Worst Case Search over a Set of Forecasting Scenarios Applied to Financial Stress-Testing

Steffen Finck
Josef Ressel Center for Applied Scientific Computing
in Energy, Finance and Logistics,
FH Vorarlberg University of Applied Sciences
Dornbirn, Austria
steffen.finck@fhv.at

ABSTRACT

Stress testing is part of today's bank risk management and often required by the governing regulatory authority. Performing such a stress test with stress scenarios derived from a distribution, instead of pre-defined expert scenarios, results in a systematic approach in which new severe scenarios can be discovered. The required scenario distribution is obtained from historical time series via a Vector-Autoregressive time series model.

The worst-case search, i.e. finding the scenario yielding the most severe situation for the bank, can be stated as an optimization problem. The problem itself is a constrained optimization problem in a high-dimensional search space. The constraints are the box constraints on the scenario variables and the plausibility of a scenario. The latter is expressed by an elliptic constraint.

As the evaluation of the stress scenarios is performed with a simulation tool, the optimization problem can be seen as black-box optimization problem. Evolution Strategy, a well-known optimizer for black-box problems, is applied here. The necessary adaptations to the algorithm are explained and a set of different algorithm design choices are investigated. It is shown that a simple box constraint handling method, i.e. setting variables which violate a box constraint to the respective boundary of the feasible domain, in combination with a repair of implausible scenarios provides good results.

CCS CONCEPTS

• **Computing methodologies** → **Rare-event simulation**; • **Theory of computation** → *Bio-inspired optimization*; • **Applied computing** → *Decision analysis*; Operations research;

KEYWORDS

risk management, simulation-based optimization, high-dimensional search space, constrained optimization, stress-testing

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3326835

ACM Reference Format:
Steffen Finck. 2019. Worst Case Search over a Set of Forecasting Scenarios Applied to Financial Stress-Testing. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3319619.3326835

1 INTRODUCTION

The real-world problem investigated comes from the area of financial stress testing. This particular research area gained quite some interest after the financial crises in the last decade and with the subsequent policies put into action by various central and national regulatory authorities. It is beyond the scope of this paper to present a detailed survey on all aspects of stress testing financial institutions and the interested reader is referred to the specialized literature like [28, 31, 35], publications like [2] describing the setup for stress tests, and many other texts.

Stress tests can be performed as systemic stress tests and as stress tests on the institutional level. Systemic stress tests [13] deal with the connections and contagions in a banking network. The work in this paper is related to stress tests on an institutional level. The general setup of such a stress test is to examine the stability and operating ability of a bank under some expert-defined stress scenarios. Each scenario consists of a set of risk parameters describing a particular economic situation. This procedure is called "1st generation stress test". The disadvantage of that procedure is that pre-defined stress scenarios may miss one or more severe scenarios or that some of the scenarios may be implausible for the bank given their respective business segments.

In a "2nd generation stress test" one aims to perform *systematic* stress-testing [10, 37]. This means, instead of considering a (usually quite small) set of pre-defined scenarios, one would like to sample from the respective distribution of the scenarios. Systematic stress-testing further includes the restriction of using *plausible* scenarios [11]. The plausibility is related to the changes in the risk parameters and their deviation from the mean of the distribution. For example, if a set of risk parameters changes differently than defined by the covariance matrix of the distribution, the scenario is less plausible.

Given the distribution for the scenarios, one would like to find the scenario yielding the worst possible business situation for the bank. However, as will be discussed later, given the computational requirements and the setup for the evaluation of the state of the bank, this aim appears unrealistic. The idea is to use an optimization approach which obtains more severe scenarios compared to simply drawing samples from the distribution and which can efficiently

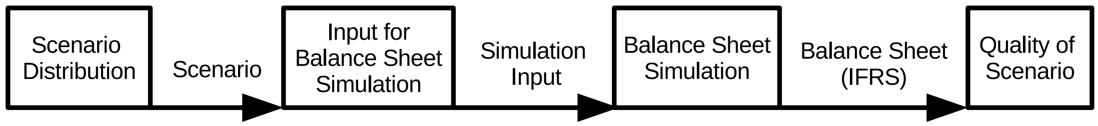

Figure 1: Flowchart for evaluation of a single scenario.

deal with the challenges posed by the problem evaluation. While the global optimum may not be reached (and may not be unique), the found worst case scenarios will provide valuable information to the bank's risk management.

Using an optimization approach is similar to robust optimization [3, 5] or worst-case optimization [14, 19, 25, 41]. However, while in these approaches the best performance in a worst case scenario is searched for, here, only the worst case scenario is of interest and the decisions made by the bank are always the same, i.e. independent of the scenario. One example for a similar investigation, but with the intention to investigate the algorithm design and using mathematical test functions is presented in [8]. The use of a quasi Monte-Carlo approach is discussed in [12, 30], however, the problem considered less complex and less computationally demanding.

Figure 1 shows the flowchart for the evaluation of a single scenario. From the scenario distribution, a scenario is drawn and then transformed into the input required for the simulation of the state of the bank. The output of this simulation are the end-of-year balance sheets for the next five years, from which the scenario quality will be derived. While there exist several models for simulating the positions of a balance sheet [7, 18], the used simulation is tailored to the bank and its business of operation. The simulation tool will be considered as a black-box for the proposed optimization approach. This assumption is valid, since the underlying model is too complex to state it in an analytical form and involves several computation steps using sub models. This prevents the use of a standard gradient method as optimization method. Since the problem is real-valued and the scenario distribution is specified as a multivariate normal distribution, Evolution Strategies [4, 21, 32, 36] (ESs) are applied. In ES, the mutation step is based on sampling from a multivariate normal distribution which allows to easily incorporate the problem knowledge into the optimization method.

The rest of the paper is organized as follows: The optimization problem is formally stated in Section 2. This section further contains the description on how the scenario distribution is derived from the available data. The optimization algorithm and the respective design choices are stated in Section 3. Section 4 presents and discusses first results, and in Section 5 some implementation concerns are discussed. The conclusion for the paper and an outlook are provided in Section 6.

2 WORST-CASE PROBLEM SETUP

In this section the optimization problem and the process of deriving the scenario distribution are presented.

2.1 Optimization Problem

The optimization problem is stated as

$$\arg\min_{s \in \mathcal{S} \subseteq \mathbb{R}^n} L(\boldsymbol{s}) \tag{1}$$

where $L : \mathbb{R}^n \to \mathbb{R}$ is the loss function and $\boldsymbol{s} \in \mathbb{R}^n$ is a scenario from the set of plausible scenarios $\mathcal{S}$. The loss function provides the quality of a scenario and its value is derived from the balance sheet simulation. A balance sheet consists of several positions and provides various information on the state of the bank. A complete description of the balance sheet positions and how they are calculated is beyond the scope of the paper. The interested reader is referred to specialized literature like [33, 34] and the IFRS standard [16].

