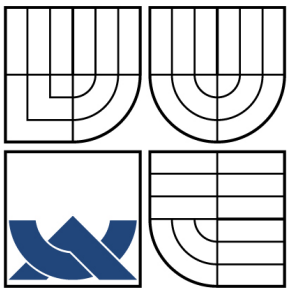
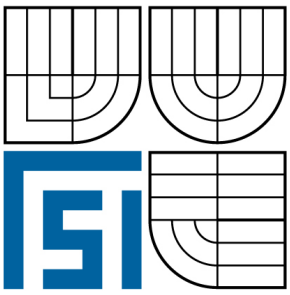


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

APPROXIMATIONS IN STOCHASTIC OPTIMIZATION AND THEIR APPLICATIONS

APROXIMACE VE STOCHASTICKÉ OPTIMALIZACI A JEJICH APLIKACE

DIZERTAČNÍ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE — Ing. Mgr. EVA ŽAMPACHOVÁ
AUTHOR

VEDOUCÍ PRÁCE — doc. RNDr. ZDENĚK KARPÍŠEK, CSc.
SUPERVISOR

BRNO 2009

Abstract

Many optimum design problems in engineering areas lead to optimization models constrained by ordinary (ODE) or partial (PDE) differential equations, and furthermore, several elements of the problems may be uncertain in practice. Three engineering problems concerning the optimization of vibrations and an optimal design of beam dimensions are considered. The uncertainty in the form of random load or random Young's modulus is involved. It is shown that two-stage stochastic programming offers a promising approach in solving such problems. Corresponding mathematical models involving ODE or PDE type constraints, uncertain parameters and multiple criteria are formulated and lead to (multi-objective) stochastic nonlinear optimization models. It is also proved for which type of problems stochastic programming approach (EO reformulation) should be used and when it is sufficient to solve simpler deterministic problem (EV reformulation). This fact has the big importance in practice in term of computational intensity of large scale problems.

Computational schemes for this type of problems are proposed, including discretization methods for random elements and ODE or PDE constraints. By means of derived approximations the mathematical models are implemented and solved in GAMS. The solution quality is determined by an interval estimate of the optimality gap computed via Monte Carlo bounding technique. Parametric analysis of multi-criteria model results in efficient frontier computation. The alternatives of approximations of the model with reliability-related probabilistic terms including mixed-integer nonlinear programming and penalty reformulations are discussed. Furthermore, the progressive hedging algorithm is implemented and tested for the selected problems with respect to future possibilities of parallel computing of large engineering problems. The results show that it can be used even when the mathematical conditions for convergence are not fulfilled. Finite difference method and finite element method are compared for deterministic version of ODE constrained problem by using GAMS and ANSYS with quite comparable results.

The Ph. D. thesis has been written as the part of the project from MSMT of the Czech Republic no. 1M06047 Center for Quality and Reliability of Production and the achieved results will be also applied to the problems from the Czech Science Foundation project reg. no. 103/08/1658 "Advanced optimum design of composed concrete structures" and the research plan from MSMT of the Czech Republic no. MSM0021630519 "Progressive reliable and durable structures". It has been also supported by two research grants from Science fund in FME BUT.

Keywords: optimum engineering design, ODE and PDE constraints, stochastic programming, chance constrained programming, multi-objective programming, Monte Carlo method, progressive hedging algorithm

ŽAMPACHOVÁ, E. Approximations in stochastic optimization and their applications. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 95 s. Vedoucí dizertační práce doc. RNDr. Zdeněk Karpíšek, CSc.

Abstrakt

Mnoho inženýrských úloh vede na optimalizační modely s omezeními ve tvaru obyčejných (ODR) nebo parciálních (PDR) diferenciálních rovnic, přičemž jsou v praxi často některé parametry neurčité. V práci jsou uvažovány tři inženýrské problémy týkající se optimalizace vibrací a optimálního návrhu rozměrů nosníku. Neurčitost je v nich zahrnuta ve formě náhodného zatížení nebo náhodného Youngova modulu. Je zde ukázáno, že dvoustupňové stochastické programování nabízí slibný přístup k řešení úloh daného typu. Odpovídající matematické modely, zahrnující ODR nebo PDR omezení, neurčité parametry a více kritérií, vedou na (vícekriteriální) stochastické nelineární optimalizační modely. Dále je dokázáno, pro jaký typ úloh je nutné použít stochastické programování (EO reformulace), a kdy naopak stačí řešit jednodušší deterministickou úlohu (EV reformulace), což má v praxi význam z hlediska výpočetní náročnosti.

Jsou navržena výpočetní schémata zahrnující diskretizační metody pro náhodné proměnné a ODR nebo PDR omezení. Matematické modely odvozené pomocí těchto aproximací jsou implementovány a řešeny v softwaru GAMS. Kvalita řešení je určena na základě intervalových odhadů "optimality gapu" spočtených pomocí metody Monte Carlo. Parametrická analýza vícekriteriálního modelu vede na výpočet "efficient frontier". Jsou studovány možnosti aproximace modelu zahrnujícího pravděpodobnostní členy související se spolehlivostí pomocí smíšeného celočíselného nelineárního programování a reformulace pomocí penalizační funkce. Dále je vzhledem k budoucím možnostem paralelních výpočtů rozsáhlých inženýrských úloh implementován a testován PHA algoritmus. Výsledky ukazují, že lze tento algoritmus použít, i když nejsou splněny matematické podmínky zaručující konvergenci. Na závěr je pro deterministickou verzi jedné z úloh porovnána metoda konečných diferencí s metodou konečných prvků za použití softwarů GAMS a ANSYS se zcela srovnatelnými výsledky.

Dizertační práce je součástí řešení projektu MŠMT ČR č. 1M06047 Centrum pro jakost a spolehlivost výroby a dosažené výsledky budou také aplikovány na problémy projektu GAČR reg. č. 103/08/1658 "Pokročilé optimalizace návrhu složených betonových konstrukcí" a výzkumného záměru MŠMT ČR č. MSM0021630519 "Progresivní spolehlivé a trvanlivé nosné stavební konstrukce". Práce byla také podpořena dvěma granty Fondu vědy FSI VUT v Brně.

Klíčová slova: optimální inženýrský návrh, ODR a PDR omezení, stochastické programování, optimalizace s pravděpodobnostními omezeními, vícekriteriální optimalizace, metoda Monte Carlo, PHA algoritmus

Affirmation

I affirm that I have written this thesis independently, following the supervisor's instructions and with the use of the presented references.

18th December 2009 Eva Žampachová

Acknowledgement

I thank my supervisor doc. RNDr. Zdeněk Karpíšek, CSc., for his support, advice and providing the most pleasant work environment.

I owe a great deal to my supervisor – specialist RNDr. Pavel Popela, Ph. D., for his constant motivation, invaluable advice and helpful ideas that proved useful in many situations. For that, I thank him deeply.

Furthermore, I would like to express my gratitude to Michal Čajánek and Libor Borák for fruitful discussions and suggestions. I also address thanks to colleagues from research seminars at the Dept. of Statistics and Optimization, IM, FME, BUT Brno, and the Dept. of Probability and Mathematical Statistics, FMP, CU Prague, for their fruitful comments and discussions.

My special thanks belong to my parents for their permanent support, love and understanding. Not least, I thank Michal for his love, patience, everlasting support and help with figures.

Contents

Chapter 1

Motivation

Many deterministic problems from a wide range of applications are often governed by ordinary (ODE) or partial (PDE) differential equation, i. e. a relationship including some continuously varying quantities (modelled by functions) and their rates of change in space and/or time is known or postulated. Their solutions are then described in the form of the function of one or more variables (e. g. designs of mechanical structures), see e. g. [44]. With the development of the computational methods we insist on obtaining not only a possible but also an optimal design of structure, see [3] and [28]. In many problems that are solved presently, it is necessary to combine these two facts. Therefore, we are often facing ODE or PDE constrained optimization problems [5]. Due to optimization, certain ODE's/PDE's elements can be viewed as the control variables, and after introducing appropriate objective function, we are able to obtain an optimal control problem. Theory of optimal control problems is very well developed for constraints in the form of ordinary differential equations [30]. However we would like to deal also with PDE constrained mathematical programming problems and due to this fact we have to challenge several difficulties.

As we can find a closed-form solution only for simple ODEs or PDEs, we have to approximate their solution by discretization in most cases. Classical approximation techniques for solving these equations include finite difference method, finite volume method and finite element method (e. g. [1], [54] and [72]). Then we can approximate our initial ODE/PDE constrained optimization problem by mathematical programs [62]. Solution methods of these deterministic problems are relatively well developed.

However, in practice, several elements of the problem, e. g. loads, are not very often given as fixed quantities but rather as random elements, therefore a problem with uncertainty results from the above mentioned deterministic problem. For example reliability check of the designed structures involving randomness is computationally intensive even with the use of simulation methods [32].

The combination of the two previously mentioned areas results in ODE/PDE constrained optimization problem with uncertainty. We can obtain two types of formulation of this problem, therefore, we need to explain the difference between stochastic programming and stochastic optimal control. The concept of the solution is slightly different in these two approaches. In stochastic programming, strictly speaking in multistage recourse model, we are especially concerned with the first-stage decision while the next stages are only included to help evaluate the costs that may result from a particular choice of the first-stage decision. On the other hand, in stochastic optimal control problems, we want to design "feedback control" (i. e. a function that will map the available information into an optimal decision) [70]. Furthermore, the multistage stochastic programming model

always implies a discrete time modelling of the underlying dynamics, whereas stochastic optimal control models allow for both continuous and discrete time modelling of dynamics. In our studied problems, the stage-related decision structure – as is in stochastic programming – is more adequate than continuous time like in optimal control. Moreover, as it is often necessary to make decision before a realization of the corresponding random variables becomes known, and after it, two-stage stochastic programming can be used.

For these reasons, we are facing ODE or PDE constrained stochastic programming models. Solution methods are not developed so much for these cases (e. g. one group in Germany [10] and in the Czech Republic [19] is concerned with the shape optimization problem constrained by PDE), and therefore, it could be very useful to be concerned with this type of problems, formulate them properly and propose suitable approximation and solution methods that can be further used in various application areas.

Chapter 2

The aims of the Ph. D. thesis

The aims of the Ph. D. thesis can be divided into the following points:

1. We will formulate PDE constrained stochastic programming problem as a theoretical model for selected engineering problems. We will propose a computational scheme for solving optimization problems containing PDE constraints and uncertainty using the two-stage scenario-based stochastic programming and suitable discretization method. Furthermore, theoretical properties of the model will be investigated. Deterministic and stochastic approaches will be compared by means of selected characteristics (VSS, etc.).

2. We will be concerned with Monte Carlo bounding techniques for determining solution quality in these stochastic programs. These techniques have been especially used for linear stochastic programs with complete recourse but they have not been applied to PDE constrained nonlinear stochastic programs.

3. We will implement and test the progressive hedging algorithm for our class of problems and find out its advantages and limitations. Furthermore, we will theoretically verify the convergence of this algorithm and develop and test heuristics for its acceleration while convergence will be conserved.

4. We will discuss the alternatives of approximations of the model that includes reliability-related probabilistic terms by means of closely studied two-stage stochastic programs.

5. We will make a conclusion about the importance and contribution of stochastic programming approach to decision making under uncertainty especially in the engineering problems.

Chapter 3

Optimization in engineering design

Optimization is undoubtedly indispensable in all engineering areas. We are especially interested in optimum design problems leading to optimization models constrained by differential equations. Therefore, we will provide brief review of different engineering areas where this type of problems can arise.

Civil engineering is an engineering discipline that deals with the design, construction and maintenance of the physical and naturally built environment, including works such as bridges, roads, canals, dams and buildings [71]. Our interest is mainly in structural engineering concerning structural design and structural analysis of buildings, bridges, towers, tunnels and other structures. Design considerations include strength, stiffness, and stability of the structure when subjected to static (self-weight) or dynamic (wind, seismic etc.) loads acting upon a structure. Other considerations include cost, safety, sustainability and nowadays also environmental impact, see e. g. [29]. Therefore, multi-objective optimization is often used (see chapter 6). Furthermore, uncertainty plays an important role in many real-world engineering problems, e. g. in optimum design of structures due to variations of the material, variations of the external loads etc. Therefore, the parameters of a structure are not given fixed coefficients, but they can be modelled by random variables with certain probability distributions (unfortunately, these distributions are not always available and must be estimated). There are also increasing requests from practitioners to reconsider traditional "expert-based" deterministic models and find more realistic risk-averse models with probabilistic nature. These requirements and necessity to solve complex problems require the development of new ideas, innovative methods and numerical tools for providing reasonable solutions in affordable computing times.

In case of deterministic-based structural optimization (DBSO), all uncertainties at determining loads, material characteristics etc., are neglected, see e. g. [57]. But as we mentioned above, real-world design problems include the uncertain parameters that can be modelled by random elements as in the case of reliability-based structural optimization (RBSO) [43]. Basic RBSO algorithm features and numerical comparisons between RBSO and DBSO algorithms have been studied for the optimization of a reinforced concrete cross-section subjected to the combinations of normal force and bending moments in [59].

Stochastic programming provides another approach. Stochastic programming models for reliable design in civil engineering, especially, for optimization of concrete cross-sections are presented e. g. in [60]. These structural optimization models involving random elements are complex in comparison with the available computational power, and therefore, model reformulations and approximating models are derived.

Another possibility of solving such problems is using stochastic finite element method (SFEM). The fundamental task in this method is simulation of random fields that is

very difficult especially for computationally intensive problems (typically nonlinear FEM calculations) [65]. SFEM has been used e. g. for optimum design of longitudinal reinforcement in a concrete framed structure [58]. Behaviour of the loaded structure is modelled by FEM method considering non-linearity arising from the stress-strain diagrams of the individual materials. Uncertainty is represented by stiffness parameters (dimension of cross-section, position of reinforcement layer, modulus of elasticity and strength of concrete and reinforcing steel) and the external load that are considered as random variables.

Reliability (safety, durability, serviceability) assessment is also very important problem in structural engineering. Computer technology advancements have enabled to use more efficient probabilistic methods and more intensive evaluation of structural reliability. Therefore, the mostly used deterministic "way of thinking" has been replaced by the probabilistic approach. One of the probabilistic assessment methods, using simulation technique, Monte Carlo method and analysis of empirical distribution, is simulation-based reliability assessment method (SBRA) [18], [32]. Another probabilistic simulation-based method, where the distribution of random variables is approximated by a theoretical distribution and its parameters are estimated by Monte Carlo method, is e. g. Latin hypercube sampling. This approach along with nonlinear finite element analysis is used in [39] to calculate the safety index of an concrete structure characterizing its reliability.

Mechanical engineering involves the analysis, design, manufacturing, and maintenance of various systems. It is divided into several sub-disciplines including mechanics, thermodynamics, mechatronics and robotics, structural analysis, biomechanics etc.

An example from thermodynamics is concerned in optimization and control of production of steel slabs on real casters with the aim of achieving the maximum possible savings and product quality [27], [36] and [61]. This problem of continuous casting process of steel involving uncertain parameters can be formulated as a stochastic PDE constrained optimization model and is solved via mathematical programming [35].

Biomechanics is an multidisciplinary research area associated with many not only engineering fields as mechanics, thermodynamics, biology, chemistry, medicine etc. There are many examples where structural optimization methods can be used in connection to biomechanics [22]. One of them is optimal design of implants interacting with tissues at macro and micro level, i. e. problems of shape and topology optimization associated with solid/fluid mechanics, acoustics, problems of interfaces between implants and tissues and their reliability, optimal design of composite elements and others. Another problem is optimal design of medical devices and instruments interacting with human body, e. g. external and internal fixators [37].

Process (chemical) engineering focuses on the design, operation and maintenance of chemical and other process manufacturing activities [71]. Its aim is the design of processes for desired physical and/or chemical transformation of materials and the design of new facilities or the modification or expansion of existing facilities. One example of using shape-based optimization in this field is experimental burner design [56]. Constraints in the form of differential equations arise from computational fluid dynamics and the aim is to find an optimal geometrical arrangement of the secondary fuel nozzles with respect to the global NOx production.

Aeronautical engineering deals with design, construction and science of aircraft. The goal of all aerodynamic design methods is to find a shape which improves an aerodynamic measure of merit while satisfying the appropriate constrains.

The formulation of optimization techniques based on control theory for aerodynamic shape design in viscous compressible flow, modelled by the Navier-Stokes equations, is described in [23]. The theory is applied to a system defined by the partial differential

equations of the flow, with the boundary shape acting as the control. The method is illustrated by designs of wings and wing-body combinations for long range transport aircraft. The aerodynamic optimization of single and multi-element airfoil configurations is presented in [38]. Design examples include lift-enhancement and multi-point lift-constrained drag minimization problems. The flow is governed by the compressible Navier-Stokes equations mentioned above in conjunction with a one-equation turbulence model. Optimization constraints are enforced through a penalty formulation and the resulting unconstrained problem is solved via a quasi-Newton method.

This type of engineering optimization problems has rather interdisciplinary character. One has to understand the physical background to formulate the problem correctly from the physical point of view. Then an appropriate mathematical model for the numerical realization has to be found. Several mathematical branches interfere here: the theory of partial or ordinary differential equations, approximation of these equations, the theory of nonlinear mathematical programming and the theory of stochastic programming in case of involved uncertainty.

There are more and less advanced techniques how to deal with these problems. More advanced methods have been developed for deterministic cases and involve more sophisticated approximation techniques of differential equations such as finite element method and also precise mathematical description of convergence and optimality condition, e. g. [19].

We will focus on optimization problems constrained by differential equations (ordinary or partial) and involving uncertainty. This type of problems can be seen from stochastic optimal control point of view [17]. However, there are several bottlenecks in this approach. There are not enough input data to obtain realistic instances of models. Solution techniques often significantly vary even for small model changes while real-world engineering problems require robust approaches. Furthermore, optimal control models are very difficult to solve especially when a dynamic system is strongly nonlinear and there are constraints on states and control [11]. Classical methods for studying optimal controls are based on Pontryagin's maximum principle, Bellman's principle of optimality or Hamilton-Jacobi-Bellman equation [30]. Unfortunately, analytical solutions cannot be found for every optimal control problem and therefore one needs to study their theoretical properties and appropriate computational methods to find at least approximate control solutions [68].

But we will follow another way. It is due to that engineers prefer widely applicable robust algorithmic schemes instead of efficient algorithms for specialised cases. Furthermore, they do not necessarily need optimal solutions and they are often satisfied enough with a significant improvement of the existing design, i. e. suboptimal solutions. Therefore, the model-based approximation where its quality is verified by the comparison of existing and suboptimal solutions is further chosen and development of the computational scheme is illustrated by several fundamental engineering examples.

Chapter 4

Mathematical programming and multi-objective optimization

In this chapter, we will briefly discuss some basic concepts from optimization theories, needed for the development of stochastic programming models. Furthermore, we will summarize principles and solution methods of multi-objective optimization.

4.1 Mathematical programming

Mathematical programs play an important role in stochastic programming because the underlying programs (5.1) are obtained from them and after deterministic reformulation we must usually solve corresponding mathematical programs. Therefore, we provide brief review of basic concepts which will be needed for formulation of stochastic programs.

Definition 4.1.1. A *mathematical programming problem* is defined as

$$\min_{\mathbf{x}} f(\mathbf{x}) \text{ s. t. } \mathbf{x} \in C, \tag{4.1}$$

where $\mathbf{x}$ is a decision variable, $C = \{\mathbf{x}|\mathbf{g}(\mathbf{x}) \diamond \mathbf{0}, \mathbf{x} \in X\}$ is a feasible set, $X \subseteq \mathbb{R}^n$, $\mathbf{g} : \mathbb{R}^{\mathbf{n}} \to \mathbb{R}^{\mathbf{m}}$ is a vector function, symbol $\diamond \in \{\leq, \geq, =\}$ expresses relations and $f : C \to \mathbb{R}$ is an objective function.

Depending on the properties of the functions f, g_i and the set X, program (4.1) is denoted as (see [26])

- *linear* for the convex polyhedral set[1] X and the linear functions f, g_i, $i = 1, \ldots, m$,
- *nonlinear* for the set X which is not a convex polyhedron or for at least one nonlinear function of f, g_i, $i = 1, \ldots, m$,
  - *convex* for the convex feasible set C and the convex objective function f,
  - *nonconvex* for the nonconvex feasible set C or the nonconvex objective function f,
- *(mixed) integer* for (at least some of) the integer decision variables x_j, $j = 1, \ldots, n$.

[1] A polyhedral set (polyhedron) is the intersection of a finite number of closed halfspaces and is always convex.

Convexity plays an important role because for a convex program any local minimum is also a global minimum. Because of the straight relation of mathematical programs to the underlying programs in stochastic programming we define parametric mathematical program.

Definition 4.1.2. A *parametric mathematical program* is defined as

$$\min_{\mathbf{x}} f(\mathbf{x}, \mathbf{a}) \text{ s. t. } \mathbf{x} \in C(\mathbf{a}), \tag{4.2}$$

where $\mathbf{a} \in \mathbb{R}^k$ is a constant parameter and $C(\mathbf{a}) = \{\mathbf{x} | \mathbf{g}(\mathbf{x}, \mathbf{a}) \diamond \mathbf{0},\ \mathbf{x} \in X\}$.

There are many theoretical concepts and results in mathematical programming theory and many methods for solving different mathematical programming problems. But their summary goes beyond the scope of this thesis and can be found in any standard textbook [3].

4.2 Multi-objective optimization

Notion of an optimal solution may become meaningless when an optimization model has more than one objective function. A solution that is the best by one criterion may be the worst on another one. Therefore the concepts of efficient points and efficient frontier have been developed (see e. g. [15] and [55]).

Definition 4.2.1. A feasible solution to a multi-objective optimization model is an *efficient point* if no other feasible solution scores at least as well in all objective functions and strictly better in one.

This point is also called *Pareto optimal* or *nondominated point*. It helps to characterize the "best" feasible solutions in multi-objective models.

Multi-objective optimization model can have many efficient points, therefore, we need to consider a range of efficient solutions.

Definition 4.2.2. An *efficient frontier* of a multi-objective optimization model is the entire set of efficient points for the model.

The set of points on the efficient frontier can be constructed by repeated optimization. New constraints enforcing achievement levels for all but one criterion are added and the remaining criterion is treated as a single-objective function.

Regarding the graphical representation, solutions are plotted in objective value space with axes for different objective functions instead of axes corresponding to the decision variables. The efficient frontier forms the boundary of the region given by the objective values for feasible points. Efficient points graph at the points along this boundary while dominated points transform into points lying in the interior. Infeasible points lie outside this region.

Range of the efficient solutions can be really large in real-life models, and therefore, the explicit construction of the efficient frontier may become computationally very intensive. Consequently, other methods which reduce multi-objective models to (sequence) of single-objective models have been developed.

The first method is the *preemptive* optimization [49] which takes the objective functions in priority order with respect to their importance. The most important criterion is taken as a single-objective function and the model is optimized. Then the second most

important criterion is optimized subject to the requirement that the first criterion achieves its optimal value and so on.

Another way how to deal with multi-objective models is to combine objectives in a *weighted sum*. This approach provides more balanced handling of the objectives than the preemptive optimization. Signs of weights orient all objectives in the same direction while the weights correspond to their relative importance. This approach also assures that an optimal solution of a single weighted-sum objective is an efficient point of the multi-objective model.

