Journal Name xx (2021) 1–24

A Biased-Randomized Algorithm for Optimizing Efficiency in Parametric Earthquake (Re)Insurance Solutions

Christopher Bayliss[a], Roberto Guidotti[b], Alejandro Estrada-Moreno[c,a], Guillermo Franco[b] and Angel A. Juan[a,*]

[a]*IN3 – Computer Science Dept., Universitat Oberta de Catalunya, Av. Carl Friedrich Gauss 5, Castelldefels, 08860, Spain*
[b]*Guy Carpenter & Company, LLC, 1166 Avenue of the Americas, New York, NY 10036, USA*
[c]*Dept. d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Av. Països Catalans 26, Tarragona, 43007, Spain*
E-mail: cbayliss@uoc.edu [C. Bayliss]; roberto.guidotti@guycarp.com [R. Guidotti]; alejandro.estrada@urv.cat [A. Estrada-Moreno]; guillermo.e.franco@guycarp.com [G. Franco]; ajuanp@uoc.edu [A.A. Juan]

Received DD MMMM YYYY; received in revised form DD MMMM YYYY; accepted DD MMMM YYYY

Abstract

Natural catastrophes with their widespread damage can overwhelm the financial systems of large communities. Catastrophe insurance is a well-understood financial risk transfer mechanism, aiming to provide resilience in the face of adversity. However, catastrophe insurance has generally a low penetration, mainly due to its high cost or to distrust of the product in providing a fast financial recovery. Parametric insurance is a form of derivative insurance that pays quickly and transparently based on a few measurable features of the event, offering a promising avenue to increase catastrophe insurance coverage. In the context of seismic risk, parametric policies may use location and magnitude of an earthquake to determine whether a payment should be made. In this paper we follow a design typology referred to as 'cat-in-a-box', where magnitude thresholds are defined over a set of cuboids that partition Earth's crust. The main challenge in the design of these tools consists in finding the optimal magnitude thresholds for a large set of cubes that maximize efficiency for the insured, subjected to a budgetary constraint. Additional geometric constraints aim to reduce the volatility of payments under uncertainty. The parametric design problem is a combinatorial problem, which is *NP-hard* and large scale. In this paper we propose a fast heuristic and a biased-randomized algorithm to solve large-sized problems in reasonably low computing times. Experimental results illustrate the computational limits and solution quality associated with the proposed approaches.

Keywords: Biased Randomization; Combinatorial Optimization; Parametric Insurance; Risk Analysis

1. Introduction

Large-scale seismic events occurred in recent years illustrate the important role of insurance as a mitigation strategy (Franco, 2014). However, when payments are urgently required to prompt the recovery

*Author to whom all correspondence should be addressed (e-mail: ajuanp@uoc.edu).

process, traditional policies may become cumbersome and insufficient. The large number of simultaneous claims and the associated damage inspections typically require a long time to process. Complexities and disagreements in policy conditions can set in motion lengthy disputes that ultimately act as deterrents to a quick and orderly recovery. Such was the case in New Zealand, where liquefaction damages experienced during the 2010-2011 seismic events prompted numerous disagreements as to whether the losses should be covered by the traditional insurance policies (King et al., 2014). These disputes ultimately prolonged the claims process over a span of several years, and ended up hurting the individuals that insurance was designed to protect.

Since the 1990s, the (re)insurance industry has explored the development of parametric risk transfer mechanisms to respond fast and transparently after a catastrophic event. These solutions typically rely on a series of parameters that describe the event's potential to cause loss in order to quickly ascertain whether a payment is justified. Often, because of their simplicity, transparency, and potential for automation and customization, they are hailed as the most promising avenue to address the challenge posed by the *protection gap*, which refers to the 70% proportion of economic losses caused by natural disasters worldwide currently not covered by any formal financial mechanism (Guy Carpenter, 2014, 2017). Although in continuous expansion, the diffusion of parametric hedges is occurring rather slowly, as regulators and consumers come to grips with the basic limitation of these tools, i.e., their *basis risk*. As discussed in Section 2, this concept denotes the lack of perfect correlation between the parameters used to trigger a payment and the loss suffered. For instance, the parametric tool may trigger a significant payment for an event that caused limited damage; or it may trigger a small payment despite the occurrence of a seismic event that causes extensive loss. These two instances are referred to as *positive* and *negative* basis risk, respectively. Hence, the parametric trigger conditions need to be designed carefully, in a way such that payments correlate well with loss-causing events.

1.1. Related Work

Parametric earthquake insurance solutions can be designed in multiple ways, considering actual measurements or the results of a model tuned to certain event characteristics. Wald and Franco (2016, 2017) provide an overview on the nomenclature and classification of these solutions, including –among others– 'pure parametrics' or *cat-in-a-box* solutions and indices. Cat-in-a-box triggers are a class of parametric earthquake insurance solutions that rely on: *(i)* the definitions of 'boxes' to classify the location of the event; and *(ii)* an event-level energy-release metric (e.g., the magnitude of the earthquake). The boxes are geographically defined by minimum and maximum values of longitude and latitude. If the triggers employ a discretization in depth, as in this paper, the boxes are also defined by minimum and maximum values of depth, representing cuboids –or cubes if regular. A threshold magnitude is assigned to each cuboid, so that if an earthquake occurs within a cuboid (i.e., its hypocenter is located inside a cuboid of the considered domain), and the magnitude of the earthquake attains or exceeds the magnitude threshold assigned to that cuboid, then a fixed, predefined payment takes place, which had been agreed by all parties in advance. Information about location of the hypocenter and the magnitude of earthquakes is publicly available in near-real-time from agencies, such as the U.S. Geological Survey (https://www.usgs.gov/). Intensity-based indices, sometimes referred to as *second generation parametric indices*, are another important class of parametric earthquake insurance solutions that rely on the

shaking intensity at given locations (stations), rather than on the event magnitude. These have been described, and often compared to cat-in-a-box solutions, in several works (Goda, 2013, 2015; Pucciano et al., 2017). While both are valid approaches depending on the context, requirements, and the geographic setting (Goda et al., 2019), this paper focuses on the cat-in-a-box mechanism to benefit from its greater simplicity and transparency.

Cat-in-a-box solutions have been employed numerous times in the financial markets since 1996 (Franco, 2014), most notably by Mexico since the first issuance of their cat bond series in 2006 (Cardenas et al., 2007). Several studies in the past years have attempted to formalize the parametric design process. A formal optimization framework was presented by Franco (2010) to automate their design, minimizing their basis risk. That study constrained the design to a handful of zones aligned with the market demand for simple and intuitive solutions. As the financial markets progressively became more comfortable with increased complexity, the possibility of considering hundreds or even thousands of boxes became palatable (Franco, 2013). However, it was not until 2015 that the first so-called 'cat-in-a-grid' transaction hit the market in the shape of the Acorn Re Ltd. cat bond (Artemis, 2015), a parametric solution that protected the Kaiser-Permanente corporation from southern Canada to northern Mexico along the western US coast, using hundreds of boxes. Since then, this approach has been used in several large transactions, including the Pacific Alliance cat bond, a USD 1.4 billion parametric earthquake risk transfer instrument that provides cover to Mexico, Colombia, Peru, and Chile (Artemis, 2018). Additional studies have showcased the potential of this approach to be used at larger scales in California in an automated framework (Franco et al., 2018).

1.2. Paper Contributions and Structure

In this paper we focus on the parametric trigger design, introducing the concept of *efficiency*, which we define as the ratio of the average annual loss (AAL) of those events triggering a payment over the AAL of all events, as detailed in Section 2. We seek to maximize the parametric trigger efficiency subjected to a series of constraints, related to the insured's budget and to the solution's performance. We investigate the computational performance and limitations associated with the production of these solutions. The study of the computational aspects is critical, as massively deployed products will require the construction of an increasing amount of policies –potentially customized to a growing group of insureds– with accuracy and speed.

The main contributions of this paper are the development of: *(i)* a mathematical model to accurately describe the optimization problem introduced above; and *(ii)* a fast heuristic and a biased-randomized algorithm –based on the previous heuristic– to tackle more efficiently large-sized instances of the problem. As described in Grasas et al. (2017), biased-randomized techniques are based on the introduction of an oriented (non-uniform) randomization process inside the constructive stage of a given heuristic. By doing so, a deterministic heuristic is transformed into a randomized algorithm that can be run multiple times (either in sequential or in parallel) without losing the logic behind the heuristic. Biased randomization algorithms have been successfully applied in many different optimization problems, including: logistics (De Armas et al., 2017; Quintero-Araujo et al., 2017; Estrada-Moreno et al., 2019a), production (Ferrer et al., 2016; Gonzalez-Neira et al., 2017), transportation (Belloso et al., 2019; Dominguez et al., 2016; Estrada-Moreno et al., 2019b,c; Almouhanna et al., 2020), etc. These techniques can also be used

to enhance existing metaheuristic frameworks (Ferone et al., 2019).

The rest of the paper is structured as follows: Section 2 introduces the conceptual aspects and formulates the governing mathematical problem; Section 3 presents a heuristic method capable of solving large-sized instances in short computing times, and extends it into a novel multi-start algorithm by employing biased-randomization techniques; Section 4 presents the numerical experiments completed to test the heuristic-based solving approaches, applied –as an example– to Greece, among the countries with the highest seismic risk worldwide; finally, Section 5 highlights the main conclusions.

