos após terremoto. http://oglobo.globo.com/ mundo / mat / 2010 / 01 / 15 / haitianos-revoltados-com-demora-na-entrega-de-ajuda-bloqueiam-estradas-com-corpos-apos-terremoto-915533088.asp

[29] Glover, F. (1975) Improved linear integer programming formulations of nonlinear integer problems, Management Science 22, pp. 455–460.

[30] Goel, V., Grossmann, I. E. (2004) A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. Computers and Chemical Engineering 28 (8), pp. 1409–1429.

[31] Goel, V., Grossmann, I. E. (2006) A Class of Stochastic Programs with Decision Dependent Uncertainty. Mathematical Programming, Series B

[32] Gueye, S., Michelon, P. (2005) “Miniaturized linearizations for quadratic 0–1 problems” Annals of Operations Research 140, pp. 235–261.

[33] Haneveld, W.K., van der Vlerk, M. (2005) Stochastic Programming Lecture Notes.

[34] Hansen, P., Meyer, C. (2009) Improved compact linearizations for the unconstrained quadratic 0–1 minimization problem. Discrete Applied Mathematics 157, pp. 1267–1290.

[35] Heitsch, H., Römisch, W. (2005) Scenario tree modelling for multistage stochastic programs. Preprint 296, DFG Research Center Matheon.

[36] Held, H., Woodruff, D. L. (2003) Heuristics for multi-stage interdiction of stochastic networks. Journal of Heuristics.

[37] Hochreiter, R., Pflug, G. (2007) Financial scenario generation for stochastic multi-stage decision processes as facility location problem. Annals of Operations Research, Volume 152, Number 1, pp. 257-272.

[38] Houming, F., Tong, Z., Xiaoyan, Z., Mingbao, J., Guosong, D. (2008) Research on emergency relief goods distribution after regional natural disaster occurring. International Conference on Information Management and Industrial Engineering.

[39] Jonsbraten, T. W., Wets, R. J. B., Woodruff, D. L. (1998) A class of stochastic programs with decision dependent random elements. Annals of Operations Research 82, pp. 83–106.

[40] Kall, P., Wallace, S. (1994) Stochastic Programming. Wiley.

[41] Kaut, M., Wallace, S. (2007) Evaluation of scenario-generation methods for stochastic programming. Pacific Journal of Optimization, 3 (2), pp. 257-271.

[42] Kaut, M., Wallace, S., Hoyland, K. (2003) A Heuristic for Moment-matching Scenario Generation. Computational Optimization and Applications, 24 (2-3), pp. 169-185.

[43] Liu, C., Fan, Y., Ordonez, F. (2006) A Two-Stage Stochastic Programming Model for Transportation Network Protection. Working Paper, Department of Civil and Environmental Engineering and Institute of Transportation Studies, University of California at Davis.

[44] Oral, M., Kettani, O. (1990) Equivalent formulations of nonlinear integer problems for efficient optimization. Management Science 36 (1), pp. 115–119.

[45] Oral, M., Kettani, O. (1992) A linearization procedure for quadratic and cubic mixed-integer problems. Operations Research 40 (Supp. 1), pp. 109–116.

[46] Parentela, E., Nambisan, S.S. (2000) Emergency response (disaster management). Urban Planning and Development Applications of GIS, 181–196.

[47] Plambeck EL, Fu B-R, Robinson SM, Suri R (1996) Sample-path optimization of convex stochastic performance functions. Math. Progr. 75: 137–176.

[48] Pflug, G. (1990) On-line optimization of simulated markovian processes. Mathematics of Operations Research 15, No.3, pp. 381-395.

[49] Poojari, C.A., Lucas, C. and Mitra, G.: "Robust solution and risk measures for a supply chain planning problem under uncertainty". Journal of the Operational Research Society 59, pp. 2–12, 2006.

[50] Römisch, W. (2009) Scenario generation in stochastic programming. Wiley Encyclopedia of Operations Research and Management Science.

[51] Rubinstein, B. Y. (1981) Simulation and the Monte Carlo Method. New York, Wiley & Sons

[52] Ruszczynski, A., Shapiro, A. (2003) Stochastic Programming. Handbooks in Operations Research and Management Science, Vol. 10, Elsevier.

[53] Santoso, T., Ahmed, S., Goetschalckx, M. and Shapiro, A.: "A stochastic programming approach for supply chain network design under uncertainty. European Journal of Operational Research 167, pp.96–115, 2005.

[54] Senay, S. (2007) Efficient Solution Procedures for Multistage Stochastic Formulations of Two Problem Classes. PhD thesis, Georgia Institute of Technology.

[55] Shapiro, A., Dentcheva, D., Ruszczynski, A. (2009) Lectures on Stochastic Programming, SIAM.

[56] Sherali, H.D., Desai, J., Glickman, T.S. (2004) Allocating emergency response resources to minimize risk with equity considerations. American Journal of Mathematical and Management Sciences 24 (3–4), 367–410.

[57] Sheu, J.-B. (2007) An emergency logistics distribution approach for quick response to urgent relief demand in disasters. Transportation Research Part E: Logistics and Transportation Review 43 (6), 687.

[58] Sohn, J. (2006) Evaluating the significance of highway network links under the flood damage: An accessibility approach. Transportation Research Part A: Policy and Practice, Volume 40, Issue 6, July 2006, pp. 491-506.

[59] Sohn, J. Kim, J., Hewings, G., Lee, J., Jong, S-G. (2003) Retrofit Priority of Transport Network Links under an Earthquake. Journal of Urban Planning and Development, Vol. 129, No. 4, December 2003, pp. 195-210.

[60] Soyster, A.L. (1973) Convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research 21, pp. 1154–1157.

[61] Talluri, K., van Ryzin, G. (2004) Revenue Management Under a General Discrete Choice Model of Consumer Behavior. Management Science 50, No. 1, pp. 15-33.

[62] Train, K. (2003) Discrete Choice Models with Simulation. Cambridge University Press.

[63] Tsai, Y., Wang, Z., Yang, C. (2002) A prototype real-time GPS/GIS-based emergency response system for locating and dispatching moving patrol vehicles with the beat-based shortest distance search. Traffic and Transportation Studies Proceedings of ICTTS 2, 1361–1368.

[64] Tomasini, R., Van Wassenhove, L. (2009) Humanitarian Logistics. Palgrave Macmillan.

[65] van der Vlerk, M. (1996-2007) “Stochastic Programming Bibliography” World Wide Web, http://mally.eco.rug.nl/spbib.html.

[66] Viswanath, K., Peeta, S., Salman, S. (2004) Investing in the links of a stochastic network to minimize expected shortest path length. Purdue University Economics Working Papers.

[67] Wallace, S. (1987) Investing in arcs in a network to maximize the expected max flow. Networks, Volume 17 Issue 1, pp. 87 – 103.

[68] Wallace, S.W. and Fleten, S.E. “Stochastic Programming Models in Energy”. Handbooks in OR & MS, Vol.10, Chapter 10, Edited by A.Ruszczynski and A.Shapiro, Elsevier Science, 2003.

[69] Wollmer, R. (1980) Investment in stochastic minimum cost generalized multicommodity networks with application to coal transport. Networks, Volume 10 Issue 4, pp. 351 – 362.

[70] Wollmer, R. (1991) Investments in stochastic maximum flow networks. Annals of Operations Research, Volume 31, Number 1, pp. 457-467.