The *goal programming* [55] is the another method. Goal (target) levels for every criterion have to be specified and a weighted-sum of the deficiency variables[2] is minimized. This objective function expresses the effort to satisfy all targets as nearly as possible.

More precise descriptions of these methods and their theoretical properties can be found in many textbooks devoted to multi-objective optimization, e. g. in [49] and [55].

[2]These nonnegative variables model the measure of violation in target or other soft constraints which may be violated in feasible solutions.

Chapter 5

Stochastic programming

This chapter will provide review of stochastic programming concepts which will be used to solve selected engineering optimization problems in the following chapters.

5.1 Probabilistic concepts

Stochastic programming deals with randomness therefore we need to introduce some basic concepts of probability theory [16].

Definition 5.1.1. $(\Omega, \mathcal{F}, \mathcal{P})$ is called a *probability space* if

- Ω is a nonempty set representing the possibilities open
- $\mathcal{F}$ is a σ-algebra of subsets of Ω called events, i. e. this collection of subsets satisfies the following conditions:
  1. $\mathcal{F} \neq \emptyset$
  2. $A \in \mathcal{F} \Rightarrow \bar{A} \in \mathcal{F}$
  3. $A_i \in \mathcal{F},\ i = 1, 2, \ldots \Rightarrow \bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$
- P is a probability measure on $(\Omega, \mathcal{F})$, i. e. a nonnegative mapping $P : \mathcal{F} \to \langle 0, 1 \rangle$ such that
  1. $P(\Omega) = 1$
  2. $A_i \in \mathcal{F},\ i = 1, 2, \ldots,\ A_i \cap A_j = \emptyset,\ i \neq j \Rightarrow P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$

Definition 5.1.2. It is said that an event $A \in \mathcal{F}$ happens *P-almost surely* (a. s.) or *almost everywhere* (a. e.) if $P(A) = 1$.

Definition 5.1.3. The function $\xi : \Omega \to \mathbb{R}$ is said to be a *random variable* if its inverse image $\xi^{-1}(A) = \{\omega \in \Omega : \xi(\omega) \in A\} \in \mathcal{F}$ for any Borel set $A \in \mathcal{B}$ [16].

The mapping ξ generates the probability distribution $\mathbb{P}(A) = P(\xi^{-1}(A))$ on $(\mathbb{R}, \mathcal{B})$. This probability distribution is completely defined by a *cumulative distribution function* $F(x) = \mathbb{P}\{\omega \in \Omega : \xi(\omega) \leq x\}$ defined $\forall x \in \mathbb{R}$.

Definition 5.1.4. A *mean value*, *expected value* or *expectation* $\mathbb{E}(\xi)$ of a random variable $\xi : \Omega \to \mathbb{R}$ is defined by the integral $\mathbb{E}(\xi) = \int_{\Omega} \xi(\omega) \mathrm{d}P(\omega)$, where the integral converges absolutely.
The expectation for discrete distributed random variable ξ, i. e. ξ can take only countable number of different values ξ_i, is given by $\mathbb{E}(\xi) = \sum_i \xi_i p_i$, where $p_i = \mathbb{P}\{\xi = \xi_i\}$ and the series converges absolutely.

Definition 5.1.5. A *variance* of an random variable $\xi : \Omega \to \mathbb{R}$ is defined as the second central moment: $\mathrm{var}(\xi) = \mathbb{E}(\xi - \mathbb{E}(\xi))^2$.

Further important definitions and concepts from the probability theory can be found for example in the classical textbook of stochastic programming [25], pp. 6–10.

5.2 Stochastic programming

Uncertainty is the key ingredient in many decision problems. Not much is lost by assigning reasonable values to the unknown elements, as long as their role is relatively insignificant. But there are many situations in which ignoring uncertainty may lead to totally misleading solutions. Therefore, unsatisfactory results of many deterministic optimization models gave rise to various approaches to optimization under uncertainty which incorporate uncertainty.

Stochastic programming (SP) uses approach based on probabilistic models of uncertainty. The objective functions and the constraints of the corresponding mathematical programming model can then be defined by averaging possible outcomes or considering probabilities of events of interest. This approach is the appropriate modelling tool when a probabilistic description of the unknown elements is at hand. The first important SP application was originated by Dantzig in 1955 [12].

The first step to obtain stochastic program is the formulation of the underlying program [45]. This can be done easily via parametric mathematical program by replacing some constant parameters in (4.2) by random variables.

Definition 5.2.1. An *underlying program* is defined as

$$\min_{\mathbf{x}} f(\mathbf{x}, \omega) \text{ s. t. } \mathbf{x} \in C(\omega), \tag{5.1}$$

where $\omega \in \Omega$ is an random element. Usually the involved random data is formed by a finite numbers of parameters. Therefore, the objective function is given as $f(\mathbf{x}, \omega) = F(\mathbf{x}, \boldsymbol{\xi}(\omega))$, where $\boldsymbol{\xi}(\omega) : \Omega \to \mathbb{R}^K$ is a finite dimensional random vector defined on probability space $(\Omega, \mathcal{F}, P)$ and $F(\mathbf{x}, \boldsymbol{\xi})$ is a function of two vector variables $\mathbf{x}$ and $\boldsymbol{\xi}$. Realization (*scenario*) of $\boldsymbol{\xi}$ is $\boldsymbol{\xi}(\omega_s)$ $\forall \omega_s \in \Omega$ and we will use the following notation $\boldsymbol{\xi}_s$.

It can be easily seen that the above mentioned mathematical programming problem depends on ω due to the objective function $f(\mathbf{x}, \omega)$ and the feasible set $C(\omega)$, and therefore, it is meaningless if the realization of the random parameters is not observed. It is clear that for different realizations $\boldsymbol{\xi}_s$ of the random parameters we would obtain different optimal solutions without any insight which one is better than the others. For this reason we use various *deterministic reformulations (equivalents)* that correctly interpret random elements. Programs that involve random parameters in syntactically correct ways are called *stochastic programs*.

The first question that must be answered is when the decision $\mathbf{x}$ will be made. Whether before the random parameters $\boldsymbol{\xi}$ are observed or after the realizations $\boldsymbol{\xi}_s$ become known. When the decision $\mathbf{x}$ is made after observing the randomness $\boldsymbol{\xi}$, this case is called the *wait-and-see* (WS) approach (e. g. [26]). This approach assumes the perfect information about the future. In this case, we can modify our decision by observation, and hence, the decision $\mathbf{x}(\omega)$ as well as the objective function $F(\mathbf{x}(\omega), \boldsymbol{\xi}(\omega))$ are random variables. But the typical decision situation is described by the lack of observations, therefore, we usually use the so-called *here-and-now* (HN) approach in stochastic programming. It means that the decision $\mathbf{x}$ must be made before the observations of $\boldsymbol{\xi}$ are known, and therefore, it is the same for any future realization of $\boldsymbol{\xi}$.

The mapping $\boldsymbol{\xi}$ induces a probability distribution $\mathbb{P}$ on $(\mathbb{R}^K, \mathcal{B})$ and we denote the corresponding probability space as $(\Xi, \mathcal{B}, \mathbb{P})$. In fact $\mathbf{x}$ in WS approach is a function of $\boldsymbol{\xi}$ and we can view $\mathbf{x}(\cdot) : \Xi \to \mathbb{R}^n$ as an element of a space of measurable mappings. Therefore, we will further use this notation to emphasize dependence on $\boldsymbol{\xi}$.

Stochastic programs are often computationally intensive, and therefore, people prefer to solve simpler versions such as EV reformulation introduced in the definition 5.2.4. But the question is whether these simplifications lead to nearly optimal solutions or not. It is answered by two values – the value of stochastic solution (see definition 5.2.7) and the expected value of perfect information (see definition 5.2.8) – which actually give the motivation for using stochastic programming in practice.

Onward, we denote the optimal objective function values for any deterministic reformulation $\bullet$ as $z^\bullet$ and optimal decision as $\mathbf{x}^\bullet$. We also assume that the expected value is taken with respect to a known probability distribution and that $\mathbb{E}(\boldsymbol{\xi})$ and $\mathbb{E}(F(\mathbf{x}, \boldsymbol{\xi}))$ exist and are well defined.

The best possible solution of the underlying program (5.1) in case if perfect information about the future is available, is provided by the so-called wait-and-see solution or equivalently by solving the distribution problem[1] [6]. This approach assumes that we are somehow able to wait until the uncertainty is resolved before implementing the optimal decisions.

Definition 5.2.2. *Wait-and-see (WS) deterministic reformulation* of the underlying program (5.1) is defined as

$$\mathbb{E}(F(\mathbf{x}^{WS}(\boldsymbol{\xi}), \boldsymbol{\xi})), \tag{5.2}$$

where

$$F(\mathbf{x}^{WS}(\boldsymbol{\xi}), \boldsymbol{\xi}) = \min_{\mathbf{x}(\boldsymbol{\xi})} F(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})) \text{ s. t. } \mathbf{x}(\boldsymbol{\xi}) \in C(\boldsymbol{\xi}),\ \forall \boldsymbol{\xi} \in \Xi. \tag{5.3}$$

In practice, finding the WS solution may be impossible if the information about the future is not available. Therefore, HN approach is usually used and several deterministic reformulations are introduced below.

Definition 5.2.3. *Individual scenario (IS) deterministic reformulation* of the underlying program (5.1) is defined as

$$\min_{\mathbf{x}} F(\mathbf{x}, \boldsymbol{\xi}_s) \text{ s. t. } \mathbf{x} \in C(\boldsymbol{\xi}_s), \tag{5.4}$$

where $\boldsymbol{\xi}_s \in \Xi$ is a specified individual scenario.

[1] The distribution problem denotes the search for the distribution of the objective function value and optimal solution in terms of random vector, i. e. we are interested in finding all solutions and related optimal objective values for all realizations of the random parameters. The distribution problem is a generalization of sensitivity analysis in linear programming [49].

This reformulation is applicable when a recommendation from the experts concerning the significance of the individual scenarios is available. It means that we can replace the random vector by selected (most significant) scenario.

Another commonly used reformulation is obtained when the random vector is replaced by its expected value.

Definition 5.2.4. *Expected value (EV) deterministic reformulation* of the underlying program (5.1) is defined as

$$\min_{\mathbf{x}} F(\mathbf{x}, \mathbb{E}(\boldsymbol{\xi})) \text{ s. t. } \mathbf{x} \in C(\mathbb{E}(\boldsymbol{\xi})), \tag{5.5}$$

where $\mathbb{E}(\boldsymbol{\xi})$ is an expected value of $\boldsymbol{\xi}$ defined in 5.1.4.

We need some characteristic for measuring how good is the solution $\mathbf{x}^{EV}$ for the underlying objective function.

Definition 5.2.5. For the EV deterministic reformulation we define the *expected result of using the EV solution* (EEV) [6] as

$$\text{EEV} = \mathbb{E}(F(\mathbf{x}^{EV}, \boldsymbol{\xi})). \tag{5.6}$$

The EEV characteristic can be used to measure whether z^{EV} looks realistic by computing the difference $\text{EEV} - z^{EV}$ between the optimistic forecasted objective function value z^{EV} and true average cost computed by EEV.

If the EV solution $\mathbf{x}^{EV}$ is infeasible for the stochastic program, i. e. the expected value scenario $\mathbb{E}(\boldsymbol{\xi})$ does not correspond to any of the possible scenarios in Ξ, EEV value is set to be $+\infty$.

Other deterministic equivalents are defined through the use of several statistical characteristic of random variable (e. g. expected value, variance).

Definition 5.2.6. *Expected objective (EO) deterministic reformulation* of the underlying program (5.1) is defined (for the case of $C = \mathbb{R}^n$) as

$$\min_{\mathbf{x}} \mathbb{E}(F(\mathbf{x}, \boldsymbol{\xi})) \text{ s. t. } \mathbf{x} \in C. \tag{5.7}$$

For a comparison between EO and EV solution we can use Jensen's inequality [52]. It says that for function $F(\mathbf{x}, \boldsymbol{\xi})$ convex in $\boldsymbol{\xi}$, the inequality $\mathbb{E}(F(\mathbf{x}, \boldsymbol{\xi})) \geq F(\mathbf{x}, \mathbb{E}(\boldsymbol{\xi}))$ is satisfied. Furthermore we define the following value that measures how good or bad the EV solution $\mathbf{x}^{EV}$ is in terms of (5.7).

Definition 5.2.7. The *value of stochastic solution* (VSS) is defined as

$$\text{VSS} = \text{EEV} - z^{EO}. \tag{5.8}$$

VSS measures how much can be saved when the true HN approach is used instead of the EV approach. It means it is the loss by not considering the random variations and it assesses the value of knowing and using distributions on future outcomes [6]. A small value of the VSS means that the approximation of the stochastic program by the EV program is a good one.
The following concept is used to compare WS and HN solutions.

Definition 5.2.8. The *expected value of perfect information* (EVPI) is defined as

$$\text{EVPI} = z^{EO} - z^{WS}. \tag{5.9}$$

It tells us how much we should pay to obtain perfect information about the future. The large EVPI says that the information about the future is valuable, a small value of EVPI informs about little savings when we reach perfect information.

Theorem 5.2.9. *The following inequalities are satisfied for various deterministic reformulations.*

$$z^{WS} \leq z^{EO} \leq \text{EEV}.$$

If $F(\mathbf{x}, \boldsymbol{\xi})$ *is convex in* $\boldsymbol{\xi}$ *then* $z^{EV} \leq z^{WS}$.

Proof. The proof can be found in [6]. □

We are interested in engineering problems where no further information about the future is usually available, and therefore, the VSS becomes more practically relevant than the EVPI. But sometimes more information can be obtained for example through more accurate weather forecast which allows better estimation of e. g. the wind characteristics (see the second engineering problem described in the section 6.1.2).

It is clear that the expected value does not guarantee that there are no outliers. For this reason we can use some other reformulations such as variance objective (VO) [6] or min-max approach [14] that guarantee avoiding the large fluctuations of $F(\mathbf{x}, \boldsymbol{\xi})$ and therefore are more risk averse. It can be also useful to find compromise between minimizing cost and fluctuations. A way of dealing with it is to aggregate EO and VO deterministic reformulations into more criteria objective function [47].

There can be found various requirements for optimization in applications, e. g. to increase reliability of some equipment (especially in engineering). Therefore, we show how to optimize probability, i. e. how to minimize probability of high costs or maximize probability of low costs.

Definition 5.2.10. *Probabilistic objective (PO) deterministic reformulation* of the underlying program (5.1) is defined (for the case of $C = \mathbb{R}^n$) as

$$\min_{\mathbf{x}} \mathbb{P}(F(\mathbf{x}, \boldsymbol{\xi}) > b) \text{ s. t. } \mathbf{x} \in C, \tag{5.10}$$

where $b \in \mathbb{R}$ is a certain upper bound for the optimal objective function value that we do not want to exceed.

Theorem 5.2.11. *For discrete finite probability distribution of* $\boldsymbol{\xi}$ *the following integer program solves (5.10):*

$$\min_{z,\mathbf{x}} \{z | F(\mathbf{x}, \boldsymbol{\xi}_s) \leq b + M_p(1 - \delta_s), \sum_{s=1}^{R} p_s(1 - \delta_s) = z, \delta_s \in \{0, 1\}, s = 1, \ldots, R\}, \tag{5.11}$$

where $p_s = \mathbb{P}(F(\mathbf{x}, \boldsymbol{\xi}) = F(\mathbf{x}, \boldsymbol{\xi}_s))\ \forall \mathbf{x} \in \mathbb{R}^n$, $F(\mathbf{x}, \boldsymbol{\xi}_s) - b$ *is bounded from above by* M_p *for* $\forall \mathbf{x} \in C = \mathbb{R}^n$, $\boldsymbol{\xi}_s \in \Xi$ *and* R *is a number of possible realizations of* $\boldsymbol{\xi}$.

Proof. The proof can be found in [47]. □

We can define PO reformulation for maximizing probability in similar way. Another deterministic equivalent introduces the quantile as a bound for the objective function level [47].

Now we will assume that the random elements are contained in the constraints, which define feasible set, i. e. $C(\omega) = \{\mathbf{g}(\mathbf{x}, \omega) \leq \mathbf{0}, \mathbf{x} \in X\}$ as in (5.1). It must be specified

what is meant by feasibility. Some values of $\mathbf{x}$ can satisfy inequalities for some ω and violate these conditions for other ω. Usually it is unrealistic to require that inequalities should hold for all $\omega \in \Omega$. Several approaches are used to introduce a meaningful notion of feasibility.

One of them is to consider the expected values of constraint nonlinear functions.

Definition 5.2.12. *Expected value constraint* deterministic reformulation is defined as

$$C = \{\mathbb{E}(g_i(\mathbf{x}, \omega)) \leq 0,\ i = 1, \ldots, m,\ \mathbf{x} \in X\}. \tag{5.12}$$

Another way to define the feasible set is to use constraints on the probability of satisfying inequalities.

Definition 5.2.13. *Probabilistic (chance) constraint* deterministic reformulation is defined as

$$C = \{P(g_i(\mathbf{x}, \omega) \leq 0) \geq 1 - \alpha_p,\ i = 1, \ldots, m,\ \mathbf{x} \in X\}, \tag{5.13}$$

where $\alpha_p \in [0; 1]$ is a fixed value. If $\alpha_p = 0$, probabilistic constraints in (5.13) mean that $g_i(\mathbf{x}, \omega) \leq 0$, $i = 1, \ldots, m$ should hold for a. e. $\omega \in \Omega$.[2]

Theorem 5.2.14. *For discrete finite probability distribution of $\boldsymbol{\xi}$ with R scenarios with corresponding probabilities p_s, the following two inequalities are equivalent with the individual chance constraints $P(g_i(\mathbf{x}, \omega) \leq 0) \geq 1 - \alpha_p$, $i = 1, \ldots, m$:*

$$g_i(\mathbf{x}, \omega_s) - M_p(1 - \delta_s) \leq 0,\ i = 1, \ldots, m,\ s = 1, \ldots, R \tag{5.14}$$

$$\sum_{s=1}^{R} p_s \delta_s \geq 1 - \alpha_p \tag{5.15}$$

where $\delta_s \in \{0, 1\}$, $s = 1, \ldots, R$ are the binary variables and M_p is a sufficiently large number which bounds $g_i(\mathbf{x}, \omega_s)$ for $i = 1, \ldots, m$, $s = 1, \ldots, R$ from above.

Proof. The proof can be derived using [47]. □

We can view some optimization problems as two-stage problems where the decision vector has two distinct parts. At the first stage, before a realization of the corresponding random vector $\boldsymbol{\xi}$ becomes known, we choose the first-stage decision variables $\mathbf{x}$ to optimize the expected value of an objective function which is the optimal value of the second stage optimization problem. The value of the second part of the decision vector $\mathbf{y}$ can be chosen after the realization of $\boldsymbol{\xi}$ becomes known and generally depends on the realization of $\boldsymbol{\xi}$ and on the choice of $\mathbf{x}$. The second-stage problem can be viewed as a penalty term for violating the constraints which contain random elements, therefore these problems are called the problems with *recourse*.

The figure 5.1 shows the scenario-tree like representations of WS, EV and EO reformulations of two-stage problems.

This model may appear very restricted in its dynamical aspects. However, "stages" of the model do not necessarily refer to time units and they correspond to stages in the decision process instead. It means that the first-stage variable refers to all the decisions that must be taken before there will be any information about realization of random vector. While the second-stage variable models all the decisions that will be made after the available information about realization of random elements will be collected [64].

[2] For a. e. $\omega \in \Omega$ means that the event happens for almost every realization of the random vector $\boldsymbol{\xi}$.

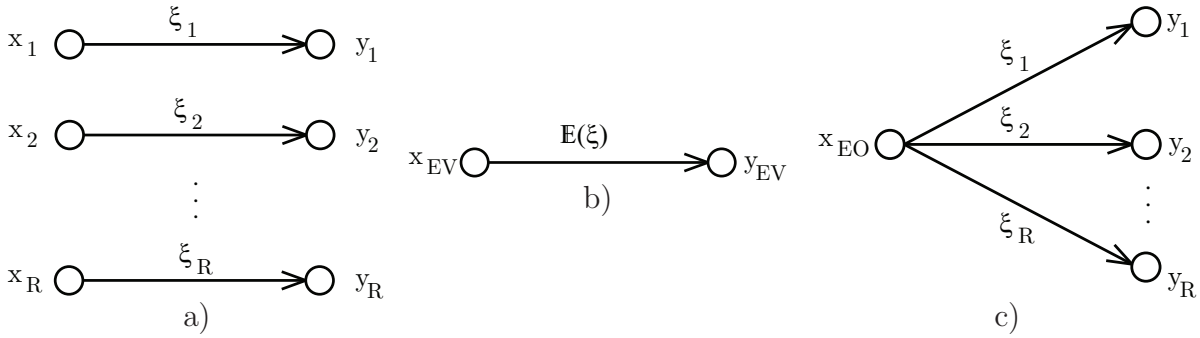


Figure 5.1: The sequence of decisions in deterministic reformulations a) WS, b) EV and c) EO.

The figures 6.4, 6.5 and 6.16 show these two stages together with corresponding decision variables for three engineering problems and it is evident that they need not to refer to all time units. In the figures 6.4, 6.5, the first-stage is connected with the time $t = 0$ while the second-stage is related to the rest times $t = 1$, $t = 2, \ldots, t = M$. On the other hand, the figure 6.16 shows the static model where the stages are by no means related to time.

Furthermore, a two-stage program combines WS and HN approaches into one model. The first-stage decision is made before the realizations of random variables are observed, i. e. it corresponds with HN approach, while the second-stage decision is taken after the realization becomes known, i. e. it relates to WS approach.

The most common two-stage programs are the linear ones, therefore, we provide their formulation.

Definition 5.2.15. *Two-stage stochastic linear program* is defined as follows

$$\min_{\mathbf{x}} \left(\mathbf{c}^T \mathbf{x} + \mathbb{E}(Q(\mathbf{x}, \boldsymbol{\xi})) \right) \text{ s. t. } \mathbb{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \tag{5.16}$$

where $Q(\mathbf{x}, \boldsymbol{\xi})$ is the optimal value of the second stage problem

$$\min_{\mathbf{y}} \mathbf{q}^T \mathbf{y} \text{ s. t. } \mathbb{T}\mathbf{x} + \mathbb{W}\mathbf{y} = \mathbf{h}, \mathbf{y} \geq \mathbf{0}. \tag{5.17}$$

The second-stage problem depends on the vector $\boldsymbol{\xi} = (\mathbf{q}, \mathbf{h}, \mathbb{T}, \mathbb{W})$ where some elements can be random. The matrices $\mathbb{T}$ and $\mathbb{W}$ are called technology and recourse matrices. If the matrix $\mathbb{W}$ is fixed, the program (5.16)-(5.17) is called the two-stage program with fixed recourse. These programs were originated by Dantzig in 1955 [12].