2. Problem Formulation

2.1. Efficiency as a Performance Metric

One of the common shortcomings in parametric insurance solutions available on the market is the difficulty in assessing their performance. Whereas the goal of a perfect correlation between experienced losses and parametric insurance recovery remains the overarching objective of any parametric insurance solution, the actual basis risk is extremely challenging to quantify and, therefore, to minimize (Franco, 2010, 2013; Ross and Williams, 2012; Figueiredo et al., 2018). In the context of parametric insurance solutions, basis risk is defined as the lack of sufficient correlation between payoff and experienced losses (Cummins et al., 2004; Cummins, 2008). As described in Franco (2010), sources of basis risk can be attributed to the inaccuracy of the underlying catastrophe model (i.e. model risk), to the intrinsic limitation of a binary or step-wise payment scheme, or to the design of the trigger mechanism. The focus of this paper is to tackle and minimize the last source of basis risk, associated with the design of the trigger mechanism. The proposed parametric solution, described in the following section, belongs to the family of binary classification problems (Jolliffe and Stephenson, 2012). While in this paper we propose a particular metric to approximately denote the basis risk of the solution attributable to the design process, the identification of proper metrics to quantify basis risk for different problems continues to be a difficult challenge.

To quantify basis risk, we introduce in this paper the concept of *efficiency*. As mentioned above, this is the ratio between the average annual loss (AAL) associated with events that would trigger the transaction over the AAL associated with all events. The AAL for a set of events, in turn, is defined as the sum product of the event losses and the events' annual rates of occurrence. A parametric solution with efficiency equal to 1 would trigger a payout for all stochastic events causing any loss. This would result in a very high frequency of payment, hence in an extremely expensive insurance policy. On the other end, a parametric solution with a low efficiency may result in a cheaper policy, but the low frequency of payout may not meet the insured's protection needs. Note that a high efficiency does not necessarily mean a high frequency of payout, i.e.: efficiency and frequency of payouts are not necessarily correlated. In fact, a solution that triggers frequently could be inefficient if those frequent payments are triggered for events that do not cause loss. This would mean the insured is paying a significantly high premium to cover events for which the cover is unnecessary. Therefore, it is possible to quantitatively compare and assess the performance of different parametric insurance solutions by comparing their efficiency at a common frequency of payment. Throughout the paper we often refer to AAL as 'risk', as both these measures aim to describe the expected loss of the insured's portfolio. Frequency is typically referenced

throughout the paper in terms of *return period* of the solution, which is the inverse of its annual rate of triggering a payment and is usually expressed in years.

Working with earthquakes is challenging, since the frequency of occurrence of damaging events is low and the historical record is insufficient to derive any meaningful statistics. This work, as is customary in similar studies, relies on the usage of a catastrophe risk model, a numerical simulation platform able to generate stochastic events in harmony with the seismic setting of interest. These models, discussed in the literature by Grossi et al. (2005) and, more recently, by Mitchell-Wallace et al. (2017), provide all the data necessary, such as event characteristics and losses that are needed to apply the framework for the optimization study presented here.

2.2. Parametric Insurance Design as an Optimization Problem

As in the previous study by Franco (2010), we will tackle the problem of parametric trigger design as an optimization problem. In contrast with the aforementioned work, we consider a much larger solution domain by increasing the number of boxes and magnitude levels considered, as well as by changing the main optimization objective. Instead of using the trigger outcomes as the basis for minimizing basis risk, we use the metric of efficiency introduced above. This metric serves as the objective function of an optimization problem that aims to provide maximum coverage to the insured for a given level of premium (which is a function of the annual frequency of trigger).

The domain of interest is divided into a grid of non overlapping cuboids. The parametric design associates a magnitude threshold to each cuboid, such that if an event happens in the region of interest and its magnitude attains or exceeds the magnitude threshold of the cuboids containing its hypocenter (nucleation or center of energy dispersion), the parametric mechanism produces an agreed payment. The underlying optimization problem consists in selecting the threshold magnitudes for each cube, such that the efficiency is maximized under a budget constraint (based on the total annual trigger rate). A fine resolution grid typically allows the design to achieve higher values of efficiency as the threshold magnitudes can vary more precisely in response to local changes in seismicity. However, this also increases the size of the optimization problem, which severely limits the use of exact optimization methods –as it will be discussed later in the paper.

2.3. A Binary Integer Programming Model to Solve the Optimization Problem

This section formulates the parametric hedge design problem previously conceptualized as a binary integer program. Let C denote the set the cubes, and $\bar{M}$ the set of J threshold magnitude levels. We shall assume that the elements of $\bar{M}$, which represent moment magnitude levels, are indexed with the symbol j and are ordered from least ($j = 1$) to greatest ($j = J$). For each cube $i \in C$ a threshold magnitude is selected. Let x_{ij} denote a binary decision variable, which takes the value 1 if the threshold earthquake magnitude of cube i is j, and takes the value 0 otherwise. Let t_{ij} denote the total risk associated with magnitude threshold j in cube i. In particular t_{ij} is the sum of the risk associated with earthquakes with magnitude greater than or equal to j in cube i. The objective of maximizing efficiency can be stated mathematically as follows:

$$\max \frac{\sum_{i \in C} \sum_{j \in \bar{M}} t_{ij} x_{ij}}{\sum_{i \in C} t_{i1}} \tag{1}$$

Since the denominator in Equation (1) is a constant value (the AAL for all events, starting with the lowest magnitude index $j = 1$), the maximization problem is equivalent to just maximizing the numerator. The annual trigger rate (i.e., the expected frequency of payment) cannot exceed r_{max}. Recall this constraint serves the purpose of limiting the premium budget for the transaction. This constraint is expressed as:

$$\sum_{i \in C} \sum_{j \in \bar{M}} s_{ij} x_{ij} \leq r_{max}, \tag{2}$$

where s_{ij} is the total rate with which earthquakes with a magnitude greater than or equal to j occurs in cube i. To account for the requirement that only a single threshold magnitude can be selected for each cube, the binary integer program includes the following constraint:

$$\sum_{j \in \bar{M}} x_{ij} = 1, \ \forall i \in C, \tag{3}$$

To account for the possibility that the input data may be inaccurate, noisy or incomplete, a number of additional geometric constraints are added to the binary integer program defined by Equations (1) to (3).

2.3.1. Depth constraint

This consistency constraint requires that threshold magnitudes are non-decreasing in depth. Earthquakes of equal magnitude are, in general, expected to have higher losses the shallower they occur (closer to the surface). While there are specific geological conditions under which a deeper earthquake may cause more damage than a shallower event of equal magnitude, this can be considered as an anomaly. Therefore, with the aim of constructing intuitive parametric designs, we incorporate this constraint expressed as:

$$\sum_{j \in \bar{M}} j x_{ij} \leq \sum_{j \in \bar{M}} j x_{(i+\Delta^x \Delta^y) j}, \ \forall i \in \left\{1, 2, \ldots, \left(\Delta^z - 1\right)\left(\Delta^x \Delta^y\right)\right\}, \tag{4}$$

where Δ^x is the number of the longitude layers of cubes, Δ^y is the number of the latitude layers of cubes, and Δ^z is the number of the depth layers of cubes. Therefore $i + \Delta^x \Delta^y$ is the index of the cube below cube i, since there are $\Delta^x \Delta^y$ in each depth layer.

2.3.2. Smoothness constraints

Smoothness constraints have a dual purpose. First they address the issue of potential incompleteness of the simulation models. If a certain cell in the design lacks information to determine its appropriate magnitude threshold, its value can be determined based on the values in the neighborhood of said cell.

Secondly, they provide a device to limit the variability of the magnitude thresholds across the domain. This helps control the volatility of payments that can appear due to numerical scatter in the simulation. For instance, it is possible that a particular model would indicate high losses in a cell that lies next to another cell with negligible loss. The smoothing constraints help limit the variability between neighboring cell thresholds in order to provide a less volatile payment behavior. Two components of smoothness constraints are introduced, one focused on limiting the maximum slope of magnitude thresholds, and a second focused on limiting the maximum curvature. In contrast to the depth constraint (4), the slope and curvature constraints are applied solely in the longitude-latitude plane, within each depth layer of cubes. For each pair of directly adjacent and diagonally adjacent cubes, i and j, the slope δ_{ij} is calculated as follows.

$$\delta_{ij} = \frac{\hat{M}_j - \hat{M}_i}{d} = \alpha_{ij}\left(\hat{M}_j - \hat{M}_i\right), \tag{5}$$

where $\hat{M}_i$ is the selected threshold magnitude for cube i, the value of which is $\sum_{m \in \bar{M}} m x_{im}$. Note that Equation (5) is a linear function of our binary decision variables, which means that such slope constraints can be added directly into a linear binary integer program. Similarly, d is the distance in degrees between cubes i and j, while $\alpha_{ij} = \frac{1}{d}$. Note that d depends on the direction of the slope constraint. In particular, we have d^x, d^y, and d^{xy} for slope constraints in the x direction, y direction, and xy direction. For each triple of directly adjacent and diagonally adjacent cubes i, j, and k, the curvature χ_{ijk} is calculated as follows.