For the purpose of the optimization, a design decision was made that L should be a scalar. From the balance sheet positions the Common Equity Tier 1 Ratio, the ratio between the bank's core equity capital and its risk-weighted assets, was chosen. Since the simulation will provide a balance sheet for each simulated year, the smallest of the obtained Common Equity Tier 1 Ratios is used as the value provided by $L(\boldsymbol{s})$. Next, the constraints for the optimization problem are stated.

The scenarios are drawn from a distribution and can undergo a transformation step afterwards to create the data used as input for the simulation (see Section 2.2). To better distinguish between these two states, $\tilde{\boldsymbol{s}}$ will indicate data stemming directly from the distribution (also referred to as genotype of a scenario) and $\boldsymbol{s}$ will indicate data used as simulation input (also referred to as phenotype of a scenario). Note, both states have the same dimensionality n.

As stated before, the scenarios should be plausible. To measure the plausibility of a scenario the square of the Mahalanobis distance is used [10],

$$(\tilde{\boldsymbol{s}} - \boldsymbol{m})^{\mathrm{T}} \boldsymbol{C}^{-1} (\tilde{\boldsymbol{s}} - \boldsymbol{m}) \leq \kappa, \tag{2}$$

where $\boldsymbol{m}$ and $\boldsymbol{C}$ are mean and covariance matrix of the distribution, respectively. Note, the plausibility is measured on the genotype of a scenario. The plausibility threshold κ is a user-defined value. Choosing κ can be based on existing information, however, in most cases it is not possible to set κ in this way. Another option is to set κ based on the fact that the left-hand side of Eq. (2) follows a chi-square distribution. Then the threshold can be set as

$$\kappa = n + \sqrt{2n},$$

which equals mean plus one standard deviation for a chi-square distribution with n degrees-of-freedom. This value for κ is used in this work.

From a theoretical point of view, the plausibility constraint in Eq. (2) is sufficient for the problem description. However, the user can change some parameters in the process of deriving the scenario distribution. Hence, the same data set may yield different scenario distributions. To avoid using scenarios where some variables take values which are deemed unrealistic by an expert in the field, bounds for the scenario variables are employed. These bounds are stated for the phenotype of a scenario. Each variable is bounded by a lower and an upper bound, i.e.

$$s_{\min,i} \leq s_i \leq s_{\max,i} \text{ for } i \in \{1, 2, \ldots, n\}. \tag{3}$$

Before the optimization algorithm is presented, the next section describes how the scenario distribution is obtained.

2.2 Deriving the sampling distribution from historical data

The data required for the balance sheet simulation consist of various parameters describing the current state of the bank, growth rates indicating future developments, and possibly singular decisions made by the bank[1]. The current state is determined by aggregate information on the actual volumes in the accounts of the bank's customers (savings and liabilities), the current values of certain economic variables (credit index values, exchange rates, stock index values), and the current values of the balance sheet positions. The growth rates describe how the current values, i.e. the economic variables, the volumes of savings and liabilities, will change over time on a yearly basis. The simulation input is finally defined by more than 100 different parameters, several of them being time series.

Given the large number of input parameters, it is not feasible to draw all input variables from a distribution. First, several variables are not driven by the economy, but rather depend on the bank's management choices. Therefore, they will not be included in the set of parameters of interest. Second, any result obtained should remain "explainable". This means given the parameters of a scenario experts should be able to validate the obtained simulation result. Together with experts from the bank, a selection of parameters was made. The following parameters were chosen: exchange rates between the Euro and the US Dollar, the Japanese Yen and the Swiss Francs; active and passive side margins for three business sections (private, commercial, treasury) of the bank; probability of default for each of the three business sections; index values for interest rates (3 month euribor rate, 5 and 10 year euro swap rates); iTraxx Europe Main as index for the credit market; and Euro Stoxx 50 as index for the stock market. Overall, this set comprises $k = 17$ different *risk parameters*. All other required input parameters are defined by an existing expert-defined scenario.

In an ideal setup, risk parameters are not influenced by a bank's decisions. While this holds for most of the chosen risk parameters, the margins are not completely independent of such decisions. Nevertheless, they are part of the selected risk parameters since they represent a main driver for the bank's business.

For each risk parameter, monthly data for the last 10 years were used for building a time series model. To improve the quality of the time series model, risk parameters can be independently transformed. The available transformations include various log-based transformations (see Appendix A) or by using the differences instead of the absolute values. The choice of applying transformations is up to the user. To indicate the transformation, the following operator

$$\mathcal{T}(\tilde{\boldsymbol{s}}, \hat{\boldsymbol{y}}) = \boldsymbol{s} \tag{4}$$

is defined. The operator $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ defines the transformation from the time series model to the balance sheet simulation input. The inverse operator $\mathcal{T}^{-1}$ is thus applied to the historical data. The operator in Eq. (4) includes the known prior values $\hat{\boldsymbol{y}}$, i.e. the last known values before the start of the simulation. These values are required as initial values for the time series model.

After transforming the risk parameters, a Vector-Autoregressive (VAR) time series model is estimated [26]. The standard form for a VAR-model with a lag of one time step is

$$\boldsymbol{y}^{(t+1)} = \boldsymbol{\nu} + \boldsymbol{A}\boldsymbol{y}^{(t)} + \boldsymbol{\epsilon}. \tag{5}$$

In Equation (5) $\boldsymbol{y}^{(t)} \in \mathbb{R}^k$ is the vector of risk parameters at time t, $\boldsymbol{\nu} \in \mathbb{R}^k$ is a vector of constants, $\boldsymbol{A} \in \mathbb{R}^{k \times k}$ is the lag-coefficient matrix, and $\boldsymbol{\epsilon} \in \mathbb{R}^k$ is the error or disruption term. The error term follows a multivariate normal distribution and is independent of the time step, i.e.

$$\boldsymbol{\epsilon}^{(0)} = \boldsymbol{\epsilon}^{(1)} = \boldsymbol{\epsilon}^{(t)} = \boldsymbol{\epsilon} \sim \mathcal{N}\left(\boldsymbol{0}, C_\epsilon\right).$$

The VAR-model in Eq. (5) can be stated as a multivariate normal distribution given that the error terms are normally distributed. Then a scenario is defined as

$$\tilde{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{y}^{(0)} \\ \boldsymbol{y}^{(1)} \\ \vdots \\ \boldsymbol{y}^{(T)} \end{bmatrix}. \tag{6}$$