Theorem 5.2.16. *Let us assume that the random vector $\boldsymbol{\xi}$ has a discrete distribution with finite number R of possible realizations $\boldsymbol{\xi}_s = (\mathbf{q}_s, \mathbf{h}_s, \mathbb{T}_s, \mathbb{W}_s)$ with the corresponding probabilities p_s. Then the two-stage discrete stochastic linear program can be written as one large linear programming problem*

$$\min_{\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_R} \left(\mathbf{c}^T \mathbf{x} + \sum_{s=1}^{R} p_s \mathbf{q}_s^T \mathbf{y}_s \right) \tag{5.18}$$

$$\text{s. t. } \mathbb{A}\mathbf{x} = \mathbf{b}, \mathbb{T}_s \mathbf{x} + \mathbb{W}_s \mathbf{y}_s = \mathbf{h}_s, \mathbf{x} \geq \mathbf{0}, \mathbf{y}_s \geq \mathbf{0}, s = 1, \ldots, R. \tag{5.19}$$

Proof. The proof can be found in [52]. □

All random parameters depend on random elements $\omega_s \in \Omega$, i. e. $\mathbf{y}_s = \mathbf{y}(\omega_s)$ etc.
The size of the obtained linear program can be enormous therefore we relax it by replacing the first-stage decision vector $\mathbf{x}$ by R possibly different vectors $\mathbf{x}_s$.

Theorem 5.2.17. *The two-stage discrete stochastic linear program with nonanticipativity constraints can be written as*

$$\min_{\substack{\mathbf{x}_1,\dots,\mathbf{x}_R \\ \mathbf{y}_1,\dots,\mathbf{y}_R}} \sum_{s=1}^{R} p_s(\mathbf{c}^T\mathbf{x}_s + \mathbf{q}_s^T\mathbf{y}_s) \tag{5.20}$$

$$\text{s. t. } \mathbb{A}\mathbf{x}_s = \mathbf{b}, \mathbb{T}_s\mathbf{x}_s + \mathbb{W}_s\mathbf{y}_s = \mathbf{h}_s, \mathbf{x}_s \geq 0, \mathbf{y}_s \geq \mathbf{0}, s = 1,\dots,R, \tag{5.21}$$

$$\mathbf{x}_s = \sum_{i=1}^{R} p_i\mathbf{x}_i,\ s = 1,\dots,R. \tag{5.22}$$

This program is equivalent to (5.18) in the sense that they have the same optimal objective values and related solutions.

Proof. The proof can be found in [52]. □

Nonanticipativity constraints (5.22) ensure that the first-stage decision variables do not depend on the second-stage realization of the random data. But this program is still not separable in the sense that it cannot be split into R smaller linear programming problems. For this reason, many decomposition methods have been developed where one version of the dual decomposition methods (progressive hedging algorithm) is introduced in the chapter 7.

We cannot restrict to linear models because we will solve the nonlinear ones. Therefore, we provide general formulation of two-stage models [52].

Definition 5.2.18. *Two-stage stochastic program with nonanticipativity constraints* is defined as follows

$$\min_{\mathbf{x}(\cdot)\in\mathcal{X},\mathbf{y}(\cdot)\in\mathcal{Y}} \mathbb{E}(F(\mathbf{x}(\omega),\mathbf{y}(\omega),\boldsymbol{\xi}(\omega))) \tag{5.23}$$

$$\text{s. t. } G_i(\mathbf{x}(\omega),\mathbf{y}(\omega),\boldsymbol{\xi}(\omega)) \leq 0,\ i = 1,\dots,m,\ \mathbf{x}(\omega) \in X,\ \mathbf{y}(\omega) \in Y,\ \text{a. e. } \omega \in \Omega, \tag{5.24}$$

$$\mathbf{x}(\omega) = \mathbb{E}(\mathbf{x}(\omega)), \forall\omega \in \Omega, \tag{5.25}$$

where $F : \mathbb{R}^n \times \mathbb{R}^d \times \mathbb{R}^K \to \mathbb{R}$, $G_i : \mathbb{R}^n \times \mathbb{R}^d \times \mathbb{R}^R \to \mathbb{R}$, $i = 1,\dots,m$, $X \subseteq \mathbb{R}^n$, $Y \subseteq \mathbb{R}^d$, $\mathcal{X}$ is a space of measurable functions from Ω to $\mathbb{R}^n$ and $\mathcal{Y}$ is a space of measurable functions from Ω to $\mathbb{R}^d$.

Summary of theoretical properties of two-stage stochastic programs with recourse, which we are interested in, is presented in [6], [26] and [52]. It includes results concerning the structure of the feasible region, the recourse function and optimality conditions. Strictly speaking, it introduces theoretical assumptions ensuring convexity and continuity of the recourse function which result (together with other assumptions) in convexity of deterministic equivalent program.

The above formulated two-stage models are special cases of a more general *multi-stage stochastic programming model*, in which the decision variables and constraints are divided into groups corresponding to stages. The objective is to design the decision process in such a way that the expected value of the total cost is minimized while optimal decisions are allowed to be made at every time period. Precise mathematical formulation can be found in [52].

Chapter 6

Optimization models and their properties

We are interested especially in the applications of stochastic programming to the engineering problems [41] where the processes are very often governed by PDEs or ODEs and we can find many uncertainties there. As the example we can mention shape optimization [20]. Algebraic description of the optimization problems from this field exists rarely, therefore the numerical approach is used instead. "Black-boxes" (e. g. ANSYS) are used for finding the numerical solution very often in the engineering. Our approach should therefore be the intermediate stage between algebraic approach and "black-boxes" approach.

In this chapter we will introduce three engineering problems, we will formulate them as two-stage stochastic programs, analyse their properties and provide their solutions using several approximations and other analyses such as sensitivity analysis.

6.1 PDE constrained stochastic programming models

According to the aims of the Ph. D. thesis from chapter 2, we will study several engineering optimization problems of a dynamic nature, e. g. describing vibrations, leading to PDE constrained stochastic programs.

6.1.1 Stochastic optimization of transverse vibrations of a string

As the first illustrative example we consider the initial-boundary problem with hyperbolic (wave) equation describing the transverse vibration of a string (see Figure 6.1):

$$\frac{\partial^2 v}{\partial t^2}(x,t) = a^2 \frac{\partial^2 v}{\partial x^2}(x,t) + h(x,t),\ x \in \langle 0, l\rangle,\ t \in \langle 0, T\rangle, \tag{6.1}$$

where l is the string length [m], $a^2 = \frac{\sigma}{\mu}$, σ is the tension in the string [Pa], μ is the mass of the string per unit length [$\mathrm{kg\,m^{-1}}$], $v(x,t)$ is the displacement [m], $h(x,t)$ is the dynamic load [$\mathrm{N\,kg^{-1}}$]. The boundary conditions meaning zero transverse displacements at the clamped end points are

$$v(0,t) = 0,\ v(l,t) = 0 \tag{6.2}$$

and the initial conditions describing initial displacement and velocity are

$$v(x,0) = \varphi(x),\ \frac{\partial v}{\partial t}(x,0) = \psi(x). \tag{6.3}$$

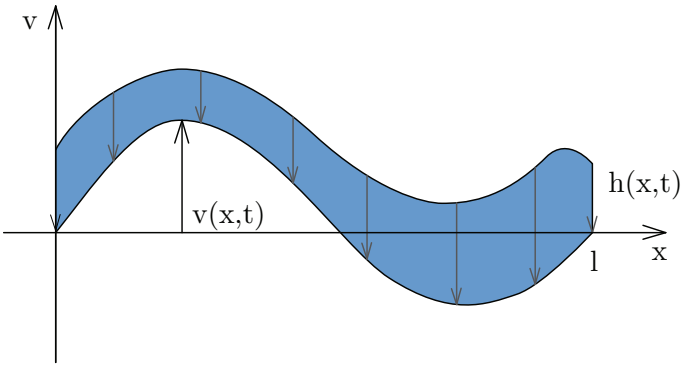


Figure 6.1: The wave equation models a loaded vibrating string.

Some information about quality of applied discretization method can be obtained from the comparison of the analytic solution with the numerical solution [73]. The analytic solution of this equation is well known and can be obtained via Fourier method (e. g. [44]).

For selected values of load $h(x,t) = \begin{cases} \frac{2w}{l}x,\ x \in \langle 0, \frac{l}{2}\rangle \\ -\frac{2w}{l}x + 2w,\ x \in (\frac{l}{2}, l\rangle \end{cases}$, initial displacement $\varphi(x) = \sin \pi x + \sin 2\pi x$ and initial velocity $\psi(x) = 0$, the analytic solution has the following form:

$$v(x,t) = \sin \pi x \cos \omega_1 t + \sin 2\pi x \cos \omega_2 t + \sum_{n=1}^{\infty} \frac{8w}{\omega_n^2 n^2 \pi^2} \sin \frac{n\pi}{2} \sin n\pi x (1 - \cos \omega_n t), \quad (6.4)$$

where $\omega_n = \frac{a\pi n}{l}$ is an angular frequency of vibration. Only finite number N_1 of terms in the infinite series is added.

Numerical solution is found as an approximation of the continuous solution of PDE (6.1) by space and time discretization. The difference equations are derived from PDE via simple finite difference method (e. g. [33]) with uniform grid spacing $x_i = id$, $i = 0, \ldots, N$, $d = \frac{l}{N}$ and $t_j = j\tau$, $j = 0, \ldots, M$, $\tau = \frac{T}{M}$. The discretization grid is shown in the figure 6.2. For convenience we use the following notation: $v(x_i, t_j) = v_{i,j}$, $h(x_i, t_j) = h_{i,j}$, $\varphi(x_i) = \varphi_i$ and $\psi(x_i) = \psi_i$.

The central-difference formulas for approximating $\frac{\partial^2 v}{\partial x^2}(x,t)$ and $\frac{\partial^2 v}{\partial t^2}(x,t)$ are

$$\frac{\partial^2 v}{\partial x^2}(x_i, t_j) = \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{d^2} + \mathcal{O}(d^2), \quad (6.5)$$

$$\frac{\partial^2 v}{\partial t^2}(x_i, t_j) = \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{\tau^2} + \mathcal{O}(\tau^2). \quad (6.6)$$

We drop the terms $\mathcal{O}(\tau^2)$ and $\mathcal{O}(d^2)$ and use the approximation $V_{i,j} \approx v_{i,j}$.

The difference formulas (6.5) and (6.6) are substituted into (6.1) and we obtain the difference equation of the transverse vibration for $i = 1, \ldots, N-1$, $j = 1, \ldots, M-1$ (values in red grid points in the figure 6.2):

$$\frac{V_{i,j+1} - 2V_{i,j} + V_{i,j-1}}{\tau^2} = a^2 \frac{V_{i+1,j} - 2V_{i,j} + V_{i-1,j}}{d^2} + h_{i,j}. \quad (6.7)$$

The substitution $r = \frac{a\tau}{d}$ is introduced in (6.7) and we obtain the relation

$$V_{i,j+1} = (2 - 2r^2)V_{i,j} + r^2(V_{i+1,j} + V_{i-1,j}) - V_{i,j-1} + \tau^2 h_{i,j}. \quad (6.8)$$

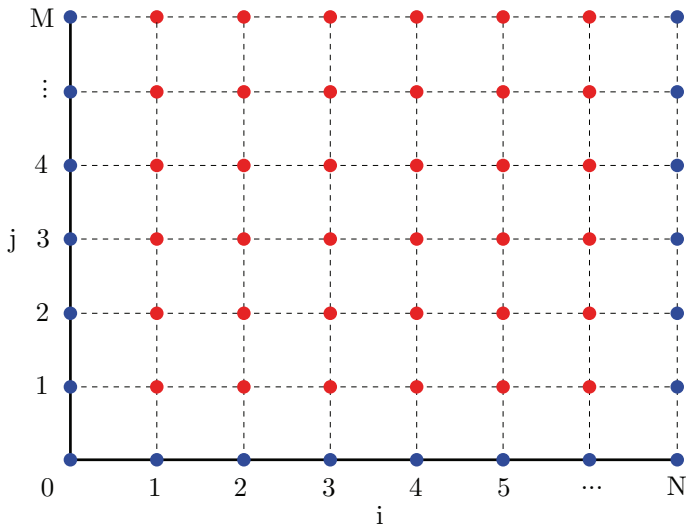


Figure 6.2: Discretization grid.

This discretization method is explicit therefore to guarantee stability of solution, it is necessary that $r \leq 1$ (i. e. there are limitations on discretization steps d and τ).
The values for $i = 0$ and $i = N$ (blue grid points in the figure 6.2) are given by the boundary conditions (6.2):

$$V_{0,j} = 0,\ V_{N,j} = 0,\ j = 0, \ldots, M. \tag{6.9}$$

The values corresponding to $j = 0$ and $j = 1$ must be supplied in order to use (6.8) to compute the values for $j = 2$. From the initial condition (6.3) we get values of displacement for $j = 0$ (blue grid points in the figure 6.2)

$$V_{i,0} = \varphi_i,\ i = 0, \ldots, N. \tag{6.10}$$

The values for $j = 1$ are constructed via the Taylor formula of order 2:

$$V_{i,1} = V_{i,0} + \dot{V}_{i,0}\tau + \frac{\ddot{V}_{i,0}\tau^2}{2}. \tag{6.11}$$

Hence, we obtain the following equation:

$$V_{i,1} = \varphi_i + \psi_i\tau + \frac{r^2}{2}(\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}) + \frac{\tau^2}{2}h_{i,0},\ i = 1, \ldots, N-1. \tag{6.12}$$

The equations (6.12) and (6.8) can be written in matrix form as follows:

$$\mathbf{V}_1 = \mathbf{\Phi} + \tau\mathbf{\Psi} + \frac{1}{2}\mathbb{K}_1\mathbf{\Phi} + \frac{\tau^2}{2}\mathbf{h}_0, \tag{6.13}$$

$$\mathbf{V}_{j+1} = \mathbb{K}\mathbf{V}_j - \mathbf{V}_{j-1} + \tau^2\mathbf{h}_j,\ j = 1, \ldots, M-1, \tag{6.14}$$

where $\mathbb{K}_1 = \begin{pmatrix} -2r^2 & r^2 & 0 & \ldots & 0 \\ r^2 & -2r^2 & r^2 & \ldots & 0 \\ & & \vdots & & \\ 0 & \ldots & r^2 & -2r^2 & r^2 \\ 0 & \ldots & 0 & r^2 & -2r^2 \end{pmatrix}$ is a square matrix of order $N-1$,

$$\mathbf{\Phi} = \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_{N-1} \end{pmatrix}, \mathbf{\Psi} = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_{N-1} \end{pmatrix}, \mathbb{K} = \begin{pmatrix} 2-2r^2 & r^2 & 0 & \dots & 0 \\ r^2 & 2-2r^2 & r^2 & \dots & 0 \\ & & \vdots & & \\ 0 & \dots & r^2 & 2-2r^2 & r^2 \\ 0 & \dots & 0 & r^2 & 2-2r^2 \end{pmatrix}$$

is a square matrix of order $N-1$, $\mathbf{h}_j = \begin{pmatrix} h_{1,j} \\ \vdots \\ h_{N-1,j} \end{pmatrix}$. Vector $\mathbf{V}_j = (V_{1,j}, \dots, V_{N-1,j})^T$, $j = 0, 1, \dots, M$, is the approximation of $v(x,t)$.

Numerical solution is found using optimization software GAMS with solver of linear programming problems CPLEX, analytic solution and graphical results are obtained using MATLAB. Comparison of the analytic and the numerical solution for the following parameters $l = 1\,\mathrm{m}$, $a^2 = 2\,\mathrm{s}^{-1}$, $T = 0.5\,\mathrm{s}$, $w = 200$, $N = 10$, $M = 10$, $N_1 = 20$ is presented in the figure 6.3.

The relative error of the adopted discretization method is $E = \max\limits_{\substack{0 \le i \le N \\ 0 \le j \le M}} \frac{|v_{i,j} - V_{i,j}|}{|v_{i,j}|} \cdot 100 =$ $= 2.3\%$. It turns out that the accuracy of described approximation depends on sizes of discretization steps d and τ (the error is only $E = 0.6\%$ for $N = 20$, $M = 20$).

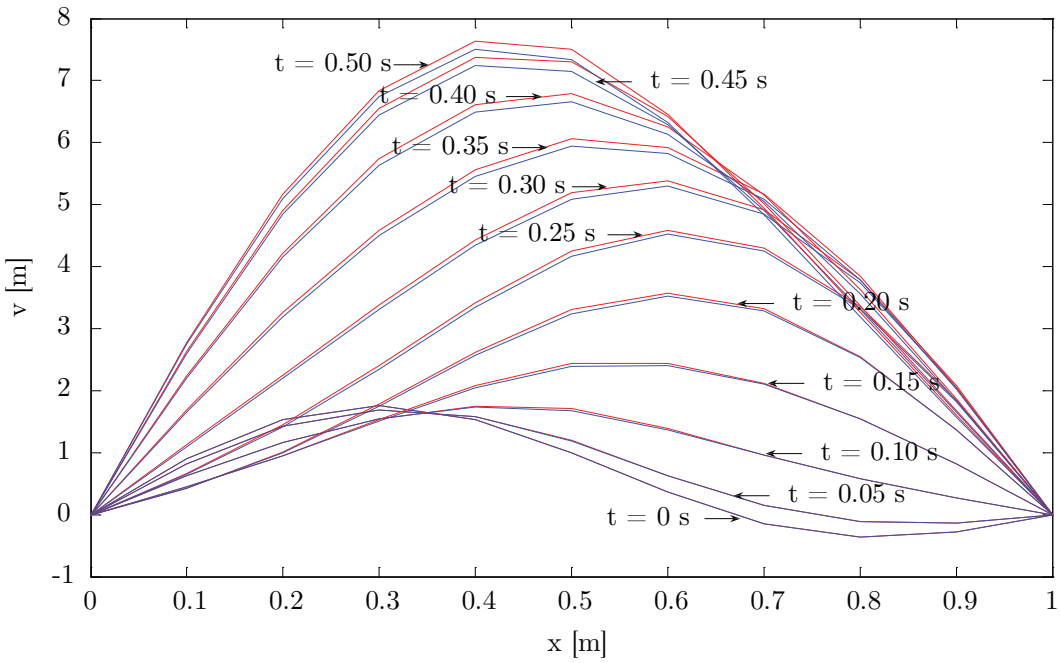


Figure 6.3: Numerical (red line) and analytic (blue line) solution of problem (6.1).

In case of real-world engineering problems, the uncertain parameters that can be modelled by random elements are also involved. Therefore, we replace deterministic dynamic load $h(x,t)$ in (6.1) by a stochastic dynamic load $h(\xi, x, t)$, where $\xi : \Xi \to \mathbb{R}$ is the random variable (see definition 5.1.3), $(\Xi, \mathcal{B}, \mathbb{P})$ is the probability space (see definition 5.1.1) and $h : \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}$.

In all our presented illustrative engineering problems as well as in many real engineering problems, the stage-based decision process is more adequate than continuously dynamic one like in optimal control. As it is often necessary to make decision before

a realization of the corresponding random variables becomes known and after it, two-stage stochastic programming offers a promising approach (see [10] for the first original models in the area of shape optimization and [74], [77] for the original model in optimization of vibrations). Hence, our models will include two types of decision variables. The first-stage decision variable does not depend on a realization of ξ, i. e. it corresponds with HN approach, while the second-stage decision variable plays the role of a recourse action to consequences of the first-stage decision variable and of course depends on a realization of ξ, i. e. it relates to WS approach.

The aim of the optimization is to obtain an optimal design of two types of decision variables describing damping forces while the difference between actual and required displacement is minimized (6.15), see the following underlying program:

$$\min_{e,\, g(\xi), v(\xi)} \int_0^T \int_0^l (v(\xi, x, t) - u(x, t))^2 \,\mathrm{d}x \,\mathrm{d}t \tag{6.15}$$

$$\text{s. t.} \quad \frac{\partial^2 v}{\partial t^2}(\xi, x, t) = a^2 \frac{\partial^2 v}{\partial x^2}(\xi, x, t) + \mathbb{T}(\xi, x) e(x) + g(\xi, x, t) + h(\xi, x, t), \tag{6.16}$$

$$x \in \langle 0, l\rangle,\ t \in \langle 0, T\rangle,\ \xi \in \Xi,$$

$$v(\xi, 0, t) = 0,\ v(\xi, l, t) = 0,\ t \in \langle 0, T\rangle,\ \xi \in \Xi, \tag{6.17}$$

$$v(\xi, x, 0) = \varphi(x),\ \frac{\partial v}{\partial t}(\xi, x, 0) = \psi(x),\ x \in \langle 0, l\rangle,\ \xi \in \Xi, \tag{6.18}$$

$$g(\xi, x, 0) = 0,\ x \in \langle 0, l\rangle,\ \xi \in \Xi, \tag{6.19}$$

$$|v(\xi, x, t)| \leq v_{limit},\ x \in \langle 0, l\rangle,\ t \in \langle 0, T\rangle,\ \xi \in \Xi, \tag{6.20}$$

where $u(x, t)$ is the required displacement [m], $e(x)$ is the first-stage decision variable [Nkg^{-1}], $g(\xi, x, t)$ is the second-stage decision variable [Nkg^{-1}], $v(\xi, x, t)$ is the displacement [m]. We assume that the external force $e(x)$ acts indirectly on the string, i. e. it takes effect through mechanical equipments (such as springs, dampers, ...) [75]. Therefore, a random matrix $\mathbb{T}(\xi, x)$ is included into model. More precisely, matrix $\mathbb{T}(\xi, x)$ is chosen as $\mathbb{T}(\xi, x) = \mathbf{T}_1(x)\xi$ (e. g. [26]) because there are different transfer equipments along the length of the string. So, $\mathbf{T}_1(x)$ describes transfer characteristics of these equipments and ξ describes random influences of the environment on equipment characteristics. The PDE (6.16) describes transverse displacement of the string, boundary conditions for clamped end points given by (6.17) mean that there are zero transverse displacements. Furthermore, the initial conditions given by (6.18) define initial displacement and velocity and there is also initial condition (6.19) for second-stage variable. Finally, the displacement of the string must be bounded, see (6.20). The objective function has been chosen in the integral form to include influence of every time and all parts of console.

We can see on our problems from the two points of view – mathematical (modelling) and interpretative. Concepts of solution (state variable) and decision are equivalent from the mathematical viewpoint. On the other hand, interpretative point of view distinguishes between these terms. Decision variables are connected with some action while the state variables just occur and we can only observe them. Therefore, the displacement $v(\xi, x, t)$ can be regarded as a decision variable, strictly speaking the second-stage decision variable, or it can be considered as a state variable. Models distinguishing states and decisions are called state-space decision models [40], see figure 6.4 for the scheme of our problem. But we will consider the mathematical viewpoint, therefore, there is no difference between $g(\xi, x, t)$ and $v(\xi, x, t)$, see [45]. The scheme of the time evolution of our optimization problem is in the figure 6.5.