$$\chi_{ijk} = \frac{\left(\frac{\hat{M}_j - \hat{M}_i}{d}\right) - \left(\frac{\hat{M}_k - \hat{M}_j}{d}\right)}{d + d} = \beta_{ijk}\left(2\hat{M}_j - \hat{M}_i - \hat{M}_k\right), \tag{6}$$

where $\beta_{ijk} = \frac{1}{2d^2}$. Note also that Equation (6) is a linear function of our binary decision variables, which means that such curvature constraints can be added to a linear binary integer program. If the absolute maximum allowable slope is denoted by S, our slope constraints are:

$$-S \leq \delta_{ij} \leq S, \ \forall\,(i,j) \in A, \tag{7}$$

where A is the set of pairs of adjacent cubes. The number of slope constraints ($|A|$) is:

$$|A| = \Delta^z\left(4\Delta^x\Delta^y - 3\Delta^x - 3\Delta^y + 2\right). \tag{8}$$

If the absolute maximum allowable curvature is C, our curvature constraints are:

$$-C \leq \chi_{ijk} \leq C, \ \forall\,(i,j,k) \in B, \tag{9}$$

where B is the set of triples of adjacent cubes. The number of curvature constraints ($|B|$) is

$$|B| = 2\Delta^z\left(2\Delta^x\Delta^y - 3\Delta^x - 3\Delta^y + 4\right). \tag{10}$$

We now consider a number of observations regarding these slope and curvature constraints. If $S < \frac{\Delta m}{d}$, where Δm is the size of a magnitude interval, then the slope constraints will have the effect of forcing

a constant threshold magnitude in each depth layer, even though $S > 0$. This is because the minimum non-zero slope between neighboring cubes, corresponding to a difference of one magnitude interval, exceeds S. Besides setting Δm to a sufficiently small value, or setting d to a sufficiently large value, the slope constraints can be applied to non-neighboring cubes. This approach effectively increases d. In terms of a positive integer number of cubes (n^*) between either end of a slope constraint, the minimum span of a slope constraint that is required –to avoid flat threshold maps in each depth layer– satisfies the following:

$$n^* = \min_{n \in \mathbb{N}^+} \left\{ n \left| \frac{\Delta m}{nd} \leq S \right. \right\}. \tag{11}$$

As a result, α_{ij} becomes $\frac{1}{n^* d}, \forall\,(i,j) \in A$. A similar effect can occur if $C < \frac{\Delta m}{d^2}$, which corresponds to the curvature constraints forcing constant threshold magnitudes in each depth layer. This situation can be avoided by increasing the distance between the three cubes within a single curvature constraint. In terms of a positive integer (n') between each cube of a curvature constraint, the minimum distance between neighboring cubes in a curvature constraint that is required –to avoid flat threshold maps in each depth layer– satisfies the following:

$$n' = \min_{n \in \mathbb{N}^+} \left\{ n \left| \frac{\Delta m}{(nd)^2} \leq C \right. \right\}. \tag{12}$$

As a result, β_{ijk} becomes $\frac{1}{2(n'd)^2}, \forall\,(i,j,k) \in B$. Whilst this approach allows the designer to select lower values for S and C, it should be noted that if either of n^* or n' are greater than one then the slope and curvature constraints in each of the directions x, y, and $x - y$ will be applied separately to different subsets of cubes, in a checkers board like arrangement. Therefore for cases where n^* or n' are greater than one, a careful tuning of the values of S and C may be necessary in order to avoid solutions which appear to have a periodic character.

Additionally, slope constraints may also constrain curvature. If the absolute maximum slope is S and no curvature constraints are applied, then the curvature of adjacent cubes i, j, k is maximum when the slopes of adjacent cube pairs i, j and j, k are $\delta_{ij} = S$ and $\delta_{jk} = -S$. Noting that the curvature is calculated as $\chi_{ijk} = \beta_{ijk}\left(\left(\hat{M}_j - \hat{M}_i\right) - \left(\hat{M}_k - \hat{M}_j\right)\right)$, and that for the case of maximum curvature under slope constraints we have $\frac{\delta_{ij}}{\alpha_{ij}} = \frac{S}{\alpha_{ij}} = \hat{M}_j - \hat{M}_i$ and $\frac{\delta_{jk}}{\alpha_{jk}} = -\frac{S}{\alpha_{jk}} = \hat{M}_k - \hat{M}_j$, which implies that the maximum curvature under slope constraints is

$$\chi^S_{max} = \beta_{ijk}\left(\left(\frac{S}{\alpha_{ij}}\right) - \left(-\frac{S}{\alpha_{jk}}\right)\right) = \beta_{ijk} S\left(\frac{1}{\alpha_{ij}} + \frac{1}{\alpha_{jk}}\right) = \frac{S}{d}.$$

Therefore, if the desired maximum curvature C is less than $\frac{S}{d}$, then the slope constraints alone are sufficient for enforcing the desired curvature constraints. Smaller cubes, however, reduce the chance that the slope constraints alone sufficiently constrain the curvature.

2.3.3. Finishing touch heuristic

As a result of incomplete simulation models, some cubes may contain no earthquake event simulations. Therefore, the threshold magnitudes of such cubes are constrained only by the non-decreasing threshold in depth, slope, and curvature constraints, but not by the total trigger rate constraint. However, since earthquakes can technically occur in cubes with no simulation data, the final step in the design consists of increasing the thresholds of the cubes with no simulation data, as well as cubes that do not transfer any risk as high as possible without violating any of the geometric constraints or decreasing the total trigger rate. This is a more honest approach than simply leaving thresholds in non-modeled cubes with low values, as the market would not have the ability to price this risk in the absence of simulation data.

3. Proposed Methods for Solving the Optimization Problem

3.1. Calculating data prerequisites

The formulation presented requires as input the AAL or risk metric t_{ij}, the annual rate of occurrence r_{ij}, and the cumulative annual rate of occurrence s_{ij} associated with earthquakes with a magnitude greater than or equal to j in each cube i. As mentioned above, this work assumes these parameters are calculated with the help of an earthquake risk model. Given a set of data E capturing a representative sample of simulated earthquake events in the target region, complete with hypocenter locations, magnitudes, associated financial losses and annual rates of occurrence, the following Algorithm 1 is used to compute the required values for t_{ij}, r_{ij}, and s_{ij}, $\forall i \in C$, $\forall j \in \bar{M}$.

Algorithm 1 Algorithm for calculating required parameters from simulated earthquake events.

Inputs: E
Set $t_{ij} = r_{ij} = s_{ij} = 0$, $\forall i \in C$, $\forall j \in \bar{M}$
for $e \in E$ **do**
  $i \leftarrow Cube\,(e)$
  $j \leftarrow magnitudeInterval\,(e)$
  $rate \leftarrow Rate\,(e)$
  $loss \leftarrow Loss\,(e)$
  $t_{ij} \leftarrow t_{ij} + (rate \times loss)$
  $r_{ij} \leftarrow r_{ij} + rate$
  $l_{ij} \leftarrow l_{ij} + loss$
  for $k \in \bar{M} | k \leq j$ **do**
    $s_{ik} \leftarrow s_{ik} + rate$
  end for
end for
Output: t_{ij}, r_{ij}, s_{ij}, $\forall i \in C$, $\forall j \in \bar{M}$

3.2. A Fast Heuristic (Greedy Algorithm)

This section presents a fast and simple (greedy) heuristic algorithm to solve the optimization problem (Algorithm 2). The heuristic algorithm presented in this section is based on initially setting all of the threshold magnitudes of all cubes to their maximum levels, which is the zero efficiency solution and which also complies with all the constraints (this solution has zero trigger rate, non-decreasing thresholds in depth, and zero slopes and curvatures). For the convenience of providing a formal presentation of the heuristic, we introduce the notation y_i to denote the index of the current selected magnitude threshold of cube i. The solution representations for the binary integer programming formulation and the heuristic are related according to $y_i = k$ if k is the index of j in $\bar{M}$ and $x_{ij} = 1$. Using this convention we introduce the functions $maxSlope\,(i, y_i)$ and $maxCurvature\,(i, y_i)$, which return the maximum slope and curvature over the cubes neighboring cube i given a new candidate threshold magnitude y_i for cube i. These functions will be used for checking whether the slope and curvature constraints are violated.

The main part of the heuristic Algorithm 2 proceeds by sequentially selecting cube thresholds to lower by one magnitude level, such that the increase in efficiency divided by the corresponding increase in the trigger rate is maximized. In particular, the cube whose threshold will be lowered by one level is that for which $\frac{l_{i,(y_i-1)}}{r_{i,(y_i-1)}}$ is maximized while, at the same time, no constraints are violated. In this way, the algorithm increases efficiency at the fastest rate possible with respect to the corresponding increase in trigger rate. Since the proposed algorithm starts from a feasible solution and permits no feasibility violating moves thereafter, the feasibility of the final solution is guaranteed. We collect all of the constraint checks for a change of threshold magnitude of h levels of cube i into a single function, which returns false if the new candidate magnitude for cube i is not feasible, true otherwise (Algorithm 3). In Algorithm 3, line 1 checks the five constraints: *(i)* cube i's new threshold is between 1 and the maximum threshold $|\bar{M}|$; *(ii)* the new total trigger rate will not exceed r_{max}; *(iii)* the new threshold value will not violate the non-decreasing thresholds in depth constraint; *(iv)* the maximum slope constraint will not be violated; and *(v)* the maximum curvature constraint will not be violated. Note also that the function is general, and can be used to check the feasibility of arbitrary increases –or decreases– of threshold magnitude of any single cube.