The dimensionality of $\tilde{s}$ is $n = kT$. The parameters of the multivariate normal distribution, i.e. the scenario distribution, are the mean

$$\boldsymbol{m} = \mathrm{E}\left[\tilde{s}\right] = \begin{bmatrix} \boldsymbol{\nu} + \boldsymbol{A}\hat{\boldsymbol{y}} \\ \left(\boldsymbol{A}^0 + \boldsymbol{A}\right)\boldsymbol{\nu} + \boldsymbol{A}^2\hat{\boldsymbol{y}} \\ \vdots \\ \left(\sum_{i=0}^{T} \boldsymbol{A}^i\right)\boldsymbol{\nu} + \boldsymbol{A}^{T+1}\hat{\boldsymbol{y}} \end{bmatrix} \tag{7}$$

and the covariance matrix

$$\boldsymbol{C} = \begin{bmatrix} \tilde{C}_{0,0} & \tilde{C}_{0,1} & \ldots & \tilde{C}_{0,T} \\ \tilde{C}_{1,0} & \tilde{C}_{1,1} & \ldots & \tilde{C}_{1,T} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{C}_{T,0} & \tilde{C}_{T,1} & \ldots & \tilde{C}_{T,T} \end{bmatrix} \tag{8}$$

where

$$\tilde{C}_{m,n} = \left(\tilde{C}_{n,m}\right)^{\mathrm{T}} = \sum_{i=0}^{\min(m,n)} \boldsymbol{A}^{m-i} C_\epsilon \left(\boldsymbol{A}^{n-i}\right)^{\mathrm{T}}. \tag{9}$$

With this multivariate normal distribution, $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$, a simple worst-case search can be performed by repeatedly drawing samples and evaluating them. This may be efficient for low dimensionalities. For the stress test case, $T = 55$ time steps have to be forecasted[2]. When using the full set of risk parameters ($k = 17$) the search space

[1]While not important for the work described, examples for such decisions can be the selling of business parts or various ways for a large capital procurement.

[2]For the first year, data until end of May exists.

dimensionality is $n = kT = 17 \cdot 55 = 935$ and repeatedly drawing samples will not be efficient.

In the next section the algorithm for the worst-case search is proposed.

3 EVOLUTION STRATEGY AS WORST CASE SEARCH OPERATOR

Algorithm 1 Evolution Strategy with fixed, non-isotropic covariance matrix and cumulative step size adaptation.

1: **procedure** ES($\lambda, \mu, \text{FE}_{\max}, \sigma_{\min}, \kappa, \boldsymbol{s}_{\min}, \boldsymbol{s}_{\max}, \boldsymbol{C}, \boldsymbol{m}, p, \hat{\boldsymbol{y}}, \gamma$)
2: $\quad$ FEs $\leftarrow 0$, $\boldsymbol{p_s} \leftarrow \boldsymbol{0}$, $n \leftarrow$ length of $\boldsymbol{m}$
3: $\quad w_i \leftarrow \frac{\log\left(\frac{\lambda+1}{2}\right)-\log(i)}{\sum_{j=1}^{\mu}\log\left(\frac{\lambda+1}{2}\right)-\log(j)} \; \forall i \in 1,\ldots,\mu$
4: $\quad \mu_{\text{eff}} \leftarrow \left(\sum_{i=1}^{\mu} w_i^2\right)^{-1}$, $c_\sigma = \frac{\mu_{\text{eff}}+2}{n+\mu_{\text{eff}}+5}$
5: $\quad$ set initial solution $\boldsymbol{m_c}$ ▷ see Section 3.1
6: $\quad$ set initial mutation strength σ ▷ see Algorithm 2
7: $\quad$ **repeat**
8: $\quad\quad$ **for** $l := 1$ to λ **do**
9: $\quad\quad\quad z_l \leftarrow \mathcal{N}(\boldsymbol{0}, \boldsymbol{C})$
10: $\quad\quad\quad \tilde{s}_l \leftarrow \boldsymbol{m_c} + \sigma z_l$
11: $\quad\quad\quad \boldsymbol{s_l} \leftarrow \mathcal{T}(\tilde{s}_l, \hat{\boldsymbol{y}})$
12: $\quad\quad\quad \boldsymbol{z_l}, \tilde{\boldsymbol{s}}_l, \boldsymbol{s_l} \leftarrow$ constraint handling ▷ see Section 3.3
13: $\quad\quad\quad f_l^p \leftarrow \max\left[(\tilde{\boldsymbol{s}}_l - \boldsymbol{m})^{\mathrm{T}} \boldsymbol{C}^{-1}(\tilde{\boldsymbol{s}}_l - \boldsymbol{m}) - \kappa, 0\right]$
14: $\quad\quad\quad f_l \leftarrow L(\boldsymbol{s_l}) + \gamma f_l^p$
15: $\quad\quad\quad$ FEs $\leftarrow$ FEs + 1
16: $\quad\quad \boldsymbol{m_c} \leftarrow \sum_{i=1}^{\mu} w_i \tilde{\boldsymbol{s}}_{i:\lambda}$
17: $\quad\quad \boldsymbol{p_s} \leftarrow (1-c_\sigma)\boldsymbol{p_s} + \sqrt{\mu_{\text{eff}} c_\sigma(2-c_\sigma)}\, C^{-\frac{1}{2}} \left(\sum_{i=1}^{\mu} w_i z_{i:\lambda}\right)$
18: $\quad\quad \sigma \leftarrow \sigma \mathrm{e}^{\frac{c_\sigma}{2}\left(\frac{\|\boldsymbol{p_s}\|^2}{n}-1\right)}$
19: $\quad$ **until** (FEs $\geq \text{FE}_{\max}$) $\vee$ ($\sigma < \sigma_{\min}$)

When selecting an Evolutionary Algorithm (EA) for a real-world problem, an important question is how the problem representation must be defined to fit the algorithm. The problem considered deals with high-dimensional samples from a multivariate normal distribution, thus Evolution Strategies (ESs) [4, 21, 32, 36] fit quite well. Especially in intermediate recombination ESs [4], abbreviated as $(\mu/\mu_I, \lambda)$-ES, the mutation operator is defined as drawing samples from a distribution around the current centroid of the parent population (i.e. the centroid is the mean of the distribution). The algorithm then updates the mean and the covariance matrix [23] during the iterative process to find improved solutions. The same approach is applied for the worst-case search, with the exception of the covariance matrix adaptation. Since the covariance matrix is known from the time series model (cf. Eq. (8)) and further is used within the plausibility constraint (cf. Eq. (2)), this fixed covariance matrix will be used in the mutation step. The pseudo-code for the ES used is given in Algorithm 1.