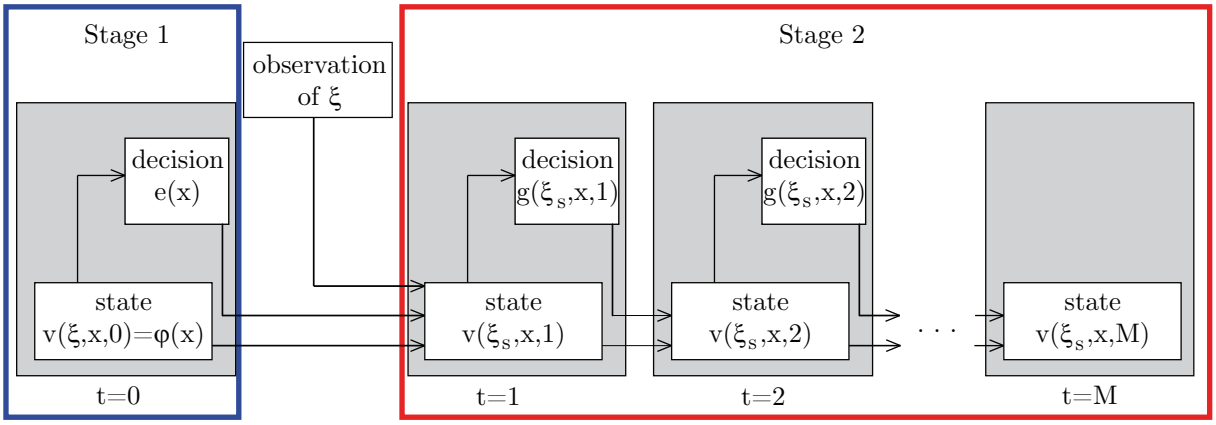


Figure 6.4: Scheme of a state-space model of the two-stage stochastic program (6.21)-(6.26).

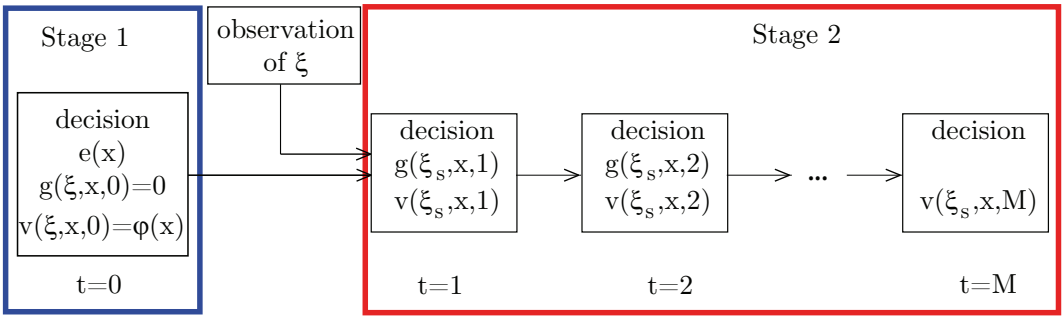


Figure 6.5: Scheme of the two-stage stochastic program (6.21)-(6.26).

The underlying program (6.15)-(6.20) is syntactically correct but its semantics is not discussed, as is usual in stochastic programming (see [52]). Deterministic reformulations are further defined. It means most often that the objective function (6.15) is optimized on average, i. e. EO reformulation is used (see definition 5.2.6), where the mean value is taken with respect to a known probability measure $\mathbb{P}$ on $(\Xi, \mathcal{B})$, and almost surely satisfying of the constraints (6.16)-(6.20). EV reformulation (see definition 5.2.4), when the random variable ξ is replaced by its mean value $\mathbb{E}(\xi)$, offers another possibility. The advantage of EV reformulation is the significant simplification of computing while the positive of EO reformulation is more reliable results. These two approaches can be compared in terms of the value of stochastic solution as will be presented later.

EO deterministic reformulation changes the underlying program (6.15)-(6.20) into a syntactically correctly defined continuous stochastic program:

$$\min_{e,\, g(\xi), v(\xi)} \ \mathbb{E}\left(\int_0^T\int_0^l (v(\xi,x,t)-u(x,t))^2 \,\mathrm{d}x\,\mathrm{d}t\right) \tag{6.21}$$

$$\text{s. t.} \quad \frac{\partial^2 v}{\partial t^2}(\xi,x,t) = a^2\frac{\partial^2 v}{\partial x^2}(\xi,x,t) + \mathbb{T}(\xi,x)e(x) + g(\xi,x,t) + h(\xi,x,t), \tag{6.22}$$

$$x \in \langle 0,l\rangle,\ t \in \langle 0,T\rangle,\ \text{a. e. } \xi \in \Xi,$$

$$v(\xi,0,t) = 0,\ v(\xi,l,t) = 0,\ t \in \langle 0,T\rangle,\ \text{a. e. } \xi \in \Xi, \tag{6.23}$$

$$v(\xi,x,0) = \varphi(x),\ \frac{\partial v}{\partial t}(\xi,x,0) = \psi(x),\ x \in \langle 0,l\rangle,\ \text{a. e. } \xi \in \Xi, \tag{6.24}$$

$$g(\xi,x,0) = 0,\ x \in \langle 0,l\rangle,\ \text{a. e. } \xi \in \Xi, \tag{6.25}$$

$$|v(\xi,x,t)| \le v_{limit},\ x \in \langle 0,l\rangle,\ t \in \langle 0,T\rangle,\ \text{a. e. } \xi \in \Xi, \tag{6.26}$$

We are not able to solve program (6.21)-(6.26) without proper approximations. The approximations are made in two steps. At first, scenario-based approach for approximation of random variable is used, see [52]. We assume that random variable ξ has a discrete probability distribution with a finite number R of equiprobable scenarios ξ_s with the corresponding probabilities $p_s = \mathbb{P}(\xi = \xi_s) = \frac{1}{R}$. Hence, the mean value is computed as follows (see definition 5.1.4): $\mathbb{E}(F(\xi,e,g(\xi),v(\xi))) = \sum_{s=1}^{R} p_s F(\xi_s,e,g_s,v_s)$.

The second step consists in discretization of the space x and time t coordinates in objective function and constraints. Following the recommendation of [4], we use simple finite difference method [33] with uniform grid spacing for $(N+1)(M+1)$ points: $x_i = id$, $i = 0,\ldots,N$, $d = \frac{l}{N}$ and $t_j = j\tau$, $j = 0,\ldots,M$, $\tau = \frac{T}{M}$. Derivation of difference equations is analogous to the previous section where the analytic and numerical solution of single PDE were compared. Some variables have now three indices s,i,j instead of two indices i,j and there are two more functions $e(x)$ and $g(\xi,x,t)$, see the following notation: $v(\xi_s,x_i,t_j) = v_{s,i,j}$, $e(x_i) = e_i$, $g(\xi_s,x_i,t_j) = g_{s,i,j}$, $h(\xi_s,x_i,t_j) = h_{s,i,j}$. Derivatives are replaced by central difference formulas (6.5) and (6.6) with only difference in number of indices and after several adjustments difference equations (6.28)-(6.29) are obtained. The objective function (6.21) is approximated by the composite Simpson's rule with coefficients $a_0 = 1 = a_N$, $a_{2k-1} = 4$, $a_{2k} = 2$, $k = 1,2,\ldots,\frac{N}{2}$, $b_0 = 1 = b_M$, $b_{2m-1} = 4$, $b_{2m} = 2$, $m = 1,2,\ldots,\frac{M}{2}$ and N, M are even numbers.

Hence, the continuous two-stage stochastic quadratic program (6.21)-(6.26) is approximated by a large deterministic quadratic program:

$$\min_{\mathbf{e},\mathbf{g}_s,\mathbf{V}_s} \ \sum_{s=1}^{R}\sum_{j=0}^{M}\sum_{i=0}^{N} p_s \frac{d\tau}{9} a_i b_j (V_{s,i,j} - u_{i,j})^2 \tag{6.27}$$

$$\text{s. t.} \quad \mathbf{V}_{s,1} = \mathbf{\Phi} + \tau\mathbf{\Psi} + \frac{1}{2}\mathbb{K}_1\mathbf{\Phi} + \frac{\tau^2}{2}\mathbf{F}_{s,0},\ s = 1,\ldots,R, \tag{6.28}$$

$$\mathbf{V}_{s,j+1} = \mathbb{K}\mathbf{V}_{s,j} - \mathbf{V}_{s,j-1} + \tau^2\mathbf{F}_{s,j},\ j = 1,\ldots,M-1,\ s = 1,\ldots,R, \tag{6.29}$$

$$V_{s,0,j} = 0,\ V_{s,N,j} = 0,\ j = 0,\ldots,M,\ s = 1,\ldots,R, \tag{6.30}$$

$$V_{s,i,0} = \varphi_i,\ i = 0,\ldots,N,\ s = 1,\ldots,R, \tag{6.31}$$

$$g_{s,i,0} = 0,\ i = 0,\ldots,N,\ s = 1,\ldots,R, \tag{6.32}$$

$$|V_{s,i,j}| \le v_{limit},\ i = 0,\ldots,N,\ j = 0,\ldots,M,\ s = 1,\ldots,R, \tag{6.33}$$

where the matrices $\mathbb{K}_1$, $\mathbb{K}$ and the vectors $\mathbf{\Phi}$, $\mathbf{\Psi}$ are the same as in the equations (6.13), (6.14) and $\mathbf{F}_{s,j} = \begin{pmatrix} \mathbb{T}(s,1)e_1 + g_{s,1,j} + h_{s,1,j} \\ \vdots \\ \mathbb{T}(s,N-1)e_{N-1} + g_{s,N-1,j} + h_{s,N-1,j} \end{pmatrix}$.

Vector $\mathbf{V}_{s,j} = (V_{s,1,j}, \ldots, V_{s,N-1,j})^T$, $j = 0, \ldots, M$, $s = 1, \ldots, R$, is the approximation of $v(\xi, x, t)$. Vector $\mathbf{g}_{s,j} = (g_{s,0,j}, \ldots, g_{s,N,j})^T$, $j = 0, \ldots, M-1$, $s = 1, \ldots, R$, is the approximation of the second-stage decision variable $g(\xi, x, t)$ and vector $\mathbf{e} = (e_1, \ldots, e_{N-1})^T$, is the approximation of the first-stage decision variable $e(x)$.

Program (6.27)-(6.33) is implemented in GAMS with solver of nonlinear programming problems CONOPT. The results are presented for the following input data: $l = 1\,\mathrm{m}$, $a^2 = 2\,\mathrm{s}^{-1}$, $T = 0.5\,\mathrm{s}$, numbers of grid points are $N = 10$, $M = 10$. We assume $R = 20$ and $R = 1000$ scenarios and the displacement limitation is $v_{limit} = 5\,\mathrm{m}$. Vector $\mathbf{T}_1(x)$ is generated from the uniform distribution $U(0,1)$ and $\xi \sim U(-5,2)$. We assume that the acting external load is a random combination of finitely many forces [10], i. e. $h(\xi, x, t) = \sum_{p=1}^{k} A_p(\xi) h_p(x,t)$. Basic loads for $k = 6$ are chosen as follows: $h_1(x,t) = bx(\frac{t}{\tau} + 1)$, $h_2(x,t) = -bx(\frac{t}{\tau} + 1)$, $h_3(x,t) = bx^2(\frac{t}{\tau} + 1)$, $h_4(x,t) = b(\frac{t}{\tau} + 1)$, $h_5(x,t) = 0$, $h_5(2d,t) = 20$, $h_6(x,t) = b \sin x(\frac{t}{\tau} + 1)^{-1}$, $b = 100$. Coefficients $A_p(\xi)$ of $h(\xi, x, t)$ are given as $A_p(\xi) \sim U(0,1)$ and $\sum_{p=1}^{k} A_p(\xi) = 1$ for $\forall \xi$. This normalization condition is guaranteed by dividing each $A_p(\xi)$ by $\sum_{p=1}^{k} A_p(\xi)$. Initial displacement and velocity are assumed as $\varphi(x) = \sin \pi x + \sin 2\pi x$, $\psi(x) = 0$, required displacement as $u(x,t) = 0$.

Several scenarios of external stochastic load and computed first-stage and second-stage variables and displacement are presented in the figures 6.6-6.9. Results of model with only 20 scenarios are displayed because of the clarity. But the numerical results are computed for the model with 1000 scenarios. The program ran on a notebook with Intel Core 2Duo 2GHz and 2GB RAM. Numbers of variables and constraints together with solution times are given in the table 6.1 for EO and EV deterministic reformulations.

No. of scenarios R	20	1000	2000	**Type of reformulation**
No. of variables	4270	213 010	426 010	EO reformulation
No. of constraints	2681	134 001	268 001	
CONOPT time [s]	0.1	100.1	222.1	
No. of variables	355	355	355	EV reformulation
No. of constraints	267	267	267	
CONOPT time [s]	0.01	0.01	0.01	

Table 6.1: Computational properties of mathematical programs.

The importance of randomness can be described by several criteria. The first one, called the value of stochastic solution (see definition 5.2.7), is defined as a difference between optimal objective function value EEV of the stochastic program with fixed first-stage variable computed from EV model and optimal objective function value z^{EO} of the approximated stochastic program (6.27)-(6.33). In our case for 1000 scenarios, we obtain $z^{EO} = 0.054$, $z^{EV} = 0.017$, $EEV = 0.115$ and $VSS = 0.061$. Hence, the optimal objective function value is more than twice better in case of including randomness into model as

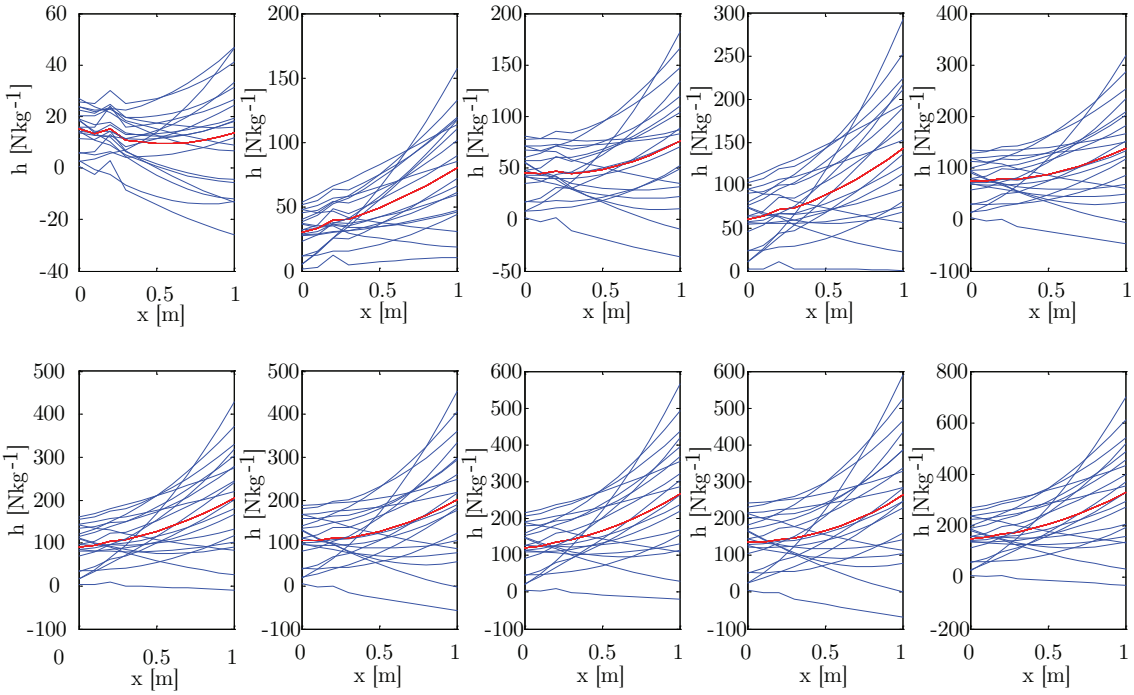


Figure 6.6: Stochastic load $h(\xi, x, t)$. Red line – EV model, blue lines – EO model.

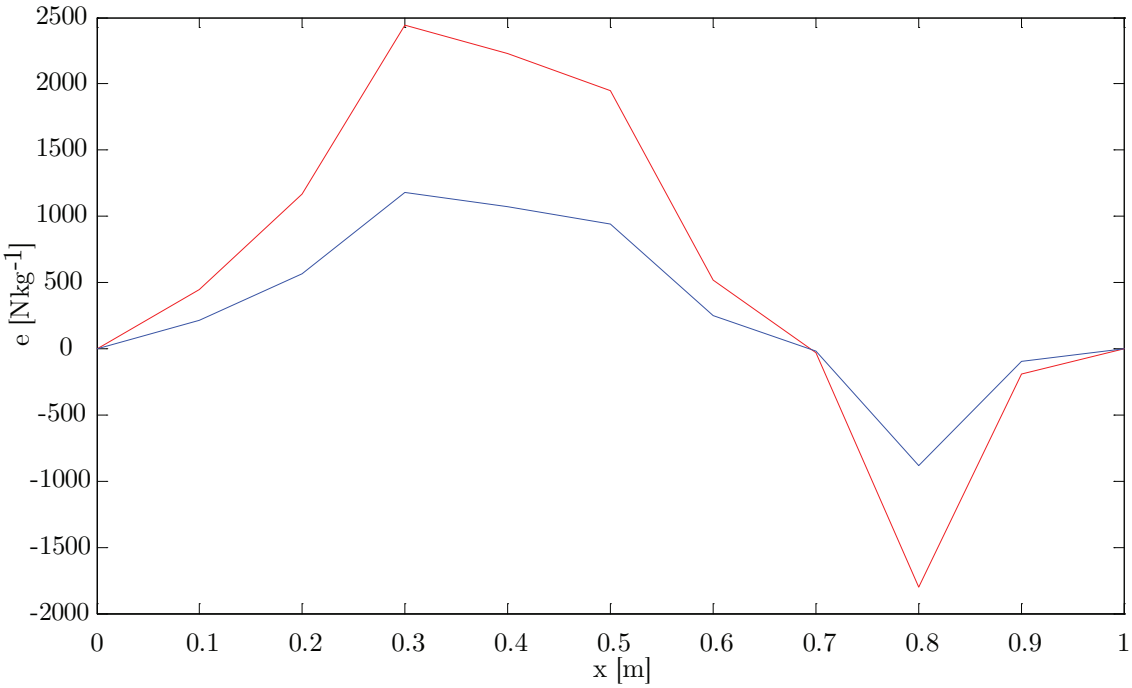


Figure 6.7: First-stage variable $e(x)$. Red line – EV model, blue lines – EO model.

in program (6.27)-(6.33) than in case of replacing the random variable by its mean value. The second criterion is called EVPI (see definition 5.2.8) and it makes sense to compute this value only if the perfect information about the future random outcomes is available, which we do not assume.

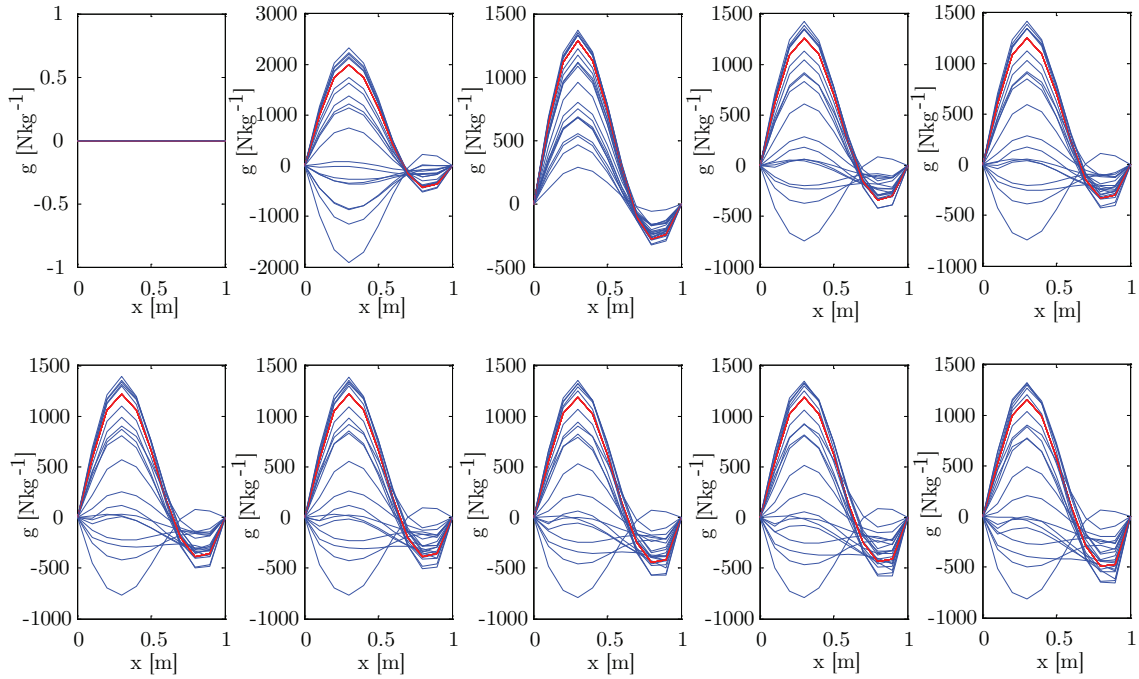


Figure 6.8: Second-stage variable $g(\xi, x, t)$. Red line – EV model, blue lines – EO model.

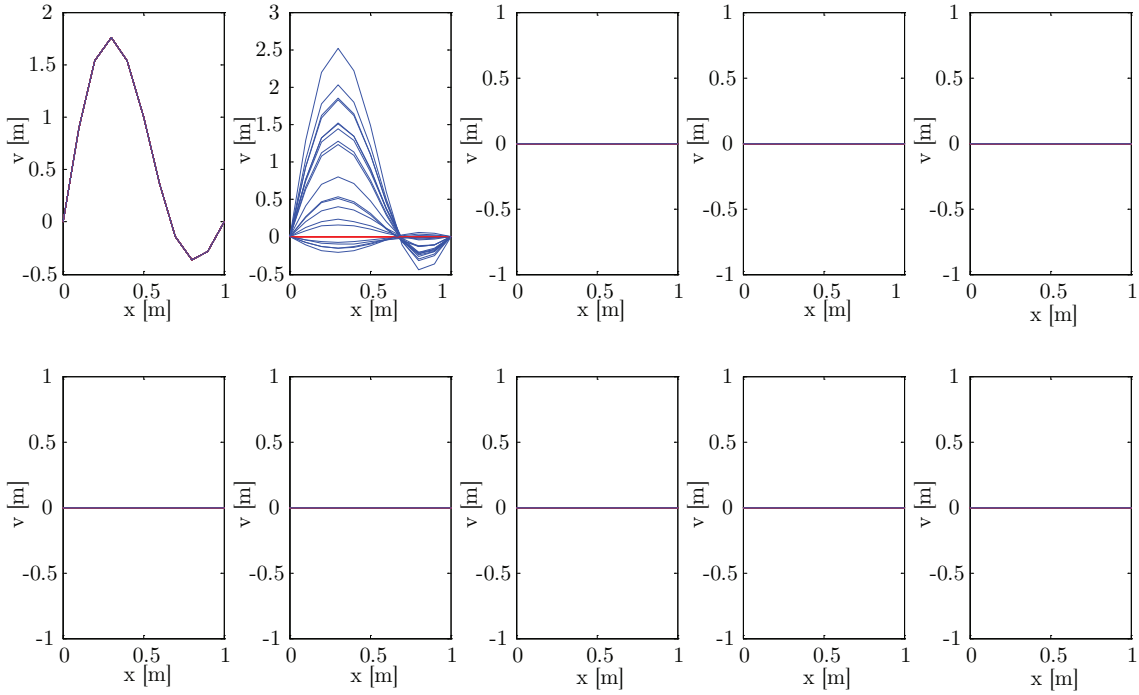


Figure 6.9: Displacement $v(\xi, x, t)$. Red line – EV model, blue lines – EO model.