The proposed heuristic Algorithm 2 requires an additional step following the initialization task of setting the threshold magnitudes of all cubes to their maximum levels (lines 4 and 5 of Algorithm 2). A fine discretization of cubes ($i \in C$) and threshold magnitude levels may lead to a case in which there is no data for some threshold magnitudes ($j \in \bar{M}$). In those cases, it is necessary for the heuristic to skip over such threshold magnitudes, since those magnitudes thresholds offer no efficiency benefit and skipping over them avoids unnecessary iterations within the heuristic algorithm (lines 6 and 7 of Algorithm 2). This task is accomplished by a function denoted $skip\,(y, p)$ (Algorithm 4), where $p = 1$ corresponds to the case of raising thresholds, while $p = -1$, corresponds to a lowering of the magnitude thresholds. The $skip\,(y, p)$ function repeatedly cycles through the set of cubes C and tries to feasibly lower (or raise) cube thresholds without changing the total efficiency or total trigger rate of the current solution y. It terminates when no further moves have been made after completing an entire cycle, as shown in Algorithm 4.

In Algorithm 2, line 11 describes mathematically how a cube threshold is selected in each iteration to be lowered by one level (y_i). The cube chosen is that for which $\frac{l_{i,(y_i-1)}}{r_{i,(y_i-1)}}$ is maximized, such that no

constraints are violated. The heuristic then attempts to skip over magnitude thresholds of cubes which do not transfer risk or do not increase the trigger rate. If on line 12 of Algorithm 2 no cube thresholds can be lowered without violating any constraints, then the algorithm terminates (line 20) and the final solution is returned (line 25).

Algorithm 2 Greedy heuristic for maximizing total expected efficiency

1: Inputs: $l_{ij}, r_{ij},\ \forall i \in C,\ \forall j \in \bar{M}$
2: Set $rateTotal = 0$
3: Set $riskTotal = 0$
4: Step 1: Set the thresholds for all cubes to the maximum magnitudes
5: $y_i \leftarrow |\bar{M}|,\ \forall i \in C$
6: Step 2: Lower the thresholds of all cubes as much as possible without increasing $rateTotal$ or $riskTotal$ and without violating any constraints (Section 2.3.3).
7: $y \leftarrow skip\,(y, -1)$
8: Step 3: While the total trigger rate constraint is not violated sequentially identify the single threshold to decrease which maximizes the increase in the total efficiency divided by the associated increase in the trigger rate
9: $finished = false$
10: **while** $notfinished$ **do**
11: $\quad i^* = \underset{i \in C}{\arg\max} \left\{ \frac{l_{i,(y_i-1)}}{r_{i,(y_i-1)}} \,|moveFeasible\,(i, -1, y, rateTotal) \right\}$
12: $\quad$ **if** $i^* \neq \emptyset$ **then**
13: $\quad\quad$ Update the threshold
14: $\quad\quad y_{i^*} \leftarrow y_{i^*} - 1$
15: $\quad\quad rateTotal \leftarrow rateTotal + r_{i,y_{i^*}}$
16: $\quad\quad riskTotal \leftarrow riskTotal + r_{i,y_{i^*}} l_{i,y_{i^*}}$
17: $\quad\quad$ Skip over thresholds feasibly and without changing the overall efficiency
18: $\quad\quad y \leftarrow skip\,(y, -1)$
19: $\quad$ **else**
20: $\quad\quad finished = true$
21: $\quad\quad$ Apply the finishing touch heuristic (Section 2.3.3)
22: $\quad\quad y \leftarrow skip\,(y, 1)$
23: $\quad$ **end if**
24: **end while**
25: return $(riskTotal, y)$

3.3. From Greedy to Biased Randomized Algorithms

3.3.1. A Biased Randomized Algorithm

In this section we convert the greedy algorithm presented in Section 3.2 into a biased randomized (BR) algorithm. BR algorithms are based on incorporating a degree of randomness into greedy heuristics

Algorithm 3 $moveFeasible\,(i, h, y, rateTotal)$

if $\left(\begin{array}{ccccc} 1 & \leq & y_i + h & \leq & |\bar{M}| \\ 0 & \leq & rateTotal + sign\,(h) \sum_{k=y_i}^{y_i+h} r_{i,k} & \leq & r_{max} \\ & & \bar{M}_{y_i+h} & \leq & \bar{M}_{y_i+\Delta^x \Delta^y} \\ -S & \leq & maxSlope\,(i, y_i + h) & \leq & S \\ -C & \leq & maxCurvature\,(i, y_i + h) & \leq & C \end{array} \right)$ **then**
    return true
else
    return false
end if

Algorithm 4 $skip\,(y, p)$

$finished = false$
while $not finished$ **do**
    $thresholdChanged = false$
    for $i \in C$ **do**
        while $\left(r_{i,y_{i+p}} = 0\right) \wedge moveFeasible\,(i, p, y, rateTotal)$ **do**
            $y_i \leftarrow y_i + p$
            $thresholdChanged = true$
        end while
    end for
    if $thresholdChanged = false$ **then**
        $finished = true$
    end if
end while
return y

whilst retaining the original logic of the greedy algorithm. In particular, whereas greedy algorithms build solutions by sequentially selecting the decisions which increase their objective value at the fastest rate, a BR algorithm applies a skewed probability distribution to the task of selecting the next solution component. In this way, it is possible to explore different configurations of the parametric insurance solution, possibly resulting in higher efficiency values. In this paper, a geometric distribution is used to select the next 'building' components to add to an emerging solution. At each stage of a BR algorithm, candidate decisions are sorted in decreasing order of the original greedy decision criteria. For the case of a geometric distribution, the following equation is used to select a solution component from the sorted candidate list.

$$index = Mod\left(|L|, \left\lfloor \frac{\ln\left(uniRand\,(0,1)\right)}{\ln\left(1-\beta\right)} \right\rfloor\right), \tag{13}$$

In Equation (13), $index$ is the position of the selected decision component in the candidate list (L), $uniRand$ is a uniformly distributed random input in the interval $[0, 1]$, and β is an input parameter which controls the balance between greediness and randomness. $\beta = 1$ simulates the original greedy algorithm, while $0 < \beta < 1$ simulates trade offs between greedy and random selection. Algorithm 2 becomes a BR algorithm if line 11 of that algorithm is replaced with $L \leftarrow sort\left(i \in C, \frac{l_{i,(y_i-1)}}{r_{i,(y_i-1)}} | moveFeasible\left(i, -1, y, rateTotal\right), descending\right)$ followed by an inserted line: $i^* \leftarrow L_{index}$, where $index$ is determined by Equation (13).

3.3.2. A Biased Randomized Algorithm with Partial Solution Learning mechanism

In this section we extend the BR algorithm that was presented in Section 3.3.1. This extension is based on using memory to store strong intermediate/partial solutions from each iteration of the biased randomization process, and then randomly selecting these as initial solutions in subsequent iterations of this process. In this way a 'learning' mechanism is incorporated into the BR algorithm. The logic behind this approach is two-fold. Firstly, applying biased randomization repeatedly starting from strong partial solutions increases the chance that better *final* solutions could be built from them. Secondly, applying biased randomization repeatedly starting from strong partial solutions increases the chance that even stronger *partial* solutions are found and therefore available to use as initial solutions in subsequent iterations.

A partial solution x' is defined as a set of trigger thresholds for all cubes, such that there exists at least one cube whose threshold can be lowered without violating the total trigger rate constraint (i.e., $r' = \sum_{i \in C} \sum_{j \in \bar{M}} x'_{ij} s_{ij} < R_{max}$). The process of storing partial solutions during an iteration of the biased randomization process can be described as follows: During an iteration of this process, a sequence of partial solutions are generated, each different from the next in the sequence by a single cube threshold. Every time a new partial solution is generated, a look-up-table of partial solutions, discretized according to the total trigger rate of partial solutions, is updated. If the new partial solution has a higher efficiency than the current best partial solution for the same total trigger rate interval, then the new partial solution is deemed the stronger of the two, and replaces the current partial solution. In particular, the new algorithm, referred to as *biased randomization with learning* (BRWL) maintains two look-up-tables, denoted Q_1 and Q_2, which differ in the criterion used to determine which of two partial solutions in the same trigger rate interval is the stronger one. Q_1 corresponds to the look-up-table update procedure described above, whereas Q_2 uses the efficiency of the final solution that was generated from a given partial solution as the criterion for replacing the partial solutions in that look-up-table.

Let $Q_j\left(v\left(r'\right)\right)$ denote the current partial solution for look up table j in trigger rate interval $v\left(r'\right)$, where $v\left(r'\right)$ is a function which returns the trigger rate interval of a partial solution with a trigger rate of r'. In each iteration of BRWL, a partial solution is randomly selected from the look up table and used as the initial solution for the BR algorithm. BRWL then records the complete sequence of partial solutions generated by a run of BR and their corresponding efficiency. Let U denote the sequence of partial solutions generated by one run of BR where u_k is the k^{th} partial solution in the sequence, such that u_0 is the initial solution and $u_{|U|}$ is the final solution. Let $z\left(u_k\right)$ denote the efficiency of the k^{th} partial solution in the sequence. Let $f\left(u_k\right)$ denote the efficiency of the final solution generated by BR starting from the solution u_k, and let $r\left(u_k\right)$ denote the trigger rate of the k^{th} partial solution. After each run of BR, the look up table is updated by cycling through U and replacing the existing entries with partial solutions with higher partial efficiency (Q_1) or total final efficiency (Q_2). This look up table

update procedure is outlined in Algorithm 5.