The first algorithm design decision concerns the choice of a specific ES variant and the respective algorithmic parameters. As stated before, the $(\mu/\mu_I, \lambda)$-ES was selected, which requires the number of offspring (λ), the number of parents (μ), and the recombination weights (w_i). Part of any ES is the mutation strength (σ) adaptation. Preliminary experiments showed that cumulative step size adaptation [22] works quite well. The required parameters ($\boldsymbol{p_s}, \mu_{\text{eff}}, c_\sigma$) are set to their default values as given in [20] (which also includes default values for the recombination weights).

The remaining parameters required are:

- for the initialization procedure (line 6) - the user-defined probability p
- for mutation (line 10) - covariance matrix $\boldsymbol{C}$
- for the genotype-phenotype transformation (line 11) - the known prior values of the risk parameters $\hat{\boldsymbol{y}}$
- for the evaluation of the constraints (lines 12 and 13) - the plausibility threshold (κ), the mean of the distribution ($\boldsymbol{m}$), and the parameter bounds ($\boldsymbol{s}_{\min}$ and $\boldsymbol{s}_{\max}$)
- for the penalty function (line 14) - penalty parameter γ
- as termination criteria (line 19) - budget of function evaluations ($\text{FE}_{\max}$) and minimal allowable mutation strength ($\sigma_{\min}$)

The pseudo-code in Algorithm 1 agrees in most parts with the $(\mu/\mu_I, \lambda)$-ES with cumulative step-size adaptation. One may note that the selection step is not specifically stated. In ES selection is done by ordering the offspring with respect to the fitness. This ordering is indicated by the order statistic notation $i : \lambda$ (see, for example, line 16), meaning the ith best offspring out of all λ offspring.

Finally, some remarks on the required adaptations for the use as worst-case search operator:

- The initialization procedures for the initial centroid $\boldsymbol{m_c}$ of the population (line 5) and the initial mutation strength (line 6) are described in Section 3.1 and Section 3.2, respectively.
- The mutation operator (see line 9) uses a non-isotropic distribution.
- Except for the evaluation of a scenario (see line 14) and the box constraint handling (see Section 3.3 and line 12), all operators work at the genotype level.
- The constraint handling approaches for infeasible scenarios (violating the box constraints in Eq. (3)) and for implausible scenarios (violating the plausibility constraint in Eq. (2)) are given in Section 3.3.
- The use of the penalty function for implausible scenarios, see line 14, is discussed in Section 3.3.

Next, the initialization procedures and the constraint handling approaches are described.

3.1 Finding an initial solution

One way for setting the initial centroid would be to always start with the mean of the known multivariate normal distribution. However, using different initial centroids may steer the algorithm towards different areas of the search space. A method for obtaining different initial centroids, is to repeatedly sample from

$$\boldsymbol{m_c} \sim \mathcal{N}(\boldsymbol{m}, a^2\boldsymbol{C})$$

starting with $a = 1$. After each sample the plausibility (cf. Eq. (2)) and box constraint (cf. Eq. (3)) satisfaction are verified. If one of those is violated, a is reduced in a deterministic manner. The designed mechanism is to reduce a by 0.1 if none of 10 samples did provide a feasible and plausible scenario.

The procedure is repeated until a feasible and plausible scenario is obtained. Given that a for a well-defined problem $\boldsymbol{m}$ is feasible and plausible by definition, if $a \to 0$ then $\boldsymbol{m_c} \to \boldsymbol{m}$ holds. Note, this step does not require any function evaluations.

3.2 Setting the initial mutation strength

The initial mutation strength should be selected such that at least a certain number of offspring will fulfill the plausibility constraint. Since in practice the set of risk parameters is allowed to vary, no fixed value for the initial mutation strength can be given here. However, by considering that the plausibility as defined in Eq. (2) follows a *scaled non-central* chi-square distribution, a method for selecting the initial mutation strength can be devised. The reason why the plausibility follows a scaled non-central chi-square distribution is due to the fact that the offspring are now drawn from $\mathcal{N}(\boldsymbol{m_c}, \sigma \boldsymbol{C})$ (see line 10 in Algorithm 1) and $\boldsymbol{m} \neq \boldsymbol{m_c}$ and $\sigma \neq 1$ hold in general. The non-centrality parameter for this distribution is

$$\Lambda = (\boldsymbol{m_c} - \boldsymbol{m})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{m_c} - \boldsymbol{m}). \tag{10}$$

The scaled non-central chi-square distribution can be expressed in terms of an unscaled non-central chi-square distribution χ^2_{nc} as

$$\sigma^2 \chi^2_{nc} \left(n, \frac{\Lambda}{\sigma^2} \right),$$

with distribution parameters n as the degree-of-freedom and $\frac{\Lambda}{\sigma^2}$ as the non-centrality parameter.

Since $n \gg 1$, the χ^2_{nc}-distribution can be approximated by a normal distribution according to the central limit theorem. This proves in the experiments also numerically more accurate than using the non-central chi-square distribution. The parameters of this normal distribution are

$$m = n + \frac{\Lambda}{\sigma^2}$$

as mean and

$$v^2 = 2n + \frac{4}{\sigma^2 \Lambda}$$

as variance. Note, both parameters depend on σ. The scaling must be considered when determining properties of the normal distribution. For example, the value of the cumulative distribution function at point x is $\Phi\left(\frac{x}{\sigma^2}\right)$.

Now one can determine σ for given plausibility threshold κ and a user-defined probability p from

$$\Phi\left(m, v^2, \frac{\kappa}{\sigma^2}\right) - p = 0 \tag{11}$$

with

$$\Phi(m, v^2, x) = \frac{1}{\sqrt{2\pi v^2}} \int_{-\infty}^{x} \exp\left(-\frac{(t-m)^2}{2v^2}\right) \mathrm{d}t. \tag{12}$$

The function on the left-hand side in Eq. (11) is a monotone function. Hence, the root of this function can be determined by a line search procedure. In the experiments Brent's method [9] is applied as line search. This method requires the definition of an interval which contains the root. The respective method for finding the interval is given in Algorithm 2.

What remains is setting a value for the user-defined probability p. Since the search of the ES will be performed in high-dimensional search space, diversity of the population plays a crucial role. In ESs diversity is directly related to the mutation strength, thus one would like to choose a large as possible initial mutation strength while at the same time ensuring at least one plausible solution. This can be achieved by using

$$p = \frac{2}{\lambda}$$

as desired probability. Note, this choice defines that *on average* two of the offspring are plausible.