6.1.2 Stochastic optimization of transverse vibrations of a console

Our second example is also from the area of civil engineering and it assumes the initial-boundary problem with the fourth order PDE describing the undamped forced transverse vibration of a console[1] [2]:

$$EJ\frac{\partial^4 v}{\partial x^4}(x,t) + \mu\frac{\partial^2 v}{\partial t^2}(x,t) = h(x,t),\ x \in \langle 0,l\rangle,\ t \in \langle 0,T\rangle, \tag{6.34}$$

with boundary conditions:

$$v(0,t) = 0,\ \frac{\partial v}{\partial x}(0,t) = 0,\ \frac{\partial^2 v}{\partial x^2}(l,t) = 0,\ \frac{\partial^3 v}{\partial x^3}(l,t) = 0,\ t \in \langle 0,T\rangle \tag{6.35}$$

and initial conditions:

$$v(x,0) = \varphi(x),\ \frac{\partial v}{\partial t}(x,0) = \psi(x),\ x \in \langle 0,l\rangle, \tag{6.36}$$

where l is the console length [m], T is the time of observation [s], E is the Young's modulus [Pa], J is the second moment of the cross section with respect to the axis z [m^4], μ is the specific mass per length unit [$\mathrm{kg\,m^{-1}}$], $v(x,t)$ is the displacement [m] and $h(x,t)$ is the external dynamic load [$\mathrm{Nm^{-1}}$].

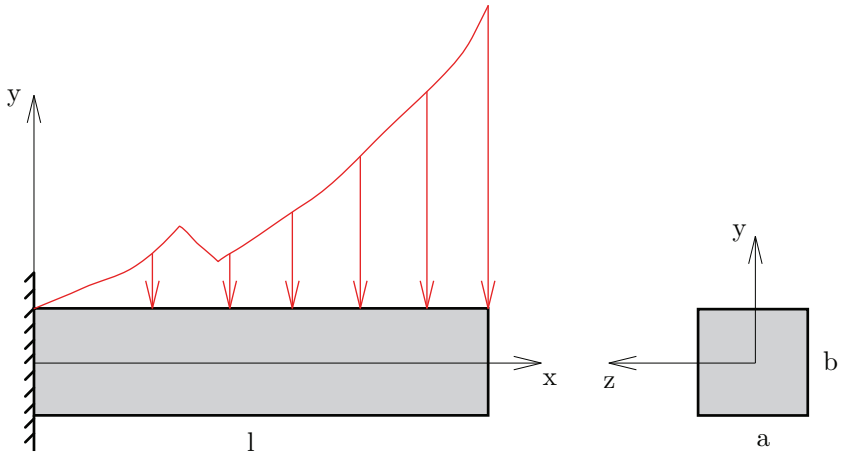


Figure 6.10: Scheme of loaded console and its cross section.

The analytic solution of this equation can be obtained via harmonic analysis (e. g. [2]). At first we consider forced free bar and express displacement and load as Fourier series. We obtain displacement and angular rotation at the end points. Then by superposition principle we add displacement of unforced console, which must satisfy boundary conditions. We could get some information about quality of applied discretization method from the comparison of the analytic solution with the numerical solution. But there is no closed-form solution (only as infinite series), therefore, we can obtain only approximation of the analytical solution by computing with finite series.

[1] A console (cantilever beam) is a beam with one end clamped and another end free. We assume that the dimensions of the constant cross section are substantially smaller than the length, i. e. the console is modelled by the prismatic bar and the PDE describes the vibrations of the centreline.

As we have mentioned above, engineering problems often involves randomness. Therefore, we will consider stochastic external load $h(\xi, x, t)$ acting on the console. Two-stage stochastic programming approach will be used in the same way as in the previous subsection.

The aim of the optimization is the same as in the previous problem, e. g. to obtain an optimal design of two types of decision variables while the difference between actual and required displacement is minimized (6.37), see the following underlying program:

$$\min_{e,\, g(\xi), v(\xi)} \int_0^T \int_0^l (v(\xi, x, t) - u(x, t))^2 \, \mathrm{d}x \, \mathrm{d}t \tag{6.37}$$

$$\text{s. t.} \quad EJ \frac{\partial^4 v}{\partial x^4}(\xi, x, t) + \mu \frac{\partial^2 v}{\partial t^2}(\xi, x, t) = \mathbb{T}(\xi, x) e(x) + g(\xi, x, t) + h(\xi, x, t), \tag{6.38}$$

$$x \in \langle 0, l \rangle,\ t \in \langle 0, T \rangle,\ \xi \in \Xi,$$

$$v(\xi, 0, t) = 0,\ \frac{\partial v}{\partial x}(\xi, 0, t) = 0,\ t \in \langle 0, T \rangle,\ \xi \in \Xi, \tag{6.39}$$

$$\frac{\partial^2 v}{\partial x^2}(\xi, l, t) = 0,\ \frac{\partial^3 v}{\partial x^3}(\xi, l, t) = 0,\ t \in \langle 0, T \rangle,\ \xi \in \Xi, \tag{6.40}$$

$$v(\xi, x, 0) = \varphi(x),\ \frac{\partial v}{\partial t}(\xi, x, 0) = \psi(x),\ x \in \langle 0, l \rangle,\ \xi \in \Xi, \tag{6.41}$$

$$g(\xi, x, 0) = 0,\ x \in \langle 0, l \rangle,\ \xi \in \Xi, \tag{6.42}$$

$$|v(\xi, x, t)| \leq v_{limit},\ x \in \langle 0, l \rangle,\ t \in \langle 0, T \rangle,\ \xi \in \Xi, \tag{6.43}$$

where $u(x, t)$ is the required displacement [m], $e(x)$ is the first-stage decision variable [Nm^{-1}], $g(\xi, x, t)$ is the second-stage decision variable [Nm^{-1}], $v(\xi, x, t)$ is the displacement [m] with the same direction as the axis y. We again assume that the force $e(x)$ acts indirectly on the console, i. e. it takes effect through different equipments. Therefore, a random matrix $\mathbb{T}(\xi, x)$ in the form $\mathbb{T}(\xi, x) = \mathbf{T}_1(x)\xi$ is included into model. Its interpretation is the same as in the previous subsection. The PDE (6.38) describes transverse displacement of the console, boundary conditions for the clamped end point given by (6.39) mean that there are zero transverse displacement and its slope, boundary conditions for the free end point given by (6.40) mean that there are zero bending moment and shear force. Furthermore, the initial conditions given by (6.41) define initial displacement and velocity and there is also initial condition (6.42) for second-stage variable. Finally, the displacement of the console must be bounded, see (6.43). The scheme of the time evolution of this optimization problem is in the figure 6.5.

We can interpret our problem as the vibrations of a roof console or supporting part of the wall. Then stochastic load $h(\xi, x, t)$ can be e. g. wind force or load generated by snow [32]. The first-stage decision variable $e(x)$ can be interpreted as a preliminary tension imposed to the console during building the house while the second-stage decision variable $g(\xi, x, t)$ is the force generated by mechanical equipment to balance the total load. Another interpretation of the presented problem is the optimization of the transverse oscillation of a television transmitter of tower structure (e. g. on the mountain Ještěd) loaded by stochastic wind force, where the first-stage decision variable $e(x)$ can be interpreted as a passive damping and the second-stage decision variable $g(\xi, x, t)$ represents an active damping.

The underlying program (6.37)-(6.43) depends as before on a realization of the random variable ξ. Therefore, appropriate deterministic equivalents are chosen and e. g. EO deterministic reformulation changes the underlying program into the following syntactically

correctly defined continuous two-stage stochastic quadratic program:

$$\min_{e,\, g(\xi), v(\xi)} \quad \mathbb{E}\left(\int_0^T\int_0^l (v(\xi,x,t)-u(x,t))^2\,\mathrm{d}x\,\mathrm{d}t\right) \tag{6.44}$$

$$\text{s. t.} \quad EJ\frac{\partial^4 v}{\partial x^4}(\xi,x,t)+\mu\frac{\partial^2 v}{\partial t^2}(\xi,x,t)=\mathbb{T}(\xi,x)e(x)+g(\xi,x,t)+h(\xi,x,t), \tag{6.45}$$

$$x\in\langle 0,l\rangle,\ t\in\langle 0,T\rangle,\ \text{a. e.}\ \xi\in\Xi,$$

$$v(\xi,0,t)=0,\ \frac{\partial v}{\partial x}(\xi,0,t)=0,\ t\in\langle 0,T\rangle,\ \text{a. e.}\ \xi\in\Xi, \tag{6.46}$$

$$\frac{\partial^2 v}{\partial x^2}(\xi,l,t)=0,\ \frac{\partial^3 v}{\partial x^3}(\xi,l,t)=0,\ t\in\langle 0,T\rangle,\ \text{a. e.}\ \xi\in\Xi, \tag{6.47}$$

$$v(\xi,x,0)=\varphi(x),\ \frac{\partial v}{\partial t}(\xi,x,0)=\psi(x),\ x\in\langle 0,l\rangle,\ \text{a. e.}\ \xi\in\Xi, \tag{6.48}$$

$$g(\xi,x,0)=0,\ x\in\langle 0,l\rangle,\ \text{a. e.}\ \xi\in\Xi, \tag{6.49}$$

$$|v(\xi,x,t)|\le v_{limit},\ x\in\langle 0,l\rangle,\ t\in\langle 0,T\rangle,\ \text{a. e.}\ \xi\in\Xi. \tag{6.50}$$

As we have mentioned in the previous subsection, the proper approximations must be made, i. e. scenario-based approach for the approximation of random variable and finite difference method for the discretization of the space x and time t coordinates in objective function and constraints are used. The notation is the same as in the previous subsection.

The central-difference formula for approximating $\frac{\partial^4 v}{\partial x^4}(\xi,x,t)$ is

$$\frac{\partial^4 v}{\partial x^4}(\xi_s,x_i,t_j)=\frac{v_{s,i+2,j}-4v_{s,i+1,j}+6v_{s,i,j}-4v_{s,i-1,j}+v_{s,i-2,j}}{d^4}+\mathcal{O}(d^2), \tag{6.51}$$

derivative $\frac{\partial^2 v}{\partial t^2}(\xi,x,t)$ is approximated by (6.6). We drop the terms $\mathcal{O}(\tau^2)$ and $\mathcal{O}(d^2)$ and use the approximation $V_{s,i,j}\approx v_{s,i,j}$.

The difference formulas (6.51) and (6.6) are substituted into (6.45) and we get difference equations of transverse vibration for $s=1,\dots,R$, $i=2,\dots,N-2$, $j=1,\dots,M-1$:

$$\begin{aligned} &E\,J\frac{V_{s,i+2,j}-4V_{s,i+1,j}+6V_{s,i,j}-4V_{s,i-1,j}+V_{s,i-2,j}}{d^4}+\mu\frac{V_{s,i,j+1}-2V_{s,i,j}+V_{s,i,j-1}}{\tau^2}\\ &=\quad \mathbb{T}(s,i)e_i+g_{s,i,j}+h_{s,i,j}. \end{aligned} \tag{6.52}$$

The substitution $r=\frac{EJ\tau^2}{\mu d^4}$ is introduced in (6.52) and we obtain the relation

$$\begin{aligned} V_{s,i,j+1}=&(2-6r)V_{s,i,j}+r(-V_{s,i+2,j}+4V_{s,i+1,j}+4V_{s,i-1,j}-V_{s,i-2,j})-V_{s,i,j-1}+\\ &+\frac{\tau^2}{\mu}(\mathbb{T}(s,i)e_i+g_{s,i,j}+h_{s,i,j}). \end{aligned} \tag{6.53}$$

The equation for $i=0$ is given by the first boundary condition (6.46):

$$V_{s,0,j}=0,\ s=1,\dots,R,\ j=1,\dots,M-1. \tag{6.54}$$

For constructing the equations for $i=1$, $i=N-1$ and $i=N$ we introduce the fictitious grid points x_{-1}, x_{N+1}, x_{N+2}. The values in these nodes are unknown because they lie outside the region $\langle 0,l\rangle\times\langle 0,T\rangle$. However, we can use the numerical differentiation formulas:

$$\frac{V_{s,1,j}-V_{s,-1,j}}{2d}=0,$$

$$\frac{V_{s,N+1,j} - 2V_{s,N,j} + V_{s,N-1,j}}{d^2} = 0,$$
$$\frac{V_{s,N+2,j} - 2V_{s,N+1,j} + 2V_{s,N-1,j} - V_{s,N-2,j}}{2d^3} = 0,$$

which approximate the second boundary condition in (6.46) and both conditions in (6.47). Hence, we obtain the approximations $V_{s,-1,j} = V_{s,1,j}$, $V_{s,N+1,j} = 2V_{s,N,j} - V_{s,N-1,j}$ and $V_{s,N+2,j} = 4V_{s,N,j} - 4V_{s,N-1,j} + V_{s,N-2,j}$ with order of accuracy $\mathcal{O}(d^2)$.

When these approximations are used in (6.53) for $i = 1$, $i = N - 1$ and $i = N$, it results in:

$$\begin{aligned} V_{s,1,j+1} &= (2-7r)V_{s,1,j} + r(-V_{s,3,j} + 4V_{s,2,j}) - V_{s,1,j-1} + \\ &+ \frac{\tau^2}{\mu}(\mathbb{T}(s,1)e_1 + g_{s,1,j} + h_{s,1,j}), \end{aligned} \tag{6.55}$$

$$\begin{aligned} V_{s,N-1,j+1} &= (2-5r)V_{s,N-1,j} + r(2V_{s,N,j} + 4V_{s,N-2,j} - V_{s,N-3,j}) - V_{s,N-1,j-1} + \\ &+ \frac{\tau^2}{\mu}(\mathbb{T}(s,N-1)e_{N-1} + g_{s,N-1,j} + h_{s,N-1,j}), \end{aligned} \tag{6.56}$$

$$\begin{aligned} V_{s,N,j+1} &= (2-2r)V_{s,N,j} + r(4V_{s,N-1,j} - 2V_{s,N-2,j}) - V_{s,N,j-1} + \\ &+ \frac{\tau^2}{\mu}(\mathbb{T}(s,N)e_N + g_{s,N,j} + h_{s,N,j}). \end{aligned} \tag{6.57}$$

Values corresponding to $j = 0$ and $j = 1$ must be supplied in order to compute the values for $j = 2$. It is made by using the initial conditions (6.48) and the Taylor formula of order two.

$$V_{s,i,0} = \varphi_i,\ i = 1,\dots,N,\ s = 1,\dots,R, \tag{6.58}$$

$$V_{s,i,1} = \varphi_i + \psi_i\tau - \frac{rd^4}{2}\left.\frac{\partial^4\varphi}{\partial x^4}\right|_{x_i} + \frac{\tau^2}{2\mu}(\mathbb{T}(s,i)e_i + g_{s,i,0} + h_{s,i,0}),\ i = 1,\dots,N,\ s = 1,\dots,R. \tag{6.59}$$

The initial displacement $\varphi(x)$ must fulfill the same boundary conditions as $v(\xi, x, t)$, therefore, we use fictitious grid points and analogical derivation of the equations for the outside grid points while approximating $\frac{\partial^4\varphi}{\partial x^4}$.

The objective function is again approximated by the composite Simpson's rule with coefficients $a_0 = 1 = a_N$, $a_{2k-1} = 4$, $a_{2k} = 2$, $k = 1, 2, \dots, \frac{N}{2}$, $b_0 = 1 = b_M$, $b_{2m-1} = 4$, $b_{2m} = 2$, $m = 1, 2, \dots, \frac{M}{2}$ and N, M are even numbers.

Finally, the continuous two-stage stochastic quadratic program (6.44)-(6.50) is approximated by a large deterministic quadratic program:

$$\min_{\mathbf{e},\mathbf{g}_s,\mathbf{V}_s} \sum_{s=1}^{R}\sum_{j=0}^{M}\sum_{i=0}^{N} p_s\frac{d\tau}{9}a_ib_j(V_{s,i,j} - u_{i,j})^2 \tag{6.60}$$

$$\text{s. t.}\quad \mathbf{V}_{s,1} = \mathbf{\Phi} + \tau\mathbf{\Psi} + \frac{1}{2}\mathbb{K}_1\mathbf{\Phi} + \frac{\tau^2}{2\mu}\mathbf{F}_{s,0},\ s = 1,\dots,R, \tag{6.61}$$

$$\mathbf{V}_{s,j+1} = \mathbb{K}\mathbf{V}_{s,j} - \mathbf{V}_{s,j-1} + \frac{\tau^2}{\mu}\mathbf{F}_{s,j},\ j = 1\dots,M-1,\ s = 1,\dots,R, \tag{6.62}$$

$$V_{s,0,j} = 0,\ j = 0,\dots,M,\ s = 1,\dots,R, \tag{6.63}$$

$$V_{s,i,0} = \varphi_i,\ i = 0,\dots,N,\ s = 1,\dots,R, \tag{6.64}$$

$$g_{s,i,0} = 0,\ i = 0,\dots,N,\ s = 1,\dots,R, \tag{6.65}$$

$$|V_{s,i,j}| \le v_{limit},\ i = 0,\dots,N,\ j = 0,\dots,M,\ s = 1,\dots,R, \tag{6.66}$$

where $\mathbb{K}_1 = \begin{pmatrix} -7r & 4r & -r & 0 & 0 & \dots & 0 \\ 4r & -6r & 4r & -r & 0 & \dots & 0 \\ -r & 4r & -6r & 4r & -r & \dots & 0 \\ & & & \vdots & & & \\ 0 & \dots & -r & 4r & -6r & 4r & -r \\ 0 & \dots & 0 & -r & 4r & -5r & 2r \\ 0 & \dots & 0 & 0 & -2r & 4r & -2r \end{pmatrix}$ is a square matrix of order N,

$\mathbf{\Phi} = \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_N \end{pmatrix}$, $\mathbf{\Psi} = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_N \end{pmatrix}$, $\mathbb{K} = \begin{pmatrix} 2-7r & 4r & -r & 0 & 0 & \dots & 0 \\ 4r & 2-6r & 4r & -r & 0 & \dots & 0 \\ -r & 4r & 2-6r & 4r & -r & \dots & 0 \\ & & & \vdots & & & \\ 0 & \dots & -r & 4r & 2-6r & 4r & -r \\ 0 & \dots & 0 & -r & 4r & 2-5r & 2r \\ 0 & \dots & 0 & 0 & -2r & 4r & 2-2r \end{pmatrix}$

is a square matrix of order N, $\mathbf{F}_{s,j} = \begin{pmatrix} \mathbf{T}(s,1)e_1 + g_{s,1,j} + h_{s,1,j} \\ \vdots \\ \mathbf{T}(s,N)e_N + g_{s,N,j} + h_{s,N,j} \end{pmatrix}$.

Vector $\mathbf{V}_{s,j} = (V_{s,1,j}, \dots, V_{s,N,j})^T$, $j = 0, \dots, M$, $s = 1, \dots, R$, is the approximation of $v(\xi, x, t)$. Vector $\mathbf{g}_{s,j} = (g_{s,0,j}, \dots, g_{s,N,j})^T$, $j = 0, \dots, M-1$, $s = 1, \dots, R$, is the approximation of of the second-stage decision variable $g(\xi, x, t)$ and vector $\mathbf{e} = (e_1, \dots, e_{N-1})^T$ is the approximation of the first-stage decision variable $e(x)$.

Deterministic quadratic program is implemented and solved in GAMS with solver CONOPT. We consider a steel beam with square cross section with dimensions $a = b = = 6 \cdot 10^{-3}\,\mathrm{m}$, i. e. the second moment of the cross section is $J = \frac{ab^3}{12} \doteq 10^{-10}\,\mathrm{m}^4$, the Young's modulus is $E = 200 \cdot 10^9\,\mathrm{Pa}$, density is $\rho = 7850\,\mathrm{kg\,m}^{-3}$, $\mu = \rho ab \doteq 0.3\,\mathrm{kg\,m}^{-1}$. The length of console is $l = 1\,\mathrm{m}$, the observation time is $T = 0.1\,\mathrm{s}$ and the displacement limitation is $v_{limit} = 10\,\mathrm{m}$. Numbers of discretization steps are $N = 10$, $M = 10$, number of scenarios R varies from 5 to 2000. Required displacement is $u(x,t) = 0$. The initial displacement is chosen as the second characteristic state of vibrations because it explicitly satisfies the boundary conditions. Therefore, $\varphi(x) = \cosh\frac{\lambda x}{l} + k\sinh\frac{\lambda x}{l} - \cos\frac{\lambda x}{l} - k\sin\frac{\lambda x}{l}$, where $k = -\frac{\cosh\lambda + \cos\lambda}{\sinh\lambda + \sin\lambda}$, $1 + \cosh\lambda\cos\lambda = 0$, $\lambda = l\sqrt{\omega} \doteq 4.694$. The initial velocity is set to $\psi(x) = 0$. Vector $\mathbf{T}_1(x)$ is generated from the uniform distribution $U(0,1)$ and $\xi \sim U(-5,2)$. The external stochastic load $h(\xi, x, t)$ is chosen in the same form as in the previous subsection.

Input data and results, i. e. the external stochastic load, computed first-stage and second-stage variables and displacement, are displayed in the figures 6.11-6.14 for model with only 20 scenarios because of the clarity. But the numerical results concerning solution quality and VSS are computed for model with 1000 scenarios. Program ran on a notebook with Intel Core 2Duo 2GHz and 2GB RAM. Numbers of variables and constraints together with CPU times are given in the table 6.2 for EO and EV deterministic reformulations.

The relevancy of including randomness into model is described by the value of stochastic solution (see definition 5.2.7). In our case for 1000 scenarios, we obtain $z^{EO} = 0.014$, $z^{EV} = 0.003$, $EEV = 0.032$ and $VSS = 0.018$. Hence, the optimal objective function value is more than twice better in case of including randomness into model as in EO reformulation, see program (6.60)-(6.66), than in case of replacing the random variable by its mean value as in EV reformulation.

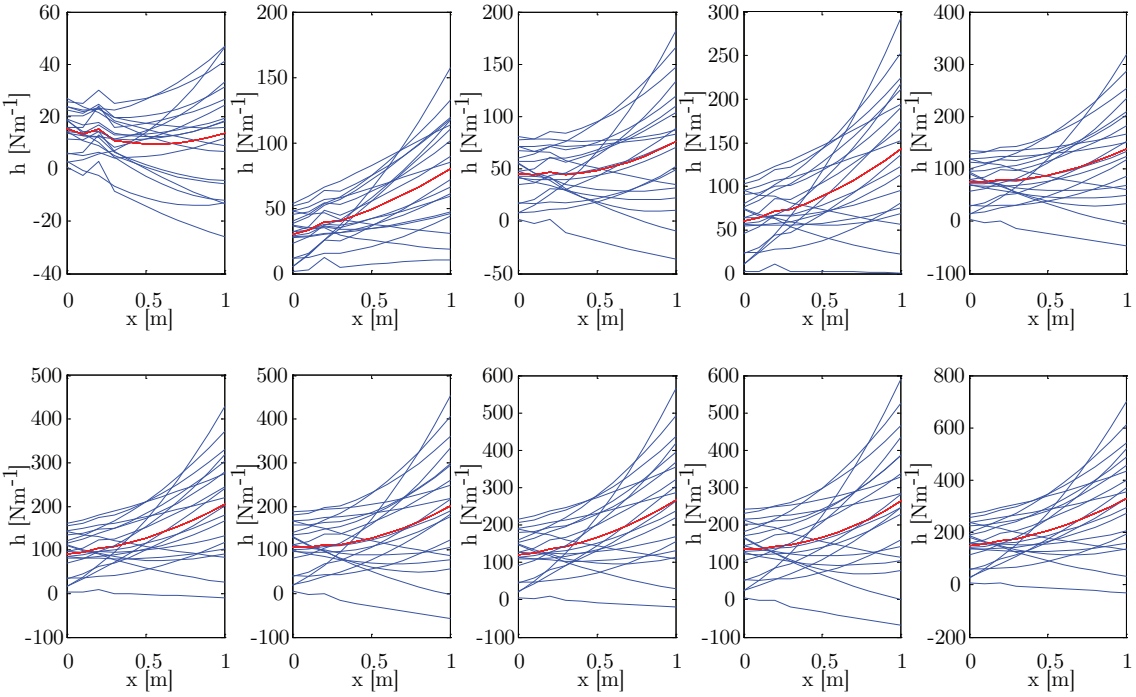


Figure 6.11: Stochastic load $h(\xi, x, t)$. Red line – EV model, blue lines – EO model.