Algorithm 5 Look up table update procedure for BRWL

Inputs: U
for $k \in \{1, 2, \ldots, |U|\}$ **do**
  //Update the partial objective value layer (1) of Q
  if $z(u_k) > z(Q_1(v(u_k)))$ **then**
    //Replace the existing entry of the look up table
    $Q_1(v(u_k)) \leftarrow u_k$
  end if
  //Update the overall objective value layer (2) of Q
  if $f(u_k) > f(Q_2(v(u_k)))$ **then**
    //Replace the existing entry of the look up table
    $Q_2(v(u_k)) \leftarrow u_k$
  end if
end for
Output: updated Q

4. Computational Results

In this section, the heuristic methods (greedy, biased randomization, and biased randomization with learning) presented in Section 3 are compared with an exact approach (binary integer programming formulation). The computational results presented in this section are based on earthquake risk data regarding the region including and surrounding Greece and extend the preliminary results presented in Franco et al. (2019). The prototype presented in this paper relies on the AIR Worldwide European Earthquake Model (Catrader v19.1), partially described in Rong et al. (2011) and Lai et al. (2012). The results are divided into two parts. Firstly, we explore the limits of the exact approach (binary integer programming), considering the maximum problem size that can be solved using commercial solvers. Secondly, and in order to explore their performance, we compare the exact approach with the proposed heuristic approaches in a range of cube discretizations and maximum slope and curvature values.

4.1. Exact Approach

The binary integer programming formulation of the optimization problem is solved using commercial integer programming solvers for a range of cube sizes, threshold magnitude intervals, and different combinations of geometric constraints. This experiment is designed to explore the limits of the maximum problem size that can be solved using commercial solvers.

The following experimental results are based on the seismic risk in a volume of the Earth's crust between 15 and 30 decimal degrees of longitude, between 32 and 45 decimal degrees of latitude, and between 0 and 100 kilometers in depth. This volume includes the seismically active region of Greece. The minimum problem size is defined by dividing this region into 30 longitude layers, 26 latitude layers,

and 2 depth layers with 5 threshold magnitude intervals. This makes a total of 7,800 binary decision variables (the product of the number of layers in each dimension and the number of magnitude intervals). The target maximum problem size is defined by dividing the same region into 150 longitude layers, 130 latitude layers, and 10 depth layers with 50 threshold magnitude intervals. This makes a total of 9,750,000 binary decision variables. The following experiment results are based on considering 50 problem instances, where the numbers of layers and number of magnitude intervals are varied linearly between the previously mentioned minimum and maximum values. Each problem instance was solved on a 2.5GHz Intel i5 dual core laptop with 8GB RAM using CPLEX version 12.8. Each experiment was repeated using the *intlinprog* function of the Matlab optimization toolbox (see * in Table 1 for the instances that were successfully solved by Matlab). Each of these 50 experiments are repeated for four combinations of geometric constraints: *(i)* trigger rate constraint only (Equation (2)); *(ii)* trigger rate and depth constraints only (Equations (2) and (4)); *(iii)* trigger rate, depth constraints and slope constraints (Equations (2), (4), and (7)); and *(iv)* trigger rate and all geometric constraints (Equations (2), (4), (7), and (9)).

Table 1 shows that, for the first case, the maximum problem size that could be solved before an out-of-memory error was encountered consisted of 81 longitude layers, 70 latitude layers, 5 depth layers, and 24 magnitude intervals. This corresponds to a binary integer programming problem with 680,400 binary decision variables. However, when geometric constraints are included, the maximum problem size that could be solved decreases. When depth constraints are included, the maximum solvable problem size that could be solved consists of 74 longitude layers, 64 latitude layers, 4 depth layers, and 21 magnitude intervals, corresponding to a binary integer programming problem with 397,824 binary decision variables. When slope constraints are included, the maximum solvable problems size consists of 33,440 binary variables. Finally, when curvature constraints are also included, the maximum solvable problems size consists of 21,216 binary variables. Table 1 also shows that the efficiency increases with the fineness of the discretization of space and magnitude intervals, which is expected because a finer discretization allows the trigger solutions to better distinguish high risk areas from low risk areas that may be part of the same cube in coarser discretizations. Table 1 also shows that the inclusion of geometric constraints had a negligible impact on the efficiency, which is a desirable result since those constraints have the purpose of decreasing the volatility of the payments, ideally without decreasing risk transfer. The impact of the particular choices of the maximum slope (S) and curvature (C) parameters are investigated in more detail in Section 4.2. Given the evident limitations on the sizes of parametric designs that can be addressed using commercial solvers, we suggest the use of heuristic approaches, which are capable of solving larger problem instances in short computing times. A similar conclusion was also reached by Dodo et al. (2007) while working in an optimization problem related to earthquake-mitigation investment.

Longitude layers	Latitude layers	Depth layers	Magnitude intervals	Trigger rate constraint only		With depth constraints		Plus slope constraints		Plus curvature constraints	
				Efficiency	Solution time	Efficiency	Solution time	Efficiency	Solution time	Efficiency	Solution time
30	26	2	5	14.819*	0.335	14.772*	0.269	14.772*	6.698	14.772*	6.572
32	28	2	5	15.309*	0.203	15.309*	0.406	15.152*	8.300	15.153	17.225
34	30	2	6	16.567*	0.163	16.531*	0.297	16.531*	6.624	16.531	14.987
37	32	2	7	17.168*	0.188	17.168*	0.303	17.168	5.735	17.168	24.338
39	34	2	8	17.540*	0.159	17.506*	0.328	17.506	14.214	17.506	48.825
42	36	2	9	18.276*	0.191	18.265*	0.383	18.267	33.324	-	-
44	38	2	10	18.415*	0.503	18.415*	0.505	18.333	152.298	-	-
47	40	3	11	18.447	0.312	18.321	1.251	-	-	-	-
49	42	3	12	18.946	0.422	18.885	1.465	-	-	-	-
52	45	3	13	18.576	0.425	18.472	1.721	-	-	-	-
54	47	3	14	19.155	0.529	19.040	1.927	-	-	-	-
56	49	3	15	19.274	0.624	19.181	2.751	-	-	-	-
59	51	3	16	19.083	0.998	19.009	2.87	-	-	-	-
61	53	4	16	19.420	0.882	19.026	4.062	-	-	-	-
64	55	4	17	19.622	1.224	19.451	5.173	-	-	-	-
66	57	4	18	19.610	1.126	19.572	5.879	-	-	-	-
69	59	4	19	19.939	1.302	19.699	6.085	-	-	-	-
71	62	4	20	19.800	1.401	19.567	7.424	-	-	-	-
74	64	4	21	19.825	1.697	19.713	7.724	-	-	-	-
76	66	5	22	19.907	2.409	-	-	-	-	-	-
78	68	5	23	19.883	3.103	-	-	-	-	-	-
81	70	5	24	20.019	3.925	-	-	-	-	-	-
...	...	...	...	-	-	-	-	-	-	-	-
150	130	10	50	-	-	-	-	-	-	-	-

Table 1
CPLEX and MATLAB integer programming solver results ("*" MATLAB successfully solved the instance as well as CPLEX. "-" Out of memory error).

4.2. Heuristic Approaches

We now compare the binary integer programming approach with the proposed heuristic approaches for a range of geographic and magnitude discretizations, as well as maximum slope and curvature values. Whereas in Section 4.1 the limits of the binary integer programming formulation were tested for a range of discretization schemes and combinations of geometric constraints, in this section the experiments include all geometric constraints. This can be done due to the fact that those constraints do not increase the memory requirements of the proposed heuristic algorithms. The considered discretization schemes are displayed in the first column of Table 2.

In the formulation of the biased randomized approaches presented in Section 3.3, we need to determine the value for the parameter β, while controlling the balance between greediness and randomness. In this section we test the effect of the β parameter of the biased randomization algorithm. The discretization details for this experiment are as follows: 39 latitude layers, 45 longitude layers, 3 depth layers, and 10 magnitude intervals. The maximum slope is 3 and the maximum curvature is 15. Figure 1 shows that, for the case of the fully constrained problem, low values of the biased randomization parameter (between 0.05 and 0.5) lead to the highest values of efficiency. The number of biased randomization iterations was set to 500. Based on the results of Figure 1, the following experiments involving the biased randomization algorithm select, in each iteration, values of β uniformly distributed between 0.05 and 0.5. Different combinations of the included constraints may result in different appropriate values for β.