Algorithm 2 Algorithm for determining the initial mutation strength (IMS) and the function for calculating the difference between the user-defined probability p and the probability from the respective normal distribution (function ProbDiff). The cumulative probability density function $\Phi(m, v^2, x)$ is defined in Eq. (12). Any line search technique can be used as root finder in line 18. The used choice for the initial α (i.e. when calling IMS) is $\alpha = 1$.

1: **function** ProbDiff$(n, \Lambda, \alpha, \kappa, p)$
2: $\quad m \leftarrow n + \frac{\Lambda}{\alpha}$
3: $\quad v^2 \leftarrow 2n + \frac{4}{\alpha\Lambda}$
4: $\quad$ **return** $\Phi\left(m, v^2, \frac{\kappa}{\alpha}\right) - p$
5: **procedure** IMS$(n, \alpha, \kappa, p, \boldsymbol{m}, \boldsymbol{m_c}, \boldsymbol{C})$
6: $\quad \Lambda \leftarrow (\boldsymbol{m_c} - \boldsymbol{m})^{\mathrm{T}} \boldsymbol{C}^{-1} (\boldsymbol{m_c} - \boldsymbol{m})$
7: $\quad r \leftarrow$ ProbDiff$(n, \Lambda, \alpha, \kappa, p)$
8: $\quad$ **if** $r < 0$ **then**
9: $\quad\quad \alpha_l \leftarrow \frac{\alpha}{2}$
10: $\quad\quad$ **while** ProbDiff$(n, \Lambda, \alpha_l, \kappa, p) < 0$ **do**
11: $\quad\quad\quad \alpha_l \leftarrow \frac{\alpha_l}{2}$
12: $\quad\quad \alpha_r \leftarrow 2\alpha_l$
13: $\quad$ **else**
14: $\quad\quad \alpha_r \leftarrow 2\alpha$
15: $\quad\quad$ **while** ProbDiff$(n, \Lambda, \alpha_r, \kappa, p) > 0$ **do**
16: $\quad\quad\quad \alpha_r \leftarrow 2\alpha_r$
17: $\quad\quad \alpha_l \leftarrow \frac{\alpha_r}{2}$
18: $\quad \alpha \leftarrow$ line search in $[\alpha_l, \alpha_r]$ of ProbDiff
19: $\quad$ **return** $\sqrt{\alpha}$

3.3 Repair of infeasible or implausible scenarios

How to handle constraints is an important decision for the algorithm design. A first question to be answered is whether infeasible solutions can be evaluated by the simulation tool. Depending on the answer, methods for handling and dealing with infeasible solutions can be devised. In case of the balance sheet simulation, it holds that infeasible solutions can be evaluated. This allows to use the infeasible search space for the approach towards the optimum.

With those considerations in mind, the box constraints of Eq. (3) will be considered first. Since the box constraints can be evaluated independent of the balance sheet simulation, a repair of infeasible scenarios is possible. There exist various approaches for handling box (or bound) constraints, see for example [6] and the reference therein. The commonality of these methods is to replace the infeasible offspring with a new candidate solution in or at the boundary of the feasible domain. One of the simplest procedures is setting the value of a parameter to the boundary if it violates the respective box constraint. For the stress scenarios, this is achieved by

$$s_i = \begin{cases} s_{\min,i} & \text{if } s_i < s_{\min,i} \\ s_{\max,i} & \text{if } s_i > s_{\max,i} \\ s_i & \text{else} \end{cases} . \tag{13}$$

For the stress testing of the bank, this is particular enticing since one expects that the worst case solution will be on the boundary. However, since single variables of a scenario are changed, Eq. (13) will potentially yield an increase in the Mahalanobis distance and therefore in a possible violation of the plausibility constraint.

Another approach is to determine the intersection points between the infeasible scenario and the planes defined by the box constraints. The first plane to be punctured defines then the feasible scenario

$$\boldsymbol{s}_{\text{feas}} = (1-\beta)\boldsymbol{m}_c + \beta\boldsymbol{s} \tag{14}$$

where

$$\beta = \min\left\{ \frac{b_i - m_{c,i}}{s_i - m_{c,i}} \middle| \forall i : s_i < s_{\min,i} \vee s_i > s_{\max,i} \right\} \tag{15}$$

with b_i as the violated lower or upper bound. If the scenario is feasible, $\beta = 1$ holds. This way of handling box constraints yields a decrease in the Mahalanobis distance, since the length of the scenario is scaled. But this method may decrease the diversity within the offspring population since it can be seen as an additional factor applied to the mutation strength. Without further investigations it is not possible to decide which method is to be preferred.

Of course, other options for handling the box constraints, e.g. penalty-based approaches, reflection or lexicographic ordering, are possible. The two presented methods, however, lead to an interesting question: Is it worth to accept a worsening in one of the constraints (here the plausibility constraint) for a higher diversity of the population which is to be preferred in high-dimensional search spaces? This question can not be answered without considering how the plausibility constraint is handled.

As for the box constraints, several options exist for handling the plausibility constraint. For a survey of handling nonlinear (and linear) constraints one is referred to [29]. Again, the plausibility constraint can be evaluated independent of the balance sheet simulation. A straightforward approach is to apply a penalty approach. In such an approach, the fitness of an infeasible solution consists of the objective function value (which may be lower than the objective function value at the global optimum in the feasible search space) and a penalty value based on the constraint violation. The designed penalty function is

$$f^p = \max\left[(\tilde{\boldsymbol{s}} - \boldsymbol{m})^{\mathrm{T}}\boldsymbol{C}^{-1}(\tilde{\boldsymbol{s}} - \boldsymbol{m}) - \kappa, 0\right] . \tag{16}$$

The value of f^p is typically much larger than the results obtained by the simulation. This means, scenarios which violate the plausibility constraint in Eq. (2) have a low probability to be selected as parents if μ plausible scenarios exist. This consideration let to the choice of $\gamma = 1$ for line 14 in Algorithm 1.

A second option is to repair implausible scenarios. The approach is analogous to Eq. (14) with

$$\beta = \sqrt{\kappa\left[(\tilde{\boldsymbol{s}} - \boldsymbol{m})^{\mathrm{T}}\boldsymbol{C}^{-1}(\tilde{\boldsymbol{s}} - \boldsymbol{m})\right]^{-1}} . \tag{17}$$

Scenarios repaired by Eq. (17) will always have a plausibility of κ.

Next some small scale experiments are performed to investigate the different constraint handling options.