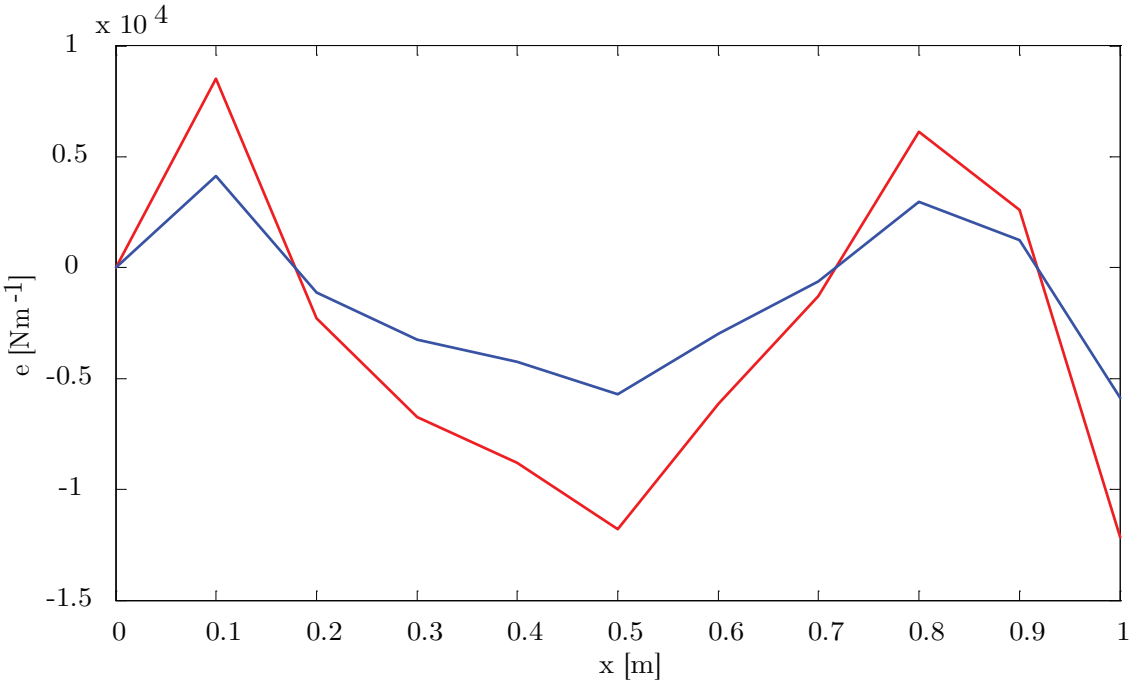


Figure 6.12: First-stage variable $e(x)$. Red line – EV model, blue lines – EO model.

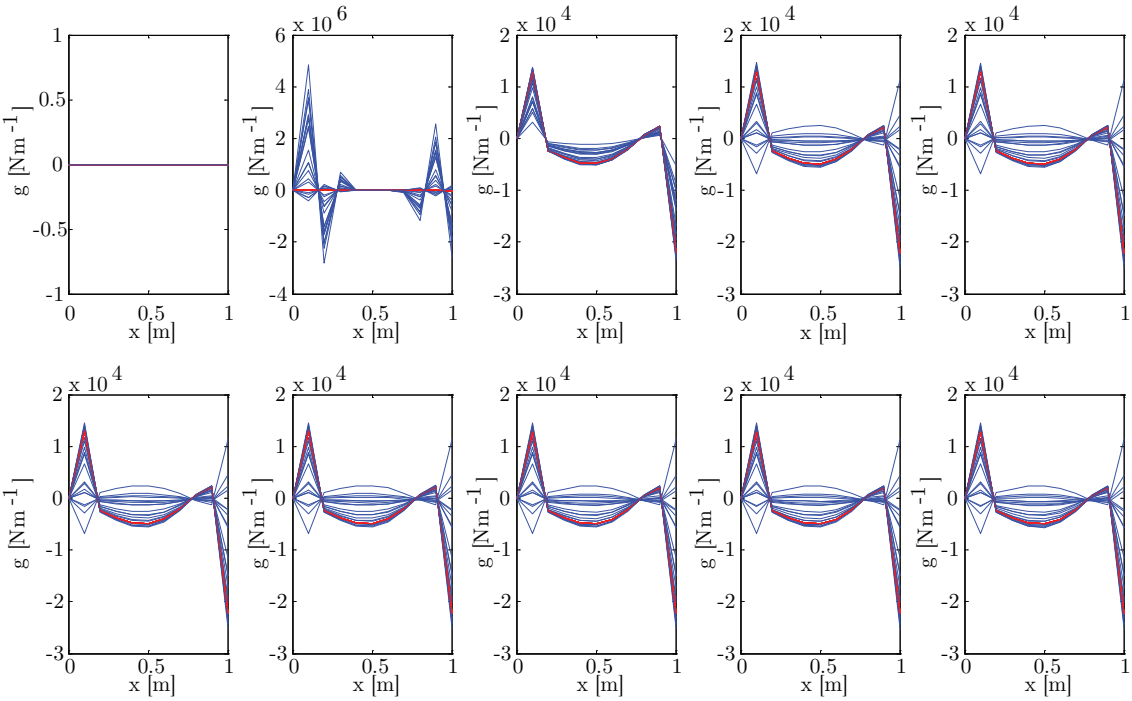


Figure 6.13: Second-stage variable $g(\xi, x, t)$. Red line – EV model, blue lines – EO model.

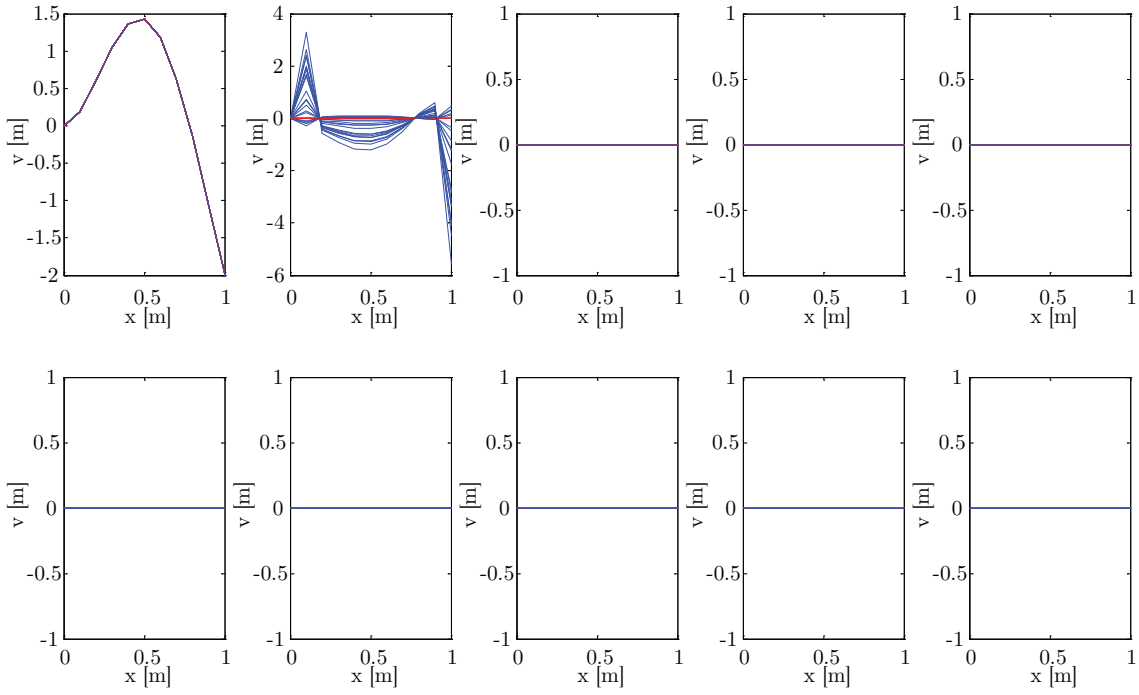


Figure 6.14: Displacement $v(\xi, x, t)$. Red line – EV model, blue lines – EO model.

No. of scenarios R	20	1000	2000	**Type of reformulation**
No. of variables	4451	222 011	444 011	EO reformulation
No. of constraints	2661	133 001	266 001	
CONOPT time [s]	0.1	69.4	154.1	
No. of variables	365	365	365	EV reformulation
No. of constraints	266	266	266	
CONOPT time [s]	0.01	0.01	0.01	

Table 6.2: Computational properties of mathematical programs.

6.2 Relationship between EO and EV solution of the two-stage stochastic quadratic program

Both models in previous section are described by the two-stage stochastic program with quadratic second-stage objective function. Therefore, we are interested in more theoretical properties of this type of models, especially in a relationship between technology matrix and optimality conditions for the first-stage variable in case of EO and EV reformulations.

Let us assume the scenario-based two-stage stochastic program with quadratic second-stage objective function:

$$\min_{\mathbf{x},\mathbf{y}_s} \mathbf{c}^T\mathbf{x} + \sum_{s\in S} p_s(\mathbf{y}_s - \mathbf{u})^T(\mathbf{y}_s - \mathbf{u}) \tag{6.67}$$

$$\text{s.t. } \mathbb{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}, \tag{6.68}$$

$$\mathbb{T}_s\mathbf{x} + \mathbb{W}_s\mathbf{y}_s = \mathbf{h}_s,\ \mathbf{y}_s \geq \mathbf{0}\ \forall s \in S. \tag{6.69}$$

Now let's simplify the previous program in the following way: in (6.67) the cost vector $\mathbf{c} = \mathbf{0}$, $\mathbf{u} = \mathbf{0}$, remove the first stage constraints (6.68) and in (6.69) suppose $\mathbb{T}_s = \mathbb{I}$, $\mathbb{W}_s = \mathbb{W}$ and remove the nonnegativity constraints for both $\mathbf{x}$ and $\mathbf{y}_s$.

Theorem 6.2.1. *Let ξ is a random variable with discrete probability distribution with finite number of scenarios ξ_s, $s \in S$, where p_s are the corresponding probabilities. Let*

$$\min_{\mathbf{x},\mathbf{y}_s} \sum_{s\in S} p_s\mathbf{y}_s^T\mathbf{y}_s \tag{6.70}$$

$$\text{s.t. } \mathbf{x} + \mathbb{W}\mathbf{y}_s = \mathbf{h}_s\ \forall s \in S \tag{6.71}$$

is an EO reformulation of a two-stage stochastic program with quadratic second-stage objective function, identity technology matrix and deterministic recourse matrix, where $\mathbf{x}$ is a first-stage decision vector and $\mathbf{y}_s$ is a second-stage decision vector. Let

$$\min_{\mathbf{x},\mathbf{y}} \mathbf{y}^T\mathbf{y} \tag{6.72}$$

$$\text{s.t. } \mathbf{x} + \mathbb{W}\mathbf{y} = \mathbb{E}(\mathbf{h}) \tag{6.73}$$

is an EV reformulation of a two-stage stochastic program with quadratic second-stage objective function, identity technology matrix and deterministic recourse matrix, where $\mathbf{x}$ is a first-stage decision vector and $\mathbf{y}$ is a second-stage decision vector.
Then the first-stage optimal solutions of EO and EV reformulations are the same.

Proof. The corresponding Lagrange function L of the program (6.70)-(6.71) with Lagrange coefficients $\boldsymbol{\lambda}_s$ is:

$$L(\mathbf{x}, (\mathbf{y}_s)_{s\in S}, (\boldsymbol{\lambda}_s)_{s\in S}) = \sum_{s\in S} p_s \mathbf{y}_s^T \mathbf{y}_s + \sum_{s\in S} (\mathbf{x} + \mathbb{W}\mathbf{y}_s - \mathbf{h}_s)^T \boldsymbol{\lambda}_s.$$

KKT conditions [3] lead to the following equations:

$$\sum_{s\in S} \boldsymbol{\lambda}_s = \mathbf{0}, \tag{6.74}$$

$$2p_s\mathbf{y}_s + \mathbb{W}^T\boldsymbol{\lambda}_s = \mathbf{0}\ \forall s \in S, \tag{6.75}$$

$$\mathbf{x} + \mathbb{W}\mathbf{y}_s - \mathbf{h}_s = \mathbf{0}\ \forall s \in S. \tag{6.76}$$

From (6.75) we get: $\mathbf{y}_s = -\frac{1}{2p_s}\mathbb{W}^T\boldsymbol{\lambda}_s\ \forall s \in S$ and after substitution into (6.76) we obtain: $-\frac{1}{2p_s}\mathbb{W}\mathbb{W}^T\boldsymbol{\lambda}_s = \mathbf{h}_s - \mathbf{x}\ \forall s \in S$. We take convex combination of these equations and obtain: $-\sum_{s\in S}\frac{1}{2}\mathbb{W}\mathbb{W}^T\boldsymbol{\lambda}_s = \sum_{s\in S} p_s(\mathbf{h}_s - \mathbf{x})$ and from (6.74) we have: $\sum_{s\in S} p_s(\mathbf{h}_s - \mathbf{x}) = \mathbf{0}$. Therefore the optimality condition describing the optimal first-stage solution $\mathbf{x}$ for EO reformulation is:

$$\mathbf{x} = \sum_{s\in S} p_s\mathbf{h}_s = \mathbb{E}(\mathbf{h}). \tag{6.77}$$

Now we take EV reformulation, where we replace random variable by its mean value. The corresponding Lagrange function L of the program (6.72)-(6.73) with Lagrange coefficients $\boldsymbol{\lambda}$ is now:

$$L(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = \mathbf{y}^T\mathbf{y} + (\mathbf{x} + \mathbb{W}\mathbf{y} - \mathbb{E}(\mathbf{h}))^T\boldsymbol{\lambda}.$$

KKT conditions give us the following equations:

$$\boldsymbol{\lambda} = \mathbf{0}, \tag{6.78}$$

$$2\mathbf{y} + \mathbb{W}^T\boldsymbol{\lambda} = \mathbf{0}, \tag{6.79}$$

$$\mathbf{x} + \mathbb{W}\mathbf{y} - \mathbb{E}(\mathbf{h}) = \mathbf{0}. \tag{6.80}$$

We substitute (6.78) into (6.79) and we obtain $\mathbf{y} = \mathbf{0}$. From (6.80) we get the optimality condition describing the optimal first-stage solution $\mathbf{x}$ for EV reformulation:

$$\mathbf{x} = \mathbb{E}(\mathbf{h}). \tag{6.81}$$

Comparing (6.77) with (6.81), EO and EV reformulations have the same first-stage optimal solutions. □

Therefore, EEV value is the same as the value z^{EO} and hence, $VSS = 0$ for this type of problems. It is useless to solve EO reformulation and it is sufficient to solve simpler EV reformulation. This result is valid also for nonzero cost vector $\mathbf{c}$.

But the following theorem holds for the random technology matrix $\mathbb{T}$.

Theorem 6.2.2. *Let ξ is a random variable with discrete probability distribution with finite number of scenarios ξ_s, $s \in S$, where p_s are the corresponding probabilities. Let*

$$\min_{\mathbf{x},\mathbf{y}_s} \sum_{s\in S} p_s\mathbf{y}_s^T\mathbf{y}_s \tag{6.82}$$

$$\text{s. t. } \mathbb{T}_s\mathbf{x} + \mathbb{W}\mathbf{y}_s = \mathbf{h}_s\ \forall s \in S \tag{6.83}$$

is an EO reformulation of a two-stage stochastic program with quadratic second-stage objective function, random technology matrix and deterministic recourse matrix, where $\mathbf{x}$ *is a first-stage decision vector and* $\mathbf{y}_s$ *is a second-stage decision vector. Let*

$$\min_{\mathbf{x},\mathbf{y}} \mathbf{y}^T\mathbf{y} \tag{6.84}$$

$$\text{s. t. } \mathbb{E}(\mathbb{T})\mathbf{x} + \mathbb{W}\mathbf{y} = \mathbb{E}(\mathbf{h}) \tag{6.85}$$

is an EV reformulation of a two-stage stochastic program with quadratic second-stage objective function, random technology matrix and deterministic recourse matrix, where $\mathbf{x}$ *is a first-stage decision vector and* $\mathbf{y}$ *is a second-stage decision vector.*
Then the sets of the first-stage optimal solutions of EO and EV reformulations have empty intersections.

Proof. The corresponding Lagrange function L of the program (6.82)-(6.83) with Lagrange coefficients $\boldsymbol{\lambda}_s$ is:

$$L(\mathbf{x}, (\mathbf{y}_s)_{s\in S}, (\boldsymbol{\lambda}_s)_{s\in S}) = \sum_{s\in S} p_s \mathbf{y}_s^T\mathbf{y}_s + \sum_{s\in S} (\mathbb{T}_s\mathbf{x} + \mathbb{W}\mathbf{y}_s - \mathbf{h}_s)^T\boldsymbol{\lambda}_s.$$

KKT conditions lead to the following equations:

$$\sum_{s\in S} \mathbb{T}_s^T\boldsymbol{\lambda}_s = \mathbf{0}, \tag{6.86}$$

$$2p_s\mathbf{y}_s + \mathbb{W}^T\boldsymbol{\lambda}_s = \mathbf{0}\ \forall s \in S, \tag{6.87}$$

$$\mathbb{T}_s\mathbf{x} + \mathbb{W}\mathbf{y}_s - \mathbf{h}_s = \mathbf{0}\ \forall s \in S. \tag{6.88}$$

We take (6.88) times $2p_s$ and subtract it from (6.87) times $\mathbb{W}$. The results is:

$$\mathbb{W}\mathbb{W}^T\boldsymbol{\lambda}_s - 2p_s(\mathbb{T}_s\mathbf{x} - \mathbf{h}_s) = \mathbf{0}\ \forall s \in S$$

and if the matrix $\mathbb{W}\mathbb{W}^T$ is regular we can write:

$$\boldsymbol{\lambda}_s = 2p_s(\mathbb{W}\mathbb{W}^T)^{-1}(\mathbb{T}_s\mathbf{x} - \mathbf{h}_s)\ \forall s \in S.$$

After substitution into (6.86) we get the optimality condition for EO reformulation:

$$\sum_{s\in S} p_s\mathbb{T}_s^T(\mathbb{W}\mathbb{W}^T)^{-1}\mathbb{T}_s\mathbf{x} = \sum_{s\in S} p_s\mathbb{T}_s^T(\mathbb{W}\mathbb{W}^T)^{-1}\mathbf{h}_s. \tag{6.89}$$

Now we take EV reformulation where we replace random variable by its mean value. The corresponding Lagrange function L of the program (6.84)-(6.85) with Lagrange coefficients $\boldsymbol{\lambda}$ is now:

$$L(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = \mathbf{y}^T\mathbf{y} + (\mathbb{E}(\mathbb{T})\mathbf{x} + \mathbb{W}\mathbf{y} - \mathbb{E}(\mathbf{h}))^T\boldsymbol{\lambda}.$$

KKT conditions give us the following equations:

$$\mathbb{E}(\mathbb{T})^T\boldsymbol{\lambda} = \mathbf{0}, \tag{6.90}$$

$$2\mathbf{y} + \mathbb{W}^T\boldsymbol{\lambda} = \mathbf{0}, \tag{6.91}$$

$$\mathbb{E}(\mathbb{T})\mathbf{x} + \mathbb{W}\mathbf{y} - \mathbb{E}(\mathbf{h}) = \mathbf{0}. \tag{6.92}$$

We take (6.92) times 2 and subtract it from (6.91) times $\mathbb{W}$. The results is:

$$\mathbb{W}\mathbb{W}^T\boldsymbol{\lambda} - 2(\mathbb{E}(\mathbb{T})\mathbf{x} - \mathbb{E}(\mathbf{h}) = \mathbf{0}\ \forall s \in S$$

and if the matrix $\mathbb{W}\mathbb{W}^T$ is regular we can write:

$$\boldsymbol{\lambda} = 2(\mathbb{W}\mathbb{W}^T)^{-1}(\mathbb{E}(\mathbb{T})\mathbf{x} - \mathbb{E}(\mathbf{h})\ \forall s \in S.$$

After substitution into (6.90) we get the optimality condition for EV reformulation:

$$\mathbb{E}(\mathbb{T})^T(\mathbb{W}\mathbb{W}^T)^{-1}\mathbb{E}(\mathbb{T})\mathbf{x} = \mathbb{E}(\mathbb{T})^T(\mathbb{W}\mathbb{W}^T)^{-1}\mathbb{E}(\mathbf{h}). \tag{6.93}$$

This equation is equivalent to the following one:

$$\sum_{s\in S} p_s\mathbb{T}_s^T(\mathbb{W}\mathbb{W}^T)^{-1}\sum_{s\in S} p_s\mathbb{T}_s\mathbf{x} = \sum_{s\in S} p_s\mathbb{T}_s^T(\mathbb{W}\mathbb{W}^T)^{-1}\sum_{s\in S} p_s\mathbf{h}_s. \tag{6.94}$$

The optimality conditions ((6.89) and (6.94)) give different first-stage optimal solutions $\mathbf{x}$ for EO and EV reformulations. □

Therefore, EEV value is different from the value z^{EO} and hence, $VSS \neq 0$ for this type of problems. It means that the problems with the random technology matrix $\mathbb{T}$ should not be solved by simple EV reformulation and more computationally intensive EO reformulation should be solved to obtain reasonable and reliable results.

6.3 ODE constrained stochastic programming models

In addition to the aims of the Ph. D. thesis from chapter 2, where we assumed dynamic engineering optimization problems leading to PDE constrained stochastic programs, we have also studied static optimization problems constrained by ODE.