Table 2 provides a summary of the efficiency and solution times associated with each algorithm for each cube and magnitude interval discretization scheme, indicating the relative limits in terms of the solvable problem size for each algorithm. The results in each row of Table 2 are based on the average results from 25 experiments, consisting of each combination of 5 maximum slope values (S) and 5 maximum curvature values (C). Full details of the results for each of those individual experiments are displayed in Figures 2, 3, and 4.

discretization	GH		BR		BRWL		CPLEX	
$\{latitude, longitude,$	Efficiency		Efficiency		Efficiency		Efficiency	
$depth, magnitude\}$	average	maximum	average	maximum	average	maximum	average	maximum
$\{26, 30, 2, 5\}$	14.504	14.637	14.760	14.772	14.772	14.772	14.772	14.773
$\{33, 37, 3, 8\}$	16.048	16.674	16.122	16.774	16.491	16.818	16.614	16.820
$\{39, 45, 3, 10\}$	17.047	18.100	17.358	18.100	17.531	18.100	17.825	18.101
$\{52, 60, 5, 20\}$	18.050	19.208	18.259	19.208	18.397	19.208	19.228	19.228
$\{65, 75, 5, 25\}$	14.992	19.307	15.383	19.794	18.497	19.794	-	-
$\{90, 110, 8, 35\}$	15.329	19.802	15.549	19.937	15.805	19.958	-	-
$\{130, 150, 10, 50\}$	15.290	20.123	15.490	20.123	18.715	20.136	-	-
Average	15.894	18.264	16.132	18.387	17.173	18.398	17.110	17.231

Table 2
Heuristic algorithm performance.

The greedy heuristic (GH) and the BR algorithm were both able to solve all problem instances of all sizes. The biased randomization with learning algorithm (BRWL), with its increased memory re-

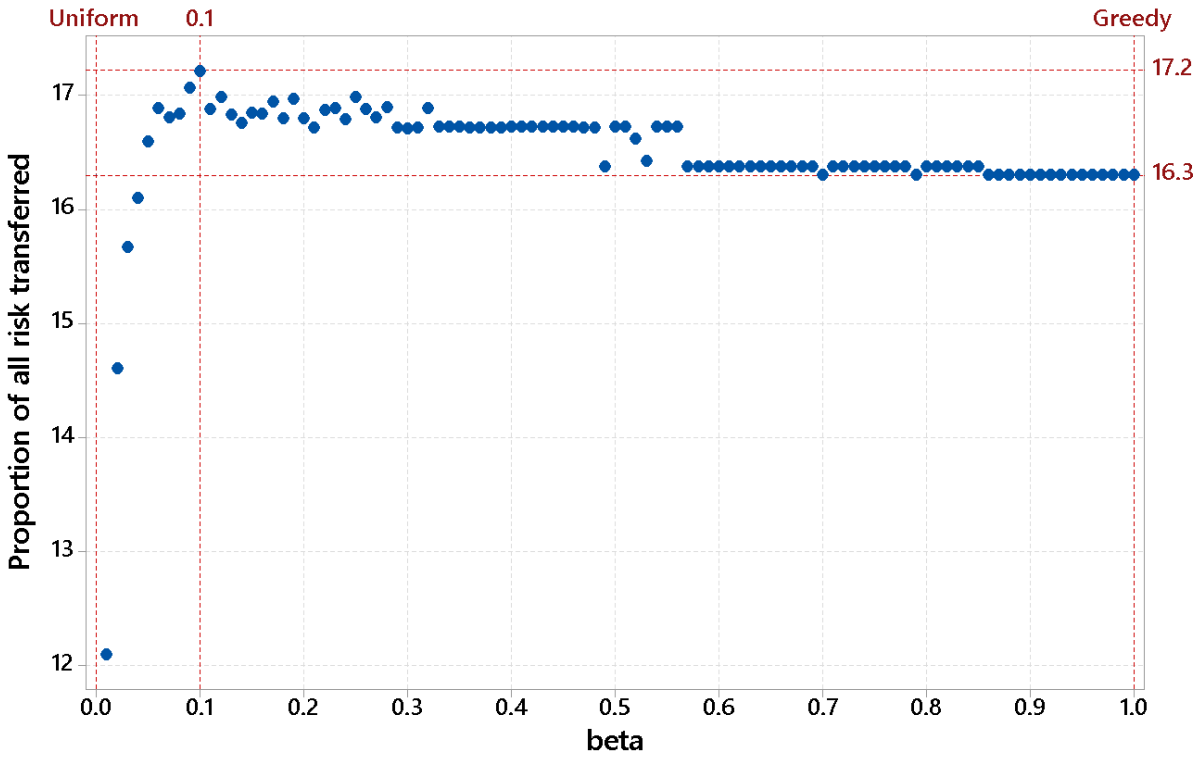

Fig. 1. The effect of the biased randomization beta parameter on the total proportion of risk transferred.

quirements, was able to solve problems of all sizes with the exception of 8 out of the 175 instances, including 4 for the third largest problem size (24,375 cubes) and 4 for the largest problem size (195,000 cubes). The binary integer programming approach (CPLEX) was able to solve all problem instances in the three smallest sets (1,560 to 5,130 cubes) as well as one instance involving 15,600 cubes, which was the one with the largest maximum slope and curvature values (i.e., least constrained with the greatest scope of row and column elimination in CPLEX's presolve procedure). Table 2 also shows that the proposed heuristic approaches allow for parametric solutions with higher efficiency (20.14%) than that of the largest problem size which can be solved using the CPLEX approach (20.02%), which was only possible when no geometric constraints were included in the binary integer programming formulation (see Table 1). Table 2 allows us to compare the proposed heuristic solution approaches and shows that the BRWL algorithm attains the highest average and maximum efficiency out of the proposed heuristic approaches for each of the 7 sets of problem sizes. Furthermore, the BR algorithm is able to attain maximum efficiency values which are close to those of BRWL and CPLEX, but achieves a significantly lower average efficiency than those algorithms for each set of problem sizes.

Figures 2 and 3 display, respectively, the efficiency values and solution times of each algorithm for each problem instance. It can be seen that, as cube sizes and magnitude interval sizes decrease, efficiency and solution times increase. For the finest discretizations, efficiency values approach a calculated upper

bound on efficiency for a 105 year return period. The upper bound is based on identifying the combination of events in the input earthquake catalog which maximize total risk without violating the trigger rate constraint, which can be formulated and solved as a knapsack problem. Figure 3 shows that the GH algorithm is three orders of magnitude faster than the BR and BRWL algorithms, which is because BR and BRWL both use a GH like algorithm in each of the 1,000 iterations of those algorithms –notice, however, that at least in the case of the BR algorithm, these iterations could be run in parallel, hence reducing the wall-clock time to the same as the one employed by the GH. The CPLEX solution times are always larger than those of the proposed heuristic approaches (by 2-3 orders of magnitude on average), noting that the maximum solution time for the CPLEX approach was capped at 3,600 seconds.

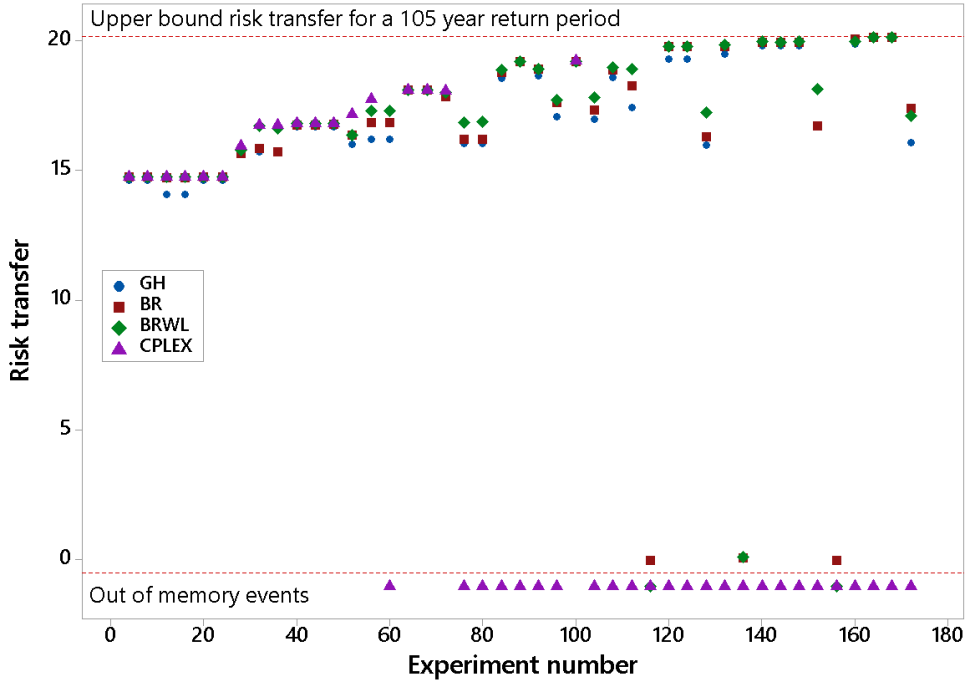

Fig. 2. Efficiency for the greedy heuristic (GH), biased randomization (BR), biased randomization with learning (BRWL) and CPLEX for a range of problem discretizations.

Figure 4 shows that, in some of the largest problem instances, low maximum slope and curvature values (highly constrained) lead to large reductions in efficiency. This is to be expected as the geometric smoothness comes at the cost of losing freedom to differentiate earthquakes within a geographic vicinity. Another trend that is visible in Figure 3 is that, for the largest problem size sets (groups of 25 experiments), the heuristic solution times decrease as the maximum slope and curvature values increase (less constrained). This is because more moves can be made by the $skip\,()$ function (Algorithm 4) before the smoothness constraints prevent further skipping.