4 FIRST RESULTS

Before presenting some empirical results, the application specific parameters for the experiments are stated. In all experiments, the full set of risk parameters is considered and each risk parameter is forecasted for a time horizon of 55 months, yielding a search space dimensionality of $n = 17 \cdot 55 = 935$ and a plausibility threshold of

$$\kappa = 935 + \sqrt{2 \cdot 935} \approx 978.24.$$

The algorithm specific parameters of the ES are: $\lambda = 24$, $\mu = 12$, and a probability of $p = \frac{1}{12}$ for the determination of the initial mutation strength. The mean and the covariance matrix derived from the time series model are the same for all experiments. The transformations for the risk parameters are: log1p-transformation for 3 month euribor rate, passive side margins for the business sections private and treasury; log-transformation for all exchange rates, the Eurostoxx 50 index, and the iTraxx Main index; logit-transformation for all probabilities of default. All other risk parameter are not transformed. The various log transformations are given in Appendix A.

One final remark on the evaluation of the scenarios. The scenarios contain monthly values for each risk parameter. The simulation, however, only requires yearly values for each risk parameter. Thus, in an aggregation step the yearly averages are calculated and used as input for the simulation.

4.1 Investigation of design choices

For the investigation of the design choices for handling the constraints, 20 independent runs per design are performed. Each run has a budget of 1000 function evaluations. To reduce the variance in the runs, each of the 20 runs has a specific random seed and this seed is reused for the same run number for all other design choices. This means, the ith run for each design choice uses the same random seed and therefore the same sequence of random numbers. Neither the box constraint handling nor the repair of implausible scenarios use any random number. The design choices are:

- label "Puncture with repair": use of Eq. (14) for the box constraints and repair of implausible scenarios with Eq. (17)
- label "Puncture": use of Eq. (14) for the box constraints and using the penalty Eq. (16) for implausible scenarios
- label "Bounds with repair": use of Eq. (13) for the box constraints and repair of implausible scenarios with Eq. (17)
- label "Bounds": use of Eq. (13) for the box constraints and using the penalty Eq. (16) for implausible scenarios

In Figure 2 the average over the runs for the best-so-far fitness (top), the mutation strength (middle), and the plausibility of the current centroid $\boldsymbol{m}_c$ (bottom) are shown. What can be observed is that "Bounds with repair", i.e. setting infeasible scenarios parameters to the boundary of the feasible domain and repairing implausible scenarios, yields the lowest fitness[3] values (see top plot in Figure 2). Using the Wilcoxon-Mann-Whitney test [27, 40] one obtains the

[3]Since the problem is a minimization problem lower fitness values are considered better.

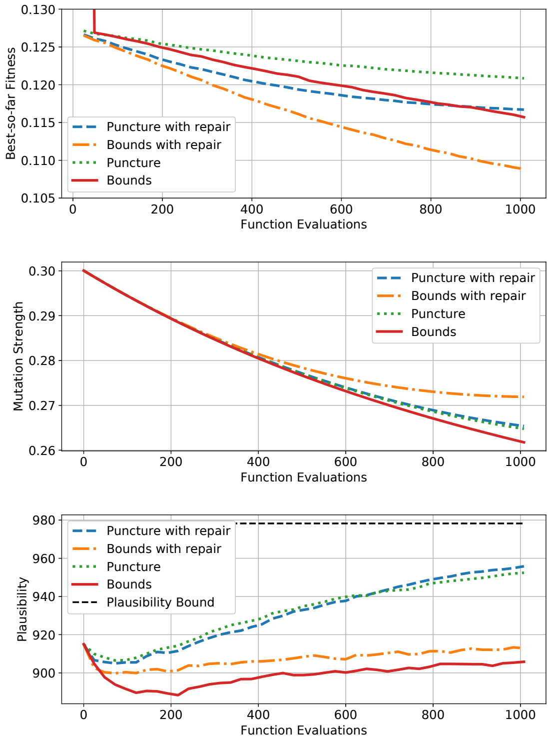

Figure 2: Curves for average values of best-so-far fitness (top), mutation strength (middle) and plausibility (bottom) over the number of function evaluations for the experiments investigating constraint handling choices.

following p-values for the one-sided test when comparing "Bounds with repair" with the other design choices:

"Bounds with repair" vs. "Puncture with repair":	$7.878 \cdot 10^{-7}$
"Bounds with repair" vs. "Puncture":	$3.398 \cdot 10^{-8}$
"Bounds with repair" vs. "Bounds":	$5.522 \cdot 10^{-6}$

If the same box constraint handling is used, but without repair, the best-so-far value, which pertains to the best offspring, at the beginning is implausible and thus receives a penalty as indicated by the respective line (solid line) leaving the top of the figure. Looking at the results for each of the runs (not shown here), one will find that in three of the 20 runs this happened.

Looking for reasons why this algorithm design performed best, one can take a look at the mutation strength (middle plot in Figure 2). As can be observed, the respective curves are quite similar in the beginning and notable differences start to occur at around 400 function evaluations. The algorithm design "Bounds with repair" achieves the largest mutation strength, which is desired for high-dimensional problems as in ESs the mutation strength is directly related to the population diversity.

Another option is to look at the plausibility of the population centroids (bottom plot in Figure 2). Note, the centroid is defined by $\boldsymbol{m}_c$ in Algorithm 1 and the centroid is never evaluated. Thus, a feasible and plausible centroid might produce infeasible and implausible offspring. If the plausibility of the centroid is further away from the plausibility threshold, the ES can operate with larger mutation strengths and produces (on average) less implausible offspring for the same mutation strength. As can be observed, the algorithm designs "Bounds with repair" and "Bounds" operate at lower centroid plausibilities.

Those considerations now enable one to answer the question posed in Section 3.3: Is it worth to accept a worsening in one of the constraints (here the plausibility constraint) for a higher diversity of the population which is to be preferred in high-dimensional search spaces? As it turns out, the effect on the plausibility is less than expected and the other box constrained handling approach (determination of the intersection points) has even a much more pronounced effect on the plausibility.

In the next experiment, longer runs of the best performing ES variant are investigated and the results are compared with results from a random search procedure.

4.2 Worst-Case Results

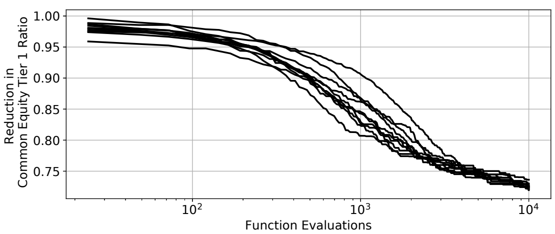

Figure 3: Sample runs for worst-case optimization with 10^4 function evaluations each.