6.3.1 Stochastic optimization of design of beam cross section dimensions – variant I

An optimization problem from the area of civil engineering describing deflection of a beam[2] has been chosen on the recommendation of specialists dealing with similar problems, e. g. [42]. The aim of the optimization is to obtain an optimal design of beam cross section dimensions while weight is minimized (6.95), rigidity is maximized (6.96) and deflection

[2]We assume that the dimensions of the constant cross section are substantially smaller than the length, i. e. the beam is modelled by the prismatic bar and the ODE describes the deflection of the centreline.

is minimized (6.97), see the following model and figure 6.15.

$$\min\ \rho abl \tag{6.95}$$

$$\max\ \frac{E(\xi)ab^3}{12} \tag{6.96}$$

$$\min\ v(\xi, x) \tag{6.97}$$

$$\text{s.t. } E(\xi)\frac{ab^3}{12}\frac{\mathrm{d}^4 v}{\mathrm{d}x^4}(\xi, x) = h(x),\ x \in \langle 0, l\rangle,\ \xi \in \Xi, \tag{6.98}$$

$$v(\xi, 0) = 0,\ \frac{\mathrm{d}v}{\mathrm{d}x}(\xi, 0) = 0,\ \xi \in \Xi, \tag{6.99}$$

$$v(\xi, l) = 0,\ \frac{\mathrm{d}v}{\mathrm{d}x}(\xi, l) = 0,\ \xi \in \Xi, \tag{6.100}$$

$$\left|E(\xi)\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}(\xi, x)\frac{b}{2}\right| \leq \sigma_{limit},\ x \in \langle 0, l\rangle,\ \xi \in \Xi, \tag{6.101}$$

$$a_{min} \leq a \leq a_{max}, \tag{6.102}$$

$$b_{min} \leq b \leq b_{max}, \tag{6.103}$$

where ρ is the beam density, l is the beam length, x is the related space coordinate, $\xi : \Xi \to \mathbb{R}$ is a random variable, E is random Young's modulus (because of varying uncertain material characteristics [43]), $h(x)$ is a deterministic static load, a, b are decision variables (dimensions of the cross section) and $v(\xi, x)$ is a deflection with the opposite direction than the axis y. The ODE (6.98) describes transverse deflection of the beam[3], boundary conditions for clamped end points given by (6.99) and (6.141) mean that there are zero transverse deflections and their slopes. Furthermore, the maximum stress σ_{max} given as $\sigma_{max}(x) = \frac{M(x)}{J} y_{max} = \pm E\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}(x)\frac{b}{2}$, where $M(x) = -EJ\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}(x)$ is the bending moment, $J = \frac{ab^3}{12}$ is the second moment of the cross section with respect to the axis z and $y_{max} = \pm\frac{b}{2}$, must be bounded because of safety reasons. Limiting value σ_{limit} relates to the proportional limit which marks the end of the area of elastic behaviour described by Hooke's law where the stress is proportional to the relative deformation [9], see constraint (6.101). Finally, the dimensions of the beam cross section must be bounded, see (6.102) and (6.103).

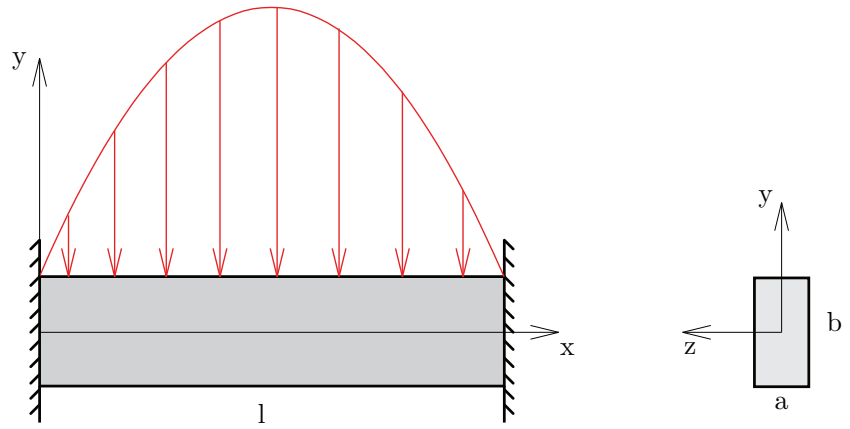


Figure 6.15: Scheme of loaded beam and its cross section with dimensions to optimize.

[3]It is derived on the assumptions that the influence of the shear force on the strain is neglected (i. e. $T = 0$), there are no temperature fluctuations and the cross section is constant.

As it is mentioned above, deterministic reformulation of the underlying program (6.95)-(6.103) have to be made. We will consider EO reformulation which means taking mean value of objective functions (6.95)-(6.97) and almost surely satisfying of constraints (6.98)-(6.101). Hence, we obtain continuous two-stage stochastic nonlinear program with a scheme in the figure 6.16.

$$\min\ \rho abl \tag{6.104}$$

$$\max\ \mathbb{E}\left(\frac{E(\xi)ab^3}{12}\right) \tag{6.105}$$

$$\min\ \mathbb{E}\left(v(\xi,x)\right) \tag{6.106}$$

$$\text{s.t. } E(\xi)\frac{ab^3}{12}\frac{\mathrm{d}^4 v}{\mathrm{d}x^4}(\xi,x) = h(x),\ x\in\langle 0,l\rangle,\ \text{a. e. } \xi\in\Xi, \tag{6.107}$$

$$v(\xi,0)=0,\ \frac{\mathrm{d}v}{\mathrm{d}x}(\xi,0)=0,\ \text{a. e. } \xi\in\Xi, \tag{6.108}$$

$$v(\xi,l)=0,\ \frac{\mathrm{d}v}{\mathrm{d}x}(\xi,l)=0,\ \text{a. e. } \xi\in\Xi, \tag{6.109}$$

$$\left|E(\xi)\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}(\xi,x)\frac{b}{2}\right| \le \sigma_{limit},\ x\in\langle 0,l\rangle,\ \text{a. e. } \xi\in\Xi, \tag{6.110}$$

$$a_{min}\le a\le a_{max}, \tag{6.111}$$

$$b_{min}\le b\le b_{max}. \tag{6.112}$$

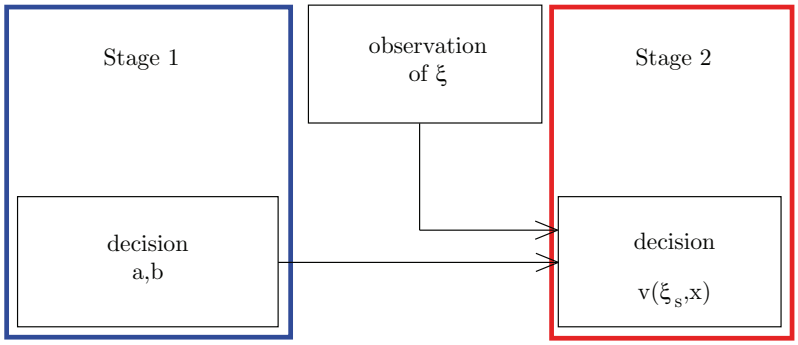


Figure 6.16: Scheme of the two-stage stochastic program (6.104)-(6.112).

As we have mentioned in the previous subsections for PDE constrained programs, the proper approximations must be made, i. e. scenario-based approach for the approximation of random variable and finite difference method with uniform grid spacing for the discretization of the space x coordinate in objective function and constraints are used. The notation is analogous as in the previous subsections with only difference that we omit coordinate t and index j.

The difference formula analogous to (6.51) with omitting index j is substituted into (6.107) and we get difference equations of transverse deflection for $s = 1,\dots,R$ and $i = 2,\dots,N-2$:

$$ab^3E_s(V_{s,i+2} - 4V_{s,i+1} + 6V_{s,i} - 4V_{s,i-1} + V_{s,i-2}) = 12d^4h_i. \tag{6.113}$$

Central difference formula analogous to (6.6) is substituted into (6.110) and discretized versions of ODE describing stress limitation are obtained for $s = 1,\dots,R$ and $i = 1,\dots,N-1$:

$$|bE_s(V_{s,i+1} - 2V_{s,i} + V_{s,i-1})| \le 2d^2\sigma_{limit}. \tag{6.114}$$

The equations for $i = 0$ and $i = N$ are given by the first boundary conditions in (6.108) and (6.109):

$$V_{s,0} = 0,\ V_{s,N} = 0,\ s = 1, \ldots, R. \tag{6.115}$$

For constructing the equations for $i = 1$ and $i = N-1$ the fictitious grid points x_{-1}, x_{N+1} are again introduced. The values in these nodes are obtained by means of the numerical differentiation formulas

$$\frac{V_{s,1} - V_{s,-1}}{2d} = 0,$$
$$\frac{V_{s,N+1} - V_{s,N-1}}{2d} = 0,$$

which approximate the second boundary conditions in (6.108) and (6.109). Hence we obtain the approximations $V_{s,-1} = V_{s,1}$ and $V_{s,N+1} = V_{s,N-1}$ with order of accuracy $\mathcal{O}(d^2)$.

When these approximations are used in (6.113) for $i = 1$ and $i = N - 1$, it results in:

$$ab^3 E_1(V_{s,3} - 4V_{s,2} + 7V_{s,1}) = 12d^4 h_1,$$
$$ab^3 E_{N-1}(7V_{s,N-1} - 4V_{s,N-2} + V_{s,N-3}) = 12d^4 h_{N-1}.$$

After substitution of these approximations into (6.114) for $i = 0$ and $i = N$, we obtain:

$$|bE_1 V_{s,1}| \leq d^2 \sigma_{limit},$$
$$|bE_{N-1} V_{s,N-1}| \leq d^2 \sigma_{limit}.$$

Because we assume discrete probability distribution of random variable ξ with finite number of equiprobable scenarios, the mean values in the objectives (6.105) and (6.106) are computed by summation as in definition 5.1.4. Furthermore, the deflection is always in the direction of the acting load, i. e. $v(\xi, x) \geq 0$ for every scenario of ξ. Therefore, only simple summation of deflection values along the beam length is needed to minimize the deflection.

Hence, the continuous two-stage stochastic nonlinear program (6.104)-(6.112) is approximated by a large multi-objective deterministic nonlinear program

$$\min\ \rho abl \tag{6.116}$$
$$\max \sum_{s=1}^{R} p_s \frac{E_s a b^3}{12} \tag{6.117}$$
$$\min \sum_{s=1}^{R} \sum_{i=0}^{N} p_s V_{s,i} \tag{6.118}$$
$$\text{s. t. } ab^3 \mathbb{K} E_s \mathbf{V}_s = \mathbf{f},\ s = 1, \ldots, R, \tag{6.119}$$
$$V_{s,0} = 0,\ V_{s,N} = 0,\ s = 1, \ldots, R, \tag{6.120}$$
$$|b\mathbb{C} E_s \mathbf{V}_s| \leq d^2 \sigma_{limit} \mathbf{g},\ s = 1, \ldots, R, \tag{6.121}$$
$$a_{min} \leq a \leq a_{max}, \tag{6.122}$$
$$b_{min} \leq b \leq b_{max}, \tag{6.123}$$

where $\mathbb{K} = \begin{pmatrix} 7 & -4 & 1 & 0 & 0 & \ldots & 0 \\ -4 & 6 & -4 & 1 & 0 & \ldots & 0 \\ 1 & -4 & 6 & -4 & 1 & \ldots & 0 \\ & & & \vdots & & & \\ 0 & \ldots & 1 & -4 & 6 & -4 & 1 \\ 0 & \ldots & 0 & 1 & -4 & 6 & -4 \\ 0 & \ldots & 0 & 0 & 1 & -4 & 7 \end{pmatrix}$ is a square matrix of order $N - 1$,

$\mathbf{f} = \begin{pmatrix} 12d^4h_1 \\ \vdots \\ 12d^4h_{N-1} \end{pmatrix}$, $\mathbf{g} = (1, 2, 2, \ldots, 2, 1)^T$ is $(N+1)$-dimensional vector and

$$\mathbb{C} = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ -2 & 1 & 0 & \ldots & 0 \\ 1 & -2 & 1 & \ldots & 0 \\ & & \vdots & & \\ 0 & \ldots & 1 & -2 & 1 \\ 0 & \ldots & 0 & 1 & -2 \\ 0 & \ldots & 0 & 0 & 1 \end{pmatrix}$$ is a matrix of order $(N+1) \times (N-1)$.

The objective function (6.118) can be also written in more compact form as $\min \sum_{s=1}^{R} p_s \mathbf{1}^T \mathbf{V}_s^*$, where $\mathbf{1}^T = (1, 1, \ldots, 1)$ is $(N+1)$-dimensional vector. Vectors $\mathbf{V}_s = (V_{s,1}, \ldots, V_{s,N-1})^T$, respectively $\mathbf{V}_s^* = (V_{s,0}, \ldots, V_{s,N})^T$, are the approximations of $v(\xi, x)$ and $E_s = E(\xi_s)$, $s = 1, \ldots, R$.

For multiple objectives we employ the weighted sum approach typically used in multi-objective optimization [55]. Therefore, our three objectives functions (6.116)-(6.118) are replaced by the following single-objective function:

$$\min_{a,b,\mathbf{V}_s} \left(-\alpha \sum_{s=1}^{R} p_s \frac{E_s a b^3}{12 c_{rigid}} + \beta \frac{\rho a b l}{c_{weight}} + \gamma \sum_{s=1}^{R} \sum_{i=0}^{N} p_s \frac{V_{s,i}}{c_{defl}} \right), \tag{6.124}$$

where α, β, $\gamma > 0$ are the weighting coefficients, $\alpha + \beta + \gamma = 1$ and $c_{rigid}, c_{weight}, c_{defl}$ are typical values of rigidity, weight and deflection of the beam (i. e. normalizing constants). These values are obtained as the optimal values of objective function of three single-objective optimization problems.

The results are presented for the following input data and related formulas. For better scaling we do not compute with SI units but we use units common in engineering computations, i. e. length is given in mm (milimeters), weight is given in t (tons) and stress is given in MPa (megapascals). The load is quadratic: $h(x) = -4h_0 \frac{x^2}{l^2} + 4h_0 \frac{x}{l}$, $h_0 = 20\,\mathrm{Nmm}^{-1}$ (see figure 6.17), the length of steel beam is $l = 1000\,\mathrm{mm}$ with density $\rho = 7.85 \cdot 10^{-9}\,\mathrm{tmm}^{-3}$. The stress limitation is $\sigma_{limit} = 100\,\mathrm{MPa}$. Number of grid points is $N = 50$, we assume $R = 100$ scenarios and bounding values of beam dimensions are $a_{min} = b_{min} = 10\,\mathrm{mm}$, $a_{max} = b_{max} = 100\,\mathrm{mm}$. The weighting coefficients are chosen as: $\alpha = 0.3$; $\beta = 0.45$; $\gamma = 0.25$. The normalizing constants have the following values: $c_{rigidity} = 1.80 \cdot 10^{12}\,\mathrm{Nmm}^2$, $c_{weight} = 0.007\,\mathrm{t}$ and $c_{defl} = 0.7\,\mathrm{mm}$. We assume random Young's modulus: $E_s = 2 \cdot 10^5\,\mathrm{MPa} + E_{random,s}$ where $E_{random,s} \sim U(-1 \cdot 10^4, 5 \cdot 10^4)\,\mathrm{MPa}$. Randomness of Young's modulus E can be caused by different heat–treating process of steel such as normalization, soft annealing, annealing or by different quality of the material.

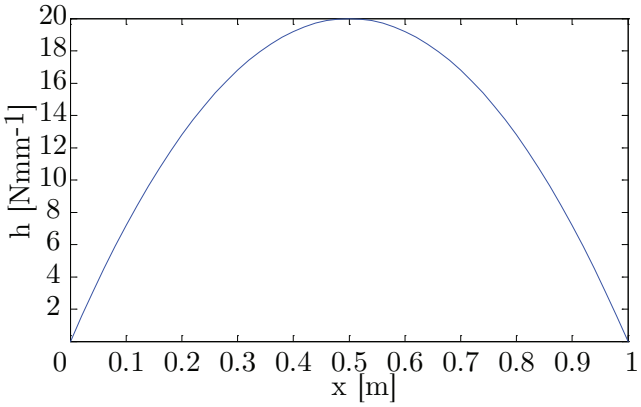


Figure 6.17: Quadratic load $h(x)$.

Program with the objective function (6.124) and constraints (6.119)-(6.123) is implemented in GAMS with solver CONOPT and ran on a notebook with Intel Core 2Duo 2GHz and 2GB RAM. Numbers of variables and constraints together with CPU times are given in the table 7.3 for EO and EV deterministic reformulations.

The optimal objective function value is $z = 2.13$. The optimal dimensions are $a = 22.4\,\text{mm}$, $b = 100\,\text{mm}$ and we use it as a candidate solution $\bar{\mathbf{a}} = (a, b)^T$ for Monte Carlo bounding technique described in the following chapter. The fact, that the optimal shape of the beam cross section is a rectangle with longer side parallel and shorter side perpendicular to direction of load, is in agreement with physical assumptions.

No. of scenarios R	20	100	1000	**Type of reformulation**
No. of variables	1023	5103	51 003	EO reformulation
No. of constraints	3061	15 301	153 001	
CONOPT time [s]	0.7	7.4	31.7	
No. of variables	54	54	54	EV reformulation
No. of constraints	154	154	154	
CONOPT time [s]	0.03	0.03	0.03	

Table 6.3: Computational properties of mathematical programs.

Deflection and maximum stress are presented in the figure 6.18 for model with only 20 scenarios because of the clarity. Maximum deflection of about 0.13 mm occurs in the middle of the beam while it decreases towards the beam's ends. Maximum tensile stress of about 35 MPa occurs at the ends of the beam while maximum compression stress of about $-20\,\text{MPa}$ occurs in the middle of the beam. Maximum stress reaches the same values for both reformulations (see figure 6.18 b)), i. e. its values does not depend on the realization of random variable ξ. This fact is in agreement with the underlying physics. Maximum stress can be also determined analytically without computing $\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}(\xi, x)$ and we will show that it depends neither on deflection $v(\xi, x)$ nor on the realization of random variable ξ.

The bending moment $M(x)$ and the normal force $N(x)$ are given as

$$M(x) = M_B - \int_0^x h(y)(x - y)\,\mathrm{d}y + F_{By}x,$$

$$N(x) = -F_{Bx},$$

where $h(x)$ is the external deterministic load, M_B is a reaction moment and F_{Bx}, F_{By} are reaction forces, see figure 6.19. We suppose that the influence of the shear force on the strain is neglected ($T(x) = 0$).

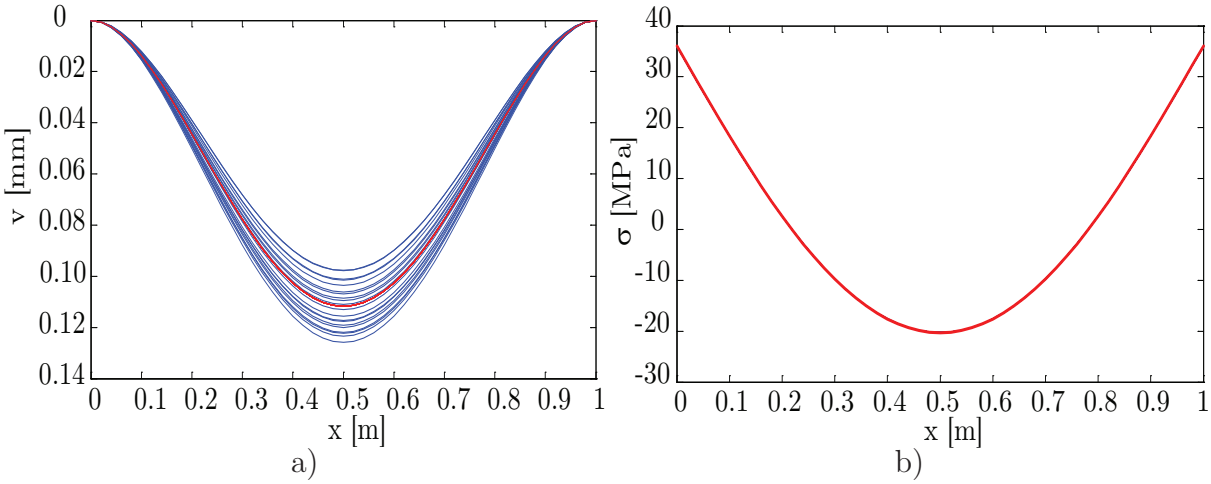


Figure 6.18: a) Deflection $v(\xi, x)$. Red line – EV model, blue lines – EO model, b) Maximum stress (red and blue lines coincide).

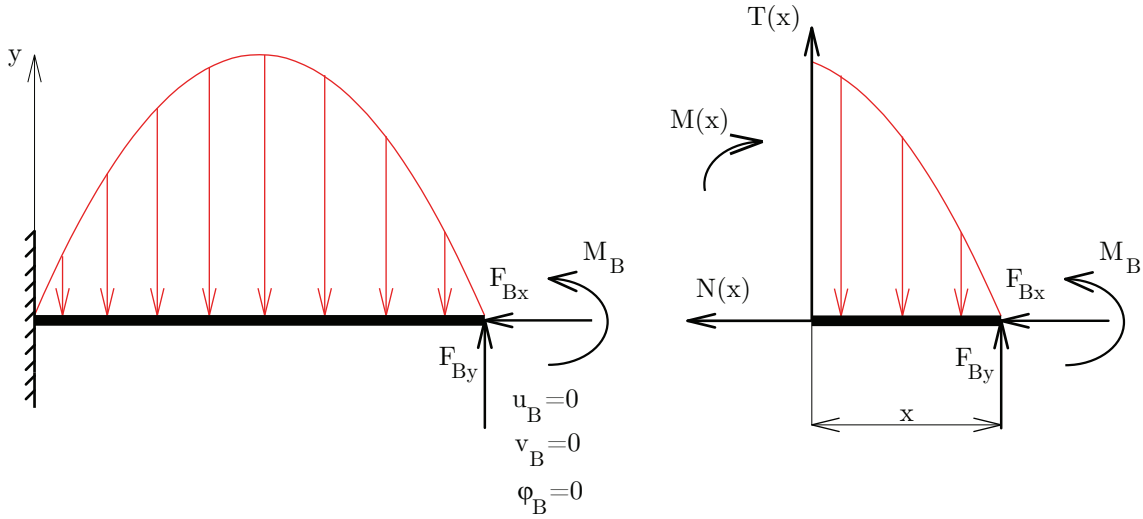


Figure 6.19: Creating a released structure and introducing the inner stress resultants. u and v mean transverse and longitudinal displacements and φ means slope.

In our case with quadratic load, we obtain after integration

$$M(x) = M_B - \frac{2h_0x^3}{3l} + \frac{h_0x^4}{3l^2} + F_{By}x. \tag{6.125}$$

Furthermore, according to the Castigliano's theorem [24] the following equations have to be fulfilled.

$$u_B = \int_0^x \frac{M(x)}{EJ}\frac{\partial M(x)}{\partial F_{Bx}}\,\mathrm{d}x + \int_0^x \frac{N(x)}{ES}\frac{\partial N(x)}{\partial F_{Bx}}\,\mathrm{d}x = 0,$$

$$v_B = \int_0^x \frac{M(x)}{EJ}\frac{\partial M(x)}{\partial F_{By}}\,\mathrm{d}x + \int_0^x \frac{N(x)}{ES}\frac{\partial N(x)}{\partial F_{By}}\,\mathrm{d}x = 0,$$

$$\varphi_B = \int_0^x \frac{M(x)}{EJ}\frac{\partial M(x)}{\partial M_B}\,\mathrm{d}x + \int_0^x \frac{N(x)}{ES}\frac{\partial N(x)}{\partial M_B}\,\mathrm{d}x = 0,$$

where S is an area of the cross section.