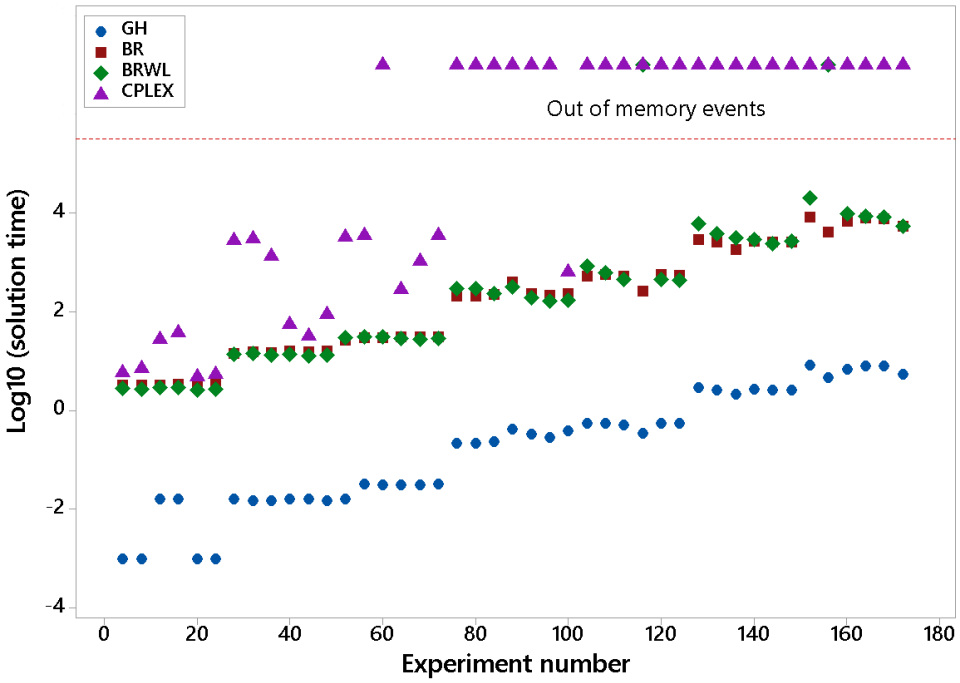

Fig. 3. Solution times for the greedy heuristic (GH), biased randomization (BR), biased randomization with learning (BRWL) and CPLEX for a range of problem discretizations.

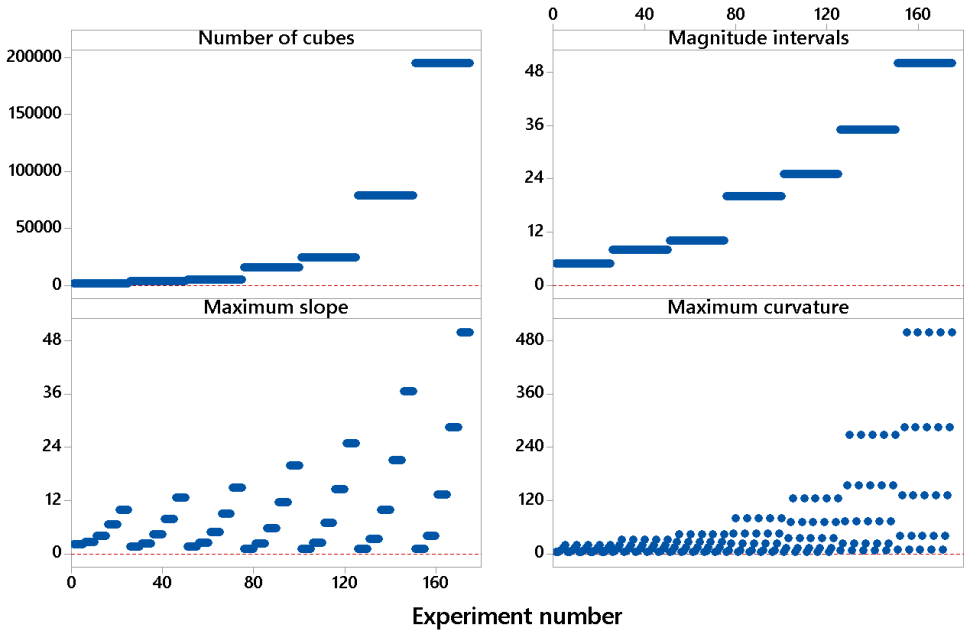

Fig. 4. Cube sizes, magnitude intervals, maximum slope and curvature inputs values for each experiment in Figure 2.

Figure 5 illustrates the effect of different maximum slopes on the resultant threshold magnitude maps for the first depth layer (deeper layers have similar maps but with higher magnitudes). In general, high maximum slopes lead to non-smooth threshold maps, whilst low maximum slopes lead to smooth threshold maps. Setting a low maximum slope has the potential beneficial effect of reduced over-fitting in the presence of limited simulation data. It remains to be studied how different combinations of slope and curvature constraints may offer optimal balances of efficiency and volatility.

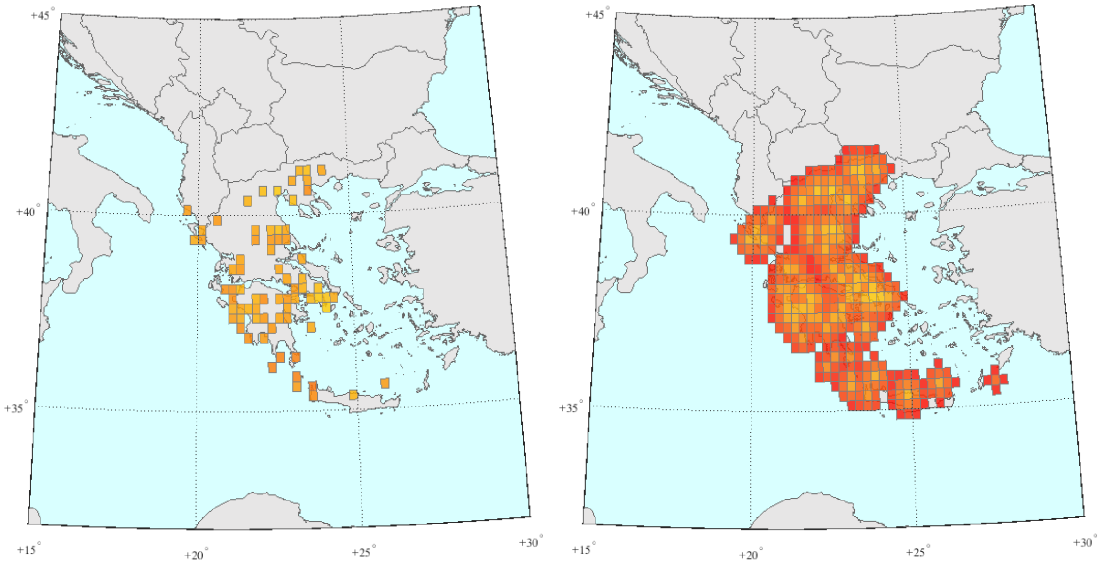

Fig. 5. First depth layer threshold magnitude maps for (left to right) $S = 100$ and $S = 5$ with $C = 100$, illustrating the effect of the maximum slope constraint parameter S.

5. Conclusions

This paper presents a novel design algorithm for 'cat-in-a-box' parametric earthquake risk transfer solutions. The design problem is formulated as an optimization problem with the objective of maximizing efficiency at a given annual trigger rate, a proxy for premium. Additional constraints are introduced to enforce certain beneficial features, such as resilience to model error.

A greedy heuristic has been used as a starting block to provide good solutions very rapidly, aided by a function designed to skip over magnitude thresholds that transfer no risk. A biased randomized extension of this greedy heuristic helps to increase efficiency, usually at the cost of increasing computational times. Extensions of the biased randomized algorithm, based on storing a look up table of strong partial solutions that can be selected at random as initial solutions in subsequent iterations, improve the efficiency and reliability of the algorithm. The heuristics presented have outperformed binary integer programming solvers, especially in scenarios with high dimensionality (lots of cubes).

In the case study presented for Greece, our results show that increasing the resolution in cube size and

threshold magnitude intervals also raises the efficiency of the mechanism, as desired. We have shown as well that, for problem instances with the finest levels of discretization, a binary integer programming approach often leads to out of memory errors. In contrast, the proposed fast heuristic approach is able to solve all problem instances, including those with the finest levels of discretization with close to optimal efficiency. Additional geometric constraints have a negligible impact on efficiency for moderate maximum slope and curvature values. This is an important observation, as it suggests that moderate smoothing may lead to solutions with high efficiency in risk transfer and low sensitivity to model error. The binary integer programming approach subjected to these additional constraints vastly under-performs the heuristics, thus decreasing the maximum problem size that can be solved.

These results are encouraging, as they support the creation of highly customized transactions with beneficial features at an acceptable computational cost. This, in turn, means that the problem is amenable of being automated for a very large number of potential insureds, and even for the retail market with reasonable response times –hence allowing, for instance, web services via an interactive web page. As future work, the proposed algorithms enable us to analyze in depth how the choice of smoothing constraints, such as those used here for slopes and curvatures, affects the performance and resilience of the trigger mechanism. This constitutes a natural extension to the work presented in this paper.