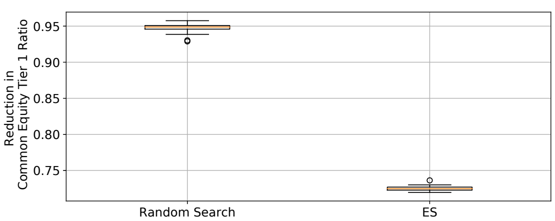

Figure 4: Comparison of Random Search and ES for the worst-case search.

In the following experiments the function evaluation budget is increased to 10^4 function evaluations. This time 10 independent runs of the ES with constraint handling "Bounds with repair" are performed. The results obtained are shown in Figure 3. The results show how much the Common Equity Tier 1 Ratio is decreased compared to its value in the reference scenario. The reference scenario is the scenario from which the remaining input parameters, i.e. the ones not in $\boldsymbol{s}$, for the balance sheet simulation are taken. The obtained reduction is around 27%.

To analyze the efficiency of the ES, a comparison with a simple random search procedure is performed. In the random search procedure, 100×10^3 scenarios are drawn from the multivariate normal distribution obtained from the time series. If a scenario is implausible it is repaired. If it is infeasible, variables violating the box constraints are set to the boundary of the feasible domain. In Figure 4, a box plot based on the best obtained results for each method is shown. As one can observe the ES is much more efficient, since random search only achieves between 5% to 7% reduction over the Common Equity Tier 1 Ratio of the reference scenario.

5 IMPLEMENTATION

An important, but often not stated part of applying an EA to a real-world problem, is how the potential users can interact with the EA and how the optimization process is embedded in the existing operational framework.

For the stress testing a framework for the evaluation of deterministic scenarios already exists and the worst-case search will be integrated there. The existing application is built in Python [17, 39] and uses the Jupyter framework [24, 38] for the interaction with the user. The plan is to build a graphical user interface which allows to control the important parameters of the optimization process. These parameters include the choice of risk parameters and their transformations and the maximum number of function evaluations. The number of offspring and parents can be taken from the proposed values in [20]

$$\lambda = 4 + \lfloor 3\ln(n) \rfloor,\ \mu = \frac{\lambda}{2}.$$

All other input parameters for the ES are either defined by the time series model or are set as stated in this work. This setup will allow users not familiar with ESs to perform the worst-case search.

Next to the control of the worst-case search, one needs to provide tools for visualizing and post-processing the results. In the framework several plots for the result and the respective statistics are provided. For example, the user can plot the values of the risk parameters of a worst case scenario and as defined by the reference scenario in the same plot. Moreover, tools like explanatory power analysis are available to gather more insights from the results.

Finally, the evaluation of a single scenario takes about 5 to 7 seconds on common hardware. To speed up the optimization, the evaluation of the scenarios is performed in parallel by using the Distributed Execution Framework [15]. For the experiments a simple master-worker approach [1] was used. The same framework exists at the bank and thus allows to perform a worst-case search in reasonable time.

6 CONCLUSION AND OUTLOOK

The paper proposes the use of an ES for finding worst case scenarios for the state of a bank. The derivation of the scenario distribution from historical time series is provided and the ES is outlined. Algorithmic design decisions for the ES are motivated and empirical results are provided.

The approach is not limited to stress testing a bank. Examples for different applications might come from the areas of project management or from engineering where, for example, different load scenarios have to be investigated. It is not claimed that the same approaches for handling the box constraints and the plausibility constraint will be the most efficient ones, but by using the same reasoning the most appropriate ones can be identified.

Several further investigations are possible. For example, what happens if the function evaluation budget is increased and whether some type of stagnation will be observed. The results presented here are not expected to be the global optimum[4] given the restriction on the function evaluation budget and the high-dimensional search space. Also, special approaches for dealing with the high-dimensional search spaces could be considered. One example would be to reduce the search space dimensionality by having the scenario vector defining the yearly forecast values instead of the monthly forecast values. Another interesting approach is to restrict the worst-case search on the surface of the hyper-ellipsoid defined by the plausibility constraint. Such an approach would require further adaptations to the ES. Finally, the use of more involved parallel approaches [1] could provide further speed-up of the optimization process and a more efficient search in the high-dimensional space.

Next to improving the worst-case search, adaptations of the problem might provide a deeper insight. A natural extension would be to use multi-criterion fitness function to capture different effects within the balance sheet. This would require the use of a multi-objective optimizer as worst-case search operator. Another interesting avenue would be to investigate different plausibility thresholds.

ACKNOWLEDGMENTS

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development is gratefully acknowledged. Sincere thanks also to Markus Seeger and Nora Nickel, both with the Hypo Vorarlberg Bank AG, who provided data, expertise, and were driving forces during the project development.

A DEFINITION OF THE LOG-TRANSFORMATIONS

$$\text{log-transformation: } \tilde{s_i} = \ln(s_i) \tag{18}$$

$$\text{log1p-transformation: } \tilde{s_i} = \ln(s_i + 1) \tag{19}$$

$$\text{logit-transformation: } \tilde{s_i} = \ln\left(\frac{s_i}{1 - s_i}\right) \tag{20}$$

REFERENCES

[1] E. Alba, G. Luque, and S. Nesmachnow. 2013. Parallel metaheuristics: recent advances and new trends. *International Transactions in Operational Research* 20, 1 (2013), 1–48. https://doi.org/10.1111/j.1475-3995.2012.00862.x

[2] European Banking Authority. 2018. *2018 EU-Wide Stress Test - Methodological Note.* Technical Report. https://eba.europa.eu/documents/10180/2106649/2018+EU-wide+stress+test+-+Methodological+Note.pdf

[3] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. 2009. *Robust Optimization.* Princeton University Press.

[4] H.-G. Beyer. 2001. *The Theory of Evolution Strategies.* Springer, Heidelberg. https://doi.org/10.1007/978-3-662-04378-3

[5] H.-G. Beyer and B. Sendhoff. 2007. Robust optimization — A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering* 196 (2007), 3190–3218. https://doi.org/10.1016/j.cma.2007.03.003

[6] R. Biedrzycki, J. Arabas, and D. Jagodziński. 2018. Bound constraints handling in Differential Evolution: An experimental study. *Swarm and Evolutionary Computation* (2018). https://doi.org/10.1016/j.swevo.2018.10.004

[4] It is not even clear whether a unique global optimum exists.

[7] J. R. Birge and P. Júdice. 2013. Long-term bank balance sheet management: Estimation and simulation of risk-factors. *Journal of Banking & Finance* 37, 12 (2013), 4711 – 4720. https://doi.org/10.1016/j.jbankfin.2013.07.040

[8] J. Branke and J. Rosenbusch. 2008. New Approaches to Coevolutionary Worst-Case Optimization. In *Parallel Problem Solving from Nature – PPSN X*, G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 144–153. https://doi.org/10.1007/978-3-540-87700-4_15

[9] R. P. Brent. 1973. *Algorithms for Minimization without Derivatives* (1st ed.). Prentice-Hall, Englewood Cliffs, New Jersey.