It means that we solve these three equations with respect to the unknown values M_B, F_{Bx} and F_{By} and we get $M_B = -\frac{1}{15}h_0l^2$, $F_{Bx} = 0$ and $F_{By} = \frac{1}{3}h_0l$. After substitution these values into (6.125) and subsequent substitution of $M(x)$ into $\sigma_{max}(x) = \pm\frac{6M(x)}{ab^2}$ the following form of the maximum stress is obtained:

$$\sigma_{max}(x) = \pm\frac{2h_0}{ab^2}\left(-\frac{1}{5}l^2 + lx - 2\frac{x^3}{l} + \frac{x^4}{l^2}\right). \tag{6.126}$$

It is obvious that the maximum stress really does not depend on the realization of random variable ξ and the constraint (6.101) can be replaced for deterministic quadratic load $h(x) = -4h_0\frac{x^2}{l^2} + 4h_0\frac{x}{l}$ by another constraint

$$\left|\frac{2h_0}{ab^2}\left(-\frac{1}{5}l^2 + lx - 2\frac{x^3}{l} + \frac{x^4}{l^2}\right)\right| \leq \sigma_{limit},\ x \in \langle 0, l\rangle. \tag{6.127}$$

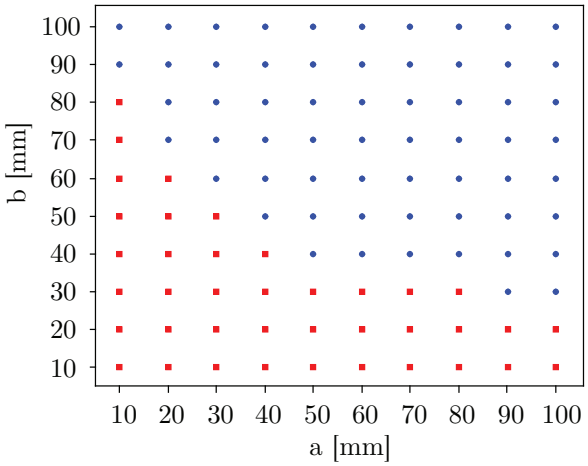


Figure 6.20: Feasibility of the first-stage solutions.

An interesting point is the convexity of the objective function (6.124). Therefore, we solved program (6.124), (6.119)-(6.123) for fixed values of the first-stage variables (dimensions a and b) and check their feasibility to obtain the feasibility set. The intervals for a and b were divided into 10 values, i. e. 100 programs were solved. The feasibility set is presented in the figure 6.20.

The objective function values were computed for feasible first-stage solutions and the function is not convex (see the figure 6.21). Therefore, the obtained local minimum solution may not be the global one. But we hope that this solution is at least suboptimal, i. e. it is better than the existing solution, which is sufficient for the engineers.

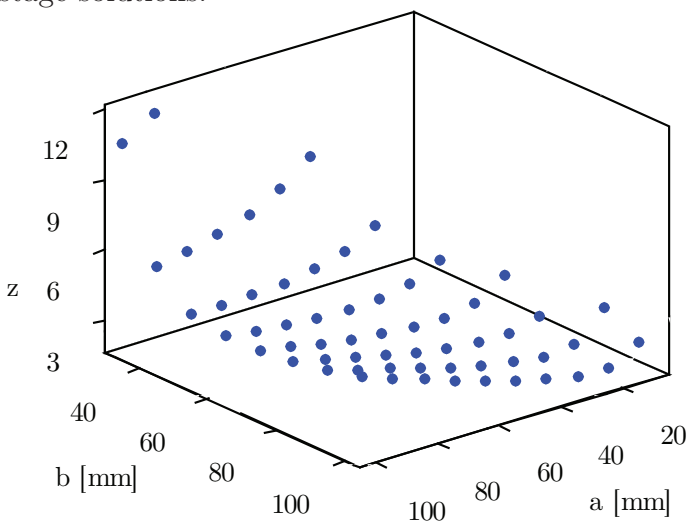


Figure 6.21: Values of objective function for feasible first-stage solutions.

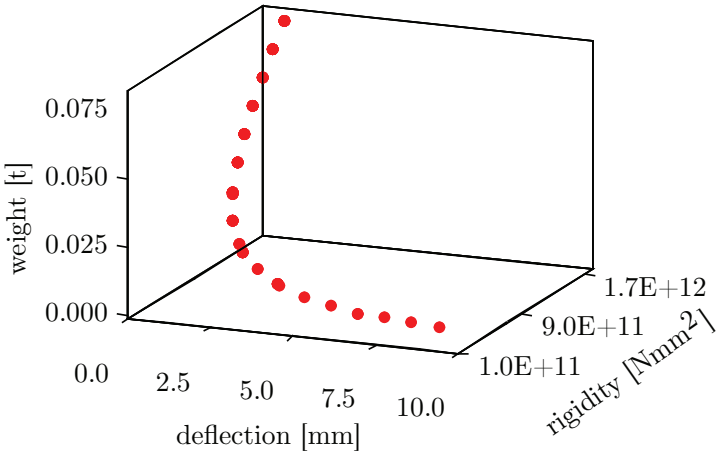
As the next step, we discuss model (6.116)-(6.123) from multi-objective viewpoint where a notion of an optimal solution is replaced by the concepts of efficient points and efficient frontier (see chapter 4). The set of points on the efficient frontier can be computed

by repeated optimization of (6.116)-(6.123) for various relevant values of selected criteria (in our case for various values of the parameters ϑ and ω, see below). New constraints enforcing achievement levels for all but one criterion (see (6.117), (6.118)) are added and the remaining criterion related to (6.116) is treated as a single-objective function. In our case we add the following two constraints:

$$\sum_{s=1}^{R} p_s \frac{E_s a b^3}{12} \geq \vartheta, \tag{6.128}$$

$$\sum_{s=1}^{R} \sum_{i=0}^{N} p_s V_{s,i} \leq \omega, \tag{6.129}$$

where the parameters ϑ and ω are varied in the range of relevant rigidity and deflection values ($\vartheta \in (1.28 \cdot 10^{11}; 1.79 \cdot 10^{12})\,\mathrm{Nmm}^2$, $\omega \in (0.67; 9.37)\,\mathrm{mm}$). Both intervals are divided into 11 uniformly distributed values, and therefore, we solve $11 \cdot 11 = 121$ optimization programs with objective function (6.116), constraints (6.119)-(6.123), (6.128), (6.129) and different values of parameters ϑ and ω. These programs are solved for $R = 10$ scenarios instead of $R = 100$ scenarios because of the computational intensity.



The efficient frontier for our three-objective problem degenerates into the curve (see figure 6.22) because of the impact of involved physical characteristics on the feasible region. It means that different efficient points produce the same point in the objective function value space.

Figure 6.22: Efficient frontier.

Because we have employed the weighted sums approach in our problem (see objective function 6.124), we are interested in parametric analysis with respect to the weighting coefficients typically required by engineers. The weighting coefficient α is varied from 0 to 1 using increments of 0.05. The weighting coefficient β is varied from 0 to $1 - \alpha$ and γ is computed as $\gamma = 1 - \alpha - \beta$. For every value of α excluding the last one ($\alpha = 1$) we have 20 values of β and γ.

Figure 6.23 shows the effect of the weighting coefficients α and β on the optimal values of rigidity, weight and deflection. Figure 6.24 shows the same for optimal values of dimensions a and b. The effect of weighting coefficients is qualitatively the same for rigidity, weight and dimension a.

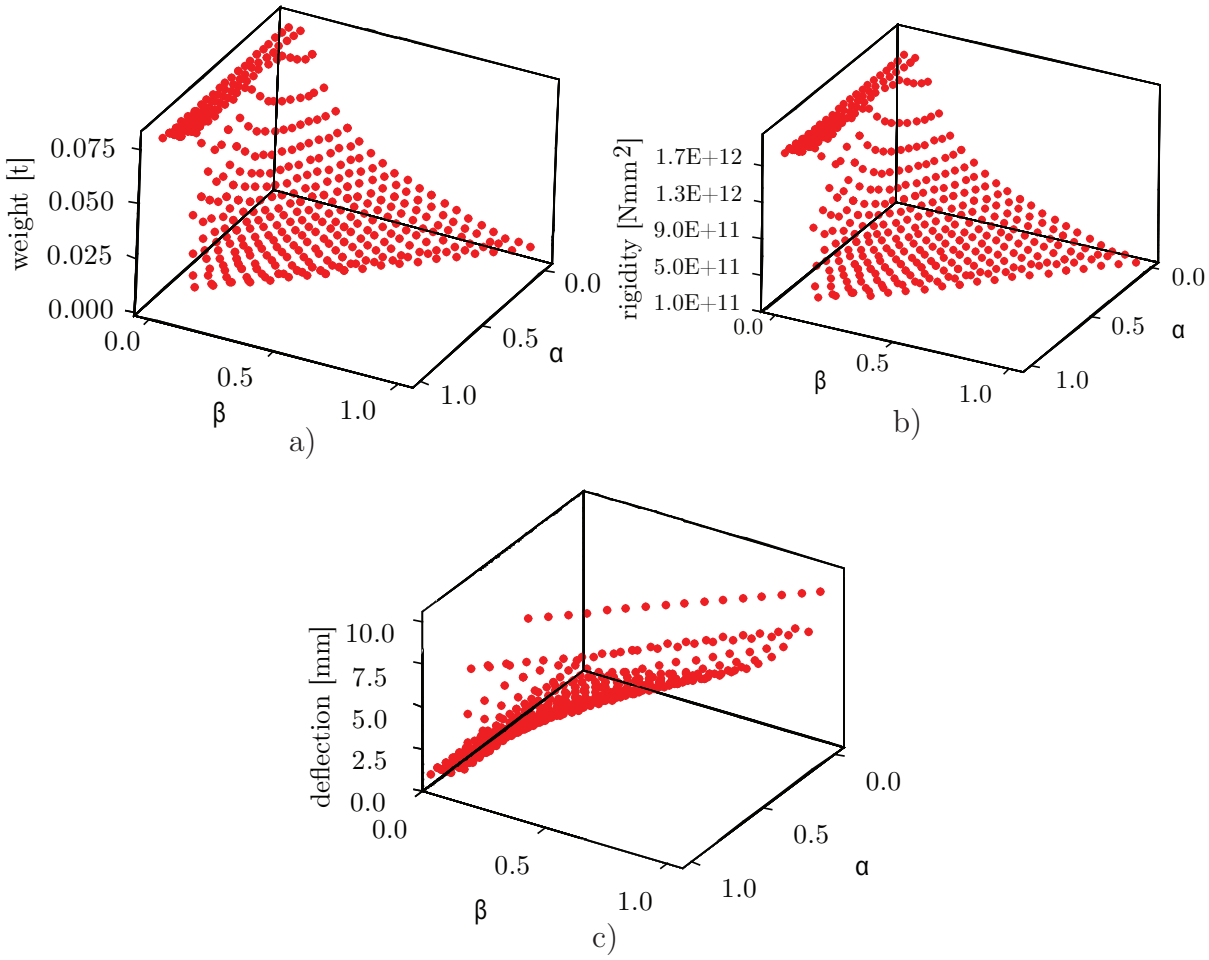


Figure 6.23: Beam weight, rigidity and deflection versus weighting coefficients α and β.

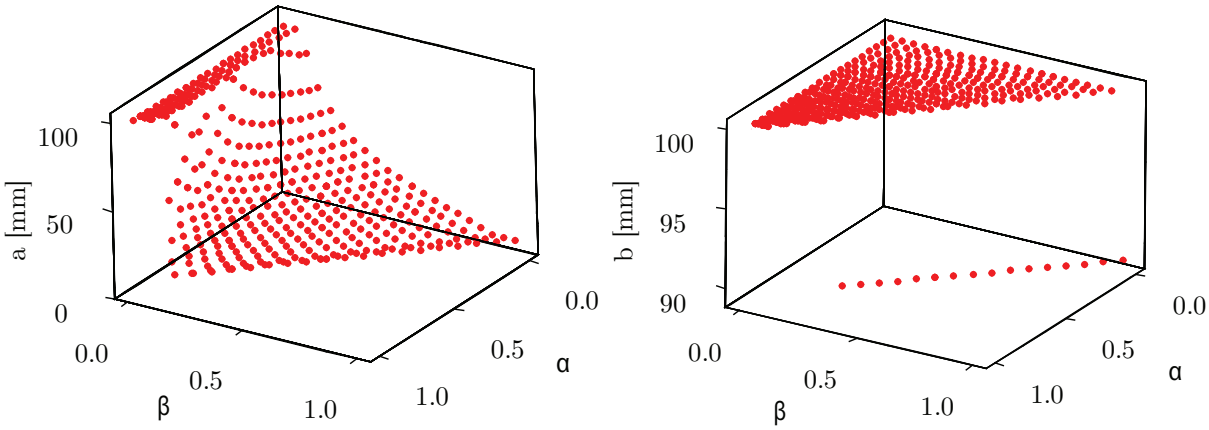


Figure 6.24: Beam cross section dimensions a and b versus weighting coefficients α, β.

We obtained two extreme solutions and many intermediate solutions by varying the weighting coefficients:

- $a = 10\,\text{mm}$, $b = 89.4\,\text{mm}$ for $\beta = 1 - \alpha$, $\alpha \in \langle 0; 0.75 \rangle$ (see figure 6.25 a)). This solution corresponds to minimum weight $0.007\,\text{t}$, minimum rigidity $1.29 \cdot 10^{11}\,\text{Nmm}^2$ and maximum deflection $9.3\,\text{mm}$.

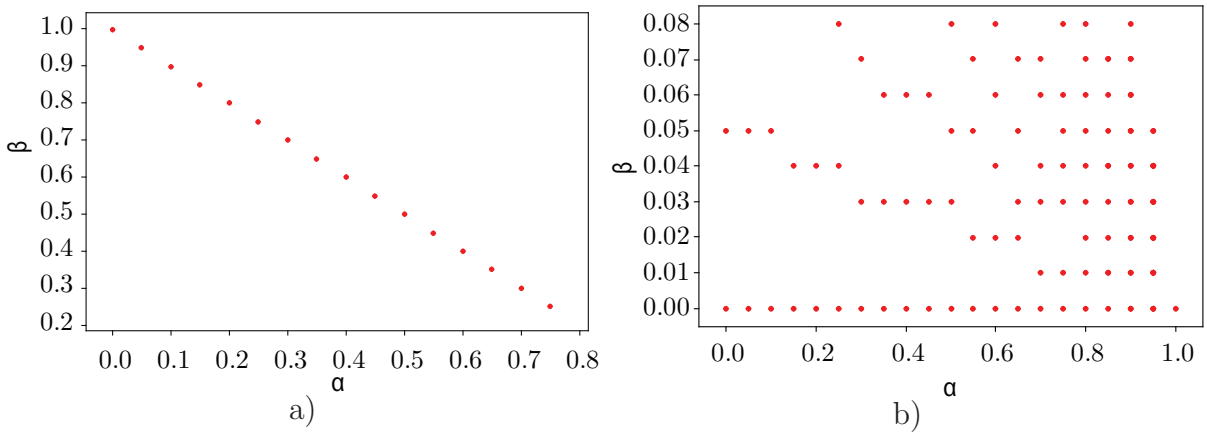


Figure 6.25: Relationship between α and β for dimensions a) $a = 10\,\text{mm}$ and $b = 89.4\,\text{mm}$, b) $a = 100\,\text{mm}$ and $b = 100\,\text{mm}$.

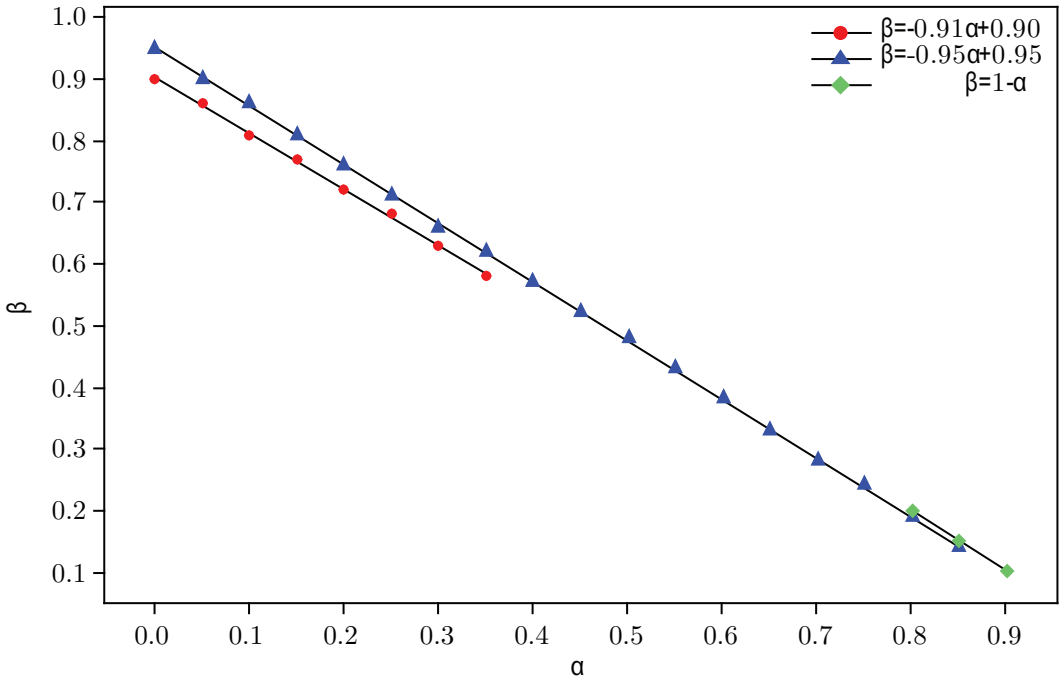


Figure 6.26: Relationship between α and β for dimensions $a = 10\,\text{mm}$ and $b = 100\,\text{mm}$.

- $a = 10\,\text{mm}$, $b = 100\,\text{mm}$ for $\beta = 1 - \alpha$, $\alpha \in \langle 0.8; 0.9\rangle$ or $\beta = -0.95\alpha + 0.95$, $\alpha \in \langle 0; 0.85\rangle$ or $\beta = -0.91\alpha + 0.9$, $\alpha \in \langle 0; 0.35\rangle$ (see figure 6.26).
  This is the intermediate solution. It corresponds to the weight $0.008\,\text{t}$, rigidity $1.80 \cdot 10^{11}\,\text{Nmm}^2$ and deflection $6.7\,\text{mm}$. The second and third equations for β are obtained by linear regression.

- $a \in (10; 100)\,\text{mm}$, $b = 100\,\text{mm}$ otherwise.
  These are also the intermediate solutions (see figure 6.24).

- $a = 100\,\text{mm}$, $b = 100\,\text{mm}$ for $\beta = 0$, $\alpha \in \langle 0; 1\rangle$ or $\beta \in (0; 0.08\rangle$, α varying (see figure 6.25 b)).
  This solution corresponds to maximum weight $0.079\,\text{t}$, maximum rigidity $1.80 \cdot 10^{12}\,\text{Nmm}^2$ and minimum deflection $0.7\,\text{mm}$.

It can be seen from the above summary of parametric analysis that the longer side of rectangular cross section b can reach only two values 89.4 mm and 100 mm while the shorter side a can vary in the whole interval $\langle 10; 100\rangle$ mm.

6.3.2 Stochastic optimization of design of beam cross section dimensions – variant II

In this section we will simplify our previous three-objective problem, omit minimization of deflection and study differences with respect to the results of previous model. It means that we will consider only two-objective optimization problem with minimizing the beam weight and maximizing the beam rigidity while the dimensions of the beam cross section have to be optimized. Hence, the underlying model will be as follows.

$$\min\ \rho abl \tag{6.130}$$

$$\max\ \frac{E(\xi)ab^3}{12} \tag{6.131}$$

$$\text{s.t. } E(\xi)\frac{ab^3}{12}\frac{\mathrm{d}^4 v}{\mathrm{d}x^4}(\xi,x) = h(x),\ x \in \langle 0,l\rangle,\ \xi \in \Xi, \tag{6.132}$$

$$v(\xi,0) = 0,\ \frac{\mathrm{d}v}{\mathrm{d}x}(\xi,0) = 0,\ \xi \in \Xi, \tag{6.133}$$

$$v(\xi,l) = 0,\ \frac{\mathrm{d}v}{\mathrm{d}x}(\xi,l) = 0,\ \xi \in \Xi, \tag{6.134}$$

$$\left|E(\xi)\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}(\xi,x)\frac{b}{2}\right| \le \sigma_{limit},\ x \in \langle 0,l\rangle,\ \xi \in \Xi, \tag{6.135}$$

$$a_{min} \le a \le a_{max}, \tag{6.136}$$

$$b_{min} \le b \le b_{max}. \tag{6.137}$$

EO deterministic reformulation converts this underlying program (6.130)-(6.137) into continuous stochastic nonlinear program. After using the approximations described in section 6.3.1, the following large multi-objective deterministic nonlinear program is obtained.

$$\min\ \rho abl \tag{6.138}$$

$$\max \sum_{s=1}^{R} p_s \frac{E_s ab^3}{12} \tag{6.139}$$

$$\text{s.t. } ab^3 \mathbb{K} E_s \mathbf{V}_s = \mathbf{f},\ s = 1,\dots,R, \tag{6.140}$$

$$V_{s,0} = 0,\ V_{s,N} = 0,\ s = 1,\dots,R, \tag{6.141}$$

$$|b\mathbb{C}E_s\mathbf{V}_s| \le d^2\sigma_{limit}\mathbf{g},\ s = 1,\dots,R, \tag{6.142}$$

$$a_{min} \le a \le a_{max}, \tag{6.143}$$

$$b_{min} \le b \le b_{max}, \tag{6.144}$$

where the matrices $\mathbb{K}$ and $\mathbb{C}$ and vector $\mathbf{V}_s$ have the same meaning as in section 6.3.1.

The weighted sum approach is used for practical implementation and the objectives functions (6.138)-(6.139) are replaced by the following single-objective function:

$$\min_{a,b,\mathbf{V}_s}\left(-\alpha\sum_{s=1}^{R} p_s\frac{E_s ab^3}{12c_{rigid}} + \beta\frac{\rho abl}{c_{weight}}\right), \tag{6.145}$$

where α, $\beta > 0$ are the weighting coefficients, $\alpha + \beta = 1$ and $c_{rigidity}, c_{weight}$ are typical values of rigidity and weight of the beam (i. e. normalizing constants). These values are obtained as the optimal values of objective function of two single-objective optimization problems.

The results are presented for the same input data and related formulas as in the previous section. The different values of the weighting coefficients $\alpha = 0.5$ and $\beta = 0.5$ are the only exception. The optimal objective function value is $z = 0.47$. The optimal dimensions are $a = 10\,\text{mm}$, $b = 89.4\,\text{mm}$. Deflection and maximum stress are presented in the figure 6.27 for model with only 20 scenarios because of the clarity. Maximum deflection of about 0.40 mm occurs in the middle of the beam while it decreases towards the beam's ends. Maximum stress reaches the same values for both reformulations, i. e. its values do not depend on the realization of random variable ξ. Maximum tensile stress 100 MPa occurs at the ends of the beam while maximum compression stress of about $-57\,\text{MPa}$ occurs in the middle of the beam. Optimal solution of this two-objectives problem fully takes advantage of the limiting value of maximum stress 100 MPa, and therefore, the optimal cross section is smaller. Hence, the beam weight is also smaller.