Acknowledgments

The authors are grateful for AIR Worldwide's permission to use the AIR Earthquake Model for Europe (Catrader v19.1) to perform all underlying risk analyses presented in this paper. This work has been partially funded by Guy Carpenter & Company, LLC. This support is gratefully acknowledged.

References

Almouhanna, A., Quintero-Araujo, C. L., Panadero, J., Juan, A. A., Khosravi, B., and Ouelhadj, D. (2020). The location routing problem using electric vehicles with constrained distance. *Computers & Operations Research*, 115:104864.

Artemis (2015). Acorn re ltd. (series 2015-1). *https://www.artemis.bm/deal-directory/acorn-re-ltd-series-2015-1/*, last accessed Feb 2020.

Artemis (2018). Pacific alliance cat bond to settle at usd 1.36bn, priced below guidance. *http://www.artemis.bm/blog/2018/02/05/pacific-alliance-cat-bond-to-settle-at-1-36bn-priced-below-guidance/*, last accessed Feb 2020.

Belloso, J., Juan, A. A., and Faulin, J. (2019). An iterative biased-randomized heuristic for the fleet size and mix vehicle-routing problem with backhauls. *International Transactions in Operational Research*, 26(1):289–301.

Cardenas, V., Hochrainer, S., Mechler, R., Pflug, G., and Linnerooth-Bayer, J. (2007). Sovereign financial disaster risk management: The case of mexico. *Environ. Haz.*, 7:40–53.

Cummins, J. D. (2008). Cat bonds and other risk-linked securities: State of the market and recent developments. *Risk Management and Insurance Review*, 11(1):23–47.

Cummins, J. D., Lalonde, D., and Phillips, R. D. (2004). The basis risk of catastrophic-loss index securities. *Journal of Financial Economics*, 71(1):77–111.

De Armas, J., Juan, A. A., Marquès, J. M., and Pedroso, J. P. (2017). Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society*, 68(10):1161–1176.

Dodo, A., Davidson, R. A., Xu, N., and Nozick, L. K. (2007). Application of regional earthquake mitigation optimization. *Computers & Operations Research*, 34(8):2478–2494.

Dominguez, O., Juan, A. A., de la Nuez, I. A., and Ouelhadj, D. (2016). An ILS-biased randomization algorithm for the two-dimensional loading HFVRP with sequential loading and items rotation. *Journal of the Operational Research Society*, 67(1):37–53.

Estrada-Moreno, A., Ferrer, A., Juan, A. A., Bagirov, A., and Panadero, J. (2019a). A biased-randomised algorithm for the capacitated facility location problem with soft constraints. *Journal of the Operational Research Society*, pages 1–17.

Estrada-Moreno, A., Fikar, C., Juan, A. A., and Hirsch, P. (2019b). A biased-randomized algorithm for redistribution of perishable food inventories in supermarket chains. *International Transactions in Operational Research*, 26(6):2077–2095.

Estrada-Moreno, A., Savelsbergh, M., Juan, A. A., and Panadero, J. (2019c). Biased-randomized iterated local search for a multiperiod vehicle routing problem with price discounts for delivery flexibility. *International Transactions in Operational Research*, 26(4):1293–1314.

Ferone, D., Gruler, A., Festa, P., and Juan, A. A. (2019). Enhancing and extending the classical GRASP framework with biased randomisation and simulation. *Journal of the Operational Research Society*, 70(8):1362–1375.

Ferrer, A., Guimarans, D., Ramalhinho, H., and Juan, A. A. (2016). A BRILS metaheuristic for non-smooth flow-shop problems with failure-risk costs. *Expert Systems with Applications*, 44:177–186.

Figueiredo, R., Martina, M. L., Stephenson, D. B., and Youngman, B. D. (2018). A probabilistic paradigm for the parametric insurance of natural hazards. *Risk Analysis*, 38(11):2400–2414.

Franco, G. (2010). Minimization of trigger error in cat-in-a-box parametric earthquake catastrophe bonds with an application to costa rica. *Earthquake Spectra*, 26(4):983–998.

Franco, G. (2013). Construction of customized payment tables for cat-in-a-box earthquake triggers as a basis risk reduction device. In *Proceedings of the International Conference on Structural Safety and Reliability (ICOSSAR)*, pages 5455–5462, New York, New York. doi:10.1201/b16387-793.

Franco, G. (2014). Earthquake mitigation strategies through insurance. In Beer, M., Kougioumtzoglou, I. A., Patelli, E., and Au, S.-K., editors, *Encyclopedia of Earthquake Engineering*, pages 1–18. Springer. doi:10.1007/978-3-642-36197-5_401-1, Berlin, Germany.

Franco, G., Guidotti, R., Bayliss, C., Estrada-Moreno, A., Juan, A. A., and Pomonis, A. (2019). Earthquake financial protection for greece: a parametric insurance cover prototype. In *Proceedings of the ICONHIC2019* 2^{nd} *International Conference on Natural Hazards and Infrastructure*, pages 1–8, Chania, Greece.

Franco, G., Tirabassi, G., Lopeman, M., Wald, D. J., and Siembieda, W. J. (2018). Increasing earthquake insurance coverage in california via parametric hedges. In *Proceedings of the* 11^{th} *US National Conference on Earthquake Engineering*, pages 4792–4803, Los Angeles, California. ISBN: 9781510873254.

Goda, K. (2013). Basis risk of earthquake catastrophe bond trigger using scenario-based versus station intensitybased approaches: A case study for southwestern british columbia. *Earthquake Spectra*, 29(3):757–775.

Goda, K. (2015). Seismic risk management of insurance portfolio using catastrophe bonds. *Computer-Aided Civil and Infrastructure Engineering*, 30:570–582.

Goda, K., Franco, G., Song, J., and Radu, A. (2019). Parametric catastrophe bonds for tsunamis: Cat-in-a-box trigger and intensity-based index trigger methods. *Earthquake Spectra*, 35(1):113–136.

Gonzalez-Neira, E. M., Ferone, D., Hatami, S., and Juan, A. A. (2017). A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 79:23–36.

Grasas, A., Juan, A. A., Faulin, J., de Armas, J., and Ramalhinho, H. (2017). Biased randomization of heuristics using skewed probability distributions: a survey and some applications. *Computers & Industrial Engineering*, 110:216–228.

Grossi, P., Kunreuther, H., and Windeler, D. (2005). An introduction to catastrophe models and insurance. In *Catastrophe modeling: A new approach to managing risk*, pages 23–42. Springer.

Guy Carpenter (2014). Chart: Gap between economic and insured losses. *Guy Carpenter Capital Ideas.*, Accessible on-line at http://www.gccapitalideas.com/2014/01/20/chart-gap-between-economic-and-insured-losses/.

Guy Carpenter (2017). Factors contributing to the gap between economic and insured losses. *Guy Carpenter Capital Ideas.*, Accessible on-line at http://www.gccapitalideas.com/2017/11/30/factors-contributing-to-the-gap-between-economic-and-insured-losses-2/.

Jolliffe, I. T. and Stephenson, D. B. (2012). *Forecast verification: a practitioner's guide in atmospheric science*. John Wiley & Sons.

King, A., Middleton, D., Brown, C., Johnston, D., and Johal, S. (2014). Insurance: its role in recovery from the 2010–2011

canterbury earthquake sequence. *Earthquake Spectra*, 30(1):475–491.

Lai, T., Nasseri, A., Katiyar, V., Tang, Y., Guin, J., and Towashiraporn, P. (2012). A uniform framework of seismic vulnerability assessment and its application in seismic risk analysis of european countries. In *Proceedings of the* 15^{th} *World Conference on Earthquake Engineering*, pages 1–10, Lisbon, Portugal.

Mitchell-Wallace, K., Jones, M., Hillier, J., and Foote, M. (2017). *Natural catastrophe risk management and modelling: A practitioner's guide*. John Wiley & Sons.

Pucciano, S., Franco, G., and Bazzurro, P. (2017). Loss predictive power of strong motion networks for usage in parametric risk transfer: Istanbul as a case study. *Earthquake Spectra*, 33(4):1513–1531.

Quintero-Araujo, C. L., Caballero-Villalobos, J. P., Juan, A. A., and Montoya-Torres, J. R. (2017). A biased-randomized meta-heuristic for the capacitated location routing problem. *International Transactions in Operational Research*, 24(5):1079–1098.

Rong, Y., a. M. M., Shen-Tu, B., and Shabestari, K. (2011). Magnitude problems in historical earthquake catalogues and their impact on seismic hazard assessment. *Geophysical Journal International*, 187(3):1687–1698.

Ross, D. and Williams, J. (2012). Basis risk from the cedant's perspective. *The Handbook of Insurance-Linked Securities*, pages 49–64.

Wald, D. J. and Franco, G. (2016). Money matters: Rapid post-earthquake financial decision-making. *Natural Hazards Observer*, 40(7):24–27.

Wald, D. J. and Franco, G. (2017). Financial decision-making based on near real time earthquake information. In *Proceedings of the* 16^{th} *World Conference on Earthquake Engineering*, pages 1–13, Santiago de Chile, Chile.