[10] T. Breuer and I. Csiszár. 2013. Systematic stress tests with entropic plausibility constraints. *Journal of Banking & Finance* 37, 5 (2013), 1552 – 1559. https://doi.org/10.1016/j.jbankfin.2012.04.013

[11] T. Breuer, M. Jandačka, K. Rheinberger, and M. Summer. 2009. How to Find Plausible, Severe and Useful Stress Scenarios. *International Journal of Central Banking* 5, 3 (September 2009), 205–224. https://www.ijcb.org/journal/ijcb09q3a7.pdf

[12] T. Breuer, K. Rheinberger, M. Jandačka, and M. Summer. 2008. Macro Stress and Worst Case Analysis of Loan Portfolios. (2008). https://www.bis.org/bcbs/events/rtf08bjrs.pdf

[13] T. Breuer and M. Summer. 2018. *Systematic Systemic Stress Tests*. Working Papers 225. Oesterreichische Nationalbank (Austrian Central Bank). https://ideas.repec.org/p/onb/oenbwp/225.html

[14] L. El Ghaoui, M. Oks, and F. Oustry. 2003. Worst-Case Value-At-Risk and Robust Portfolio Optimization: A Conic Programming Approach. *Operations Research* 51, 4 (2003), 543–556. https://doi.org/10.1287/opre.51.4.543.16101

[15] T. Feilhauer and M. Sobotka. 2016. DEF-a programming language agnostic framework and execution environment for the parallel execution of library routines. *Journal of Cloud Computing* 5, 1 (2016), 20. https://doi.org/10.1186/s13677-016-0070-z

[16] IFRS Foundation. 2019. International Financial Reporting Standards. (2019). Retrieved March 22, 2019 from https://www.ifrs.org/

[17] Python Software Foundation. 2019. Python 3.6.x Documentation. (2019). Retrieved March 22, 2019 from https://docs.python.org/3.6/

[18] N. Grubmann. 1987. Besmod: A strategic balance sheet simulation model. *European Journal of Operational Research* 30, 1 (1987), 30 – 34. https://doi.org/10.1016/0377-2217(87)90007-5 OR in Banking.

[19] N. Gülpinar and B. Rustem. 2007. Worst-case robust decisions for multi-period mean-variance portfolio optimization. *European Journal of Operational Research* 183, 3 (2007), 981 – 1000. https://doi.org/10.1016/j.ejor.2006.02.046

[20] N. Hansen. 2016. The CMA Evolution Strategy: A Tutorial. *CoRR* abs/1604.00772 (2016). http://arxiv.org/abs/1604.00772

[21] N. Hansen, D. V. Arnold, and A. Auger. 2015. *Evolution Strategies*. Springer Berlin Heidelberg, Berlin, Heidelberg, 871–898. https://doi.org/10.1007/978-3-662-43505-2_44

[22] N. Hansen and A. Ostermeier. 1996. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation*. 312–317. https://doi.org/10.1109/ICEC.1996.542381

[23] N. Hansen and A. Ostermeier. 2001. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9, 2 (2001), 159–195. https://doi.org/10.1162/106365601750190398

[24] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing. 2016. Jupyter Notebooks – a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, F. Loizides and B. Schmidt (Eds.). IOS Press, 87 – 90. https://doi.org/10.3233/978-1-61499-649-1-87

[25] R. Korn and O. Menkens. 2005. Worst-Case Scenario Portfolio Optimization: a New Stochastic Control Approach. *Mathematical Methods of Operations Research* 62, 1 (01 Sep 2005), 123–140. https://doi.org/10.1007/s00186-005-0444-3

[26] H. Lütkepohl. 2005. *New Introduction to Multiple Time Series Analysis*. Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-540-27752-1

[27] H. B. Mann and D. R. Whitney. 1947. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Ann. Math. Statist.* 18, 1 (03 1947), 50–60. https://doi.org/10.1214/aoms/1177730491

[28] A. Marcelo, A. Rodríguez, and C. Trucharte. 2008. Stress tests and their contribution to financial stability. *Journal of Banking Regulation* 9, 2 (01 Feb 2008), 65–81. https://doi.org/10.1057/jbr.2008.1

[29] E. Mezura-Montes and C. A. Coello Coello. 2011. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation* 1, 4 (2011), 173 – 194. https://doi.org/10.1016/j.swevo.2011.10.001

[30] F. Pistovčák and T. Breuer. 2004. Using Quasi-Monte Carlo Scenarios in Risk Management. In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, H. Niederreiter (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 379–392. https://doi.org/10.1007/978-3-642-18743-8_24

[31] M. Quagliariello. 2009. *Stress-testing the Banking System: Methodologies and Applications*. Cambridge University Press.

[32] I. Rechenberg. 1973. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart. https://doi.org/10.1002/fedr.19750860506

[33] H. Schierenbeck. 2013. *Ertragsorientiertes Bankmanagement: Band 1: Grundlagen, Marktzinsmethode und Rentabilitäts-Controlling* (6. ed.). Gabler Verlag.

[34] H. Schierenbeck, M. Lister, and S. Kirmße. 2010. *Ertragsorientiertes Bankmanagement: Band 2: Risiko-Controlling und integrierte Rendite-/Risikosteuerung* (9. ed.). Gabler Verlag.

[35] T. Schuermann. 2014. Stress testing banks. *International Journal of Forecasting* 30, 3 (2014), 717 – 728. https://doi.org/10.1016/j.ijforecast.2013.10.003

[36] H.-P. Schwefel. 1977. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*. Birkhäuser, Basel. https://doi.org/10.1007/978-3-0348-5927-1

[37] G. Studer. 1997. *Maximum loss for measurement of market risk*. Ph.D. Dissertation. ETH Zurich, Zurich, Switzerland. https://doi.org/10.3929/ethz-a-001891697

[38] Jupyter Team. 2019. Jupyter. (2019). Retrieved March 22, 2019 from https://jupyter.readthedocs.io/en/latest/index.html

[39] G. van Rossum. 1995. *Python tutorial*. Technical Report CS-R9526. Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

[40] F. Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* 1, 6 (1945), 80–83. https://doi.org/10.2307/3001968

[41] S. Zhu and M. Fukushima. 2009. Worst-Case Conditional Value-at-Risk with Application to Robust Portfolio Management. *Operations Research* 57, 5 (2009), 1155–1168. https://doi.org/10.1287/opre.1080.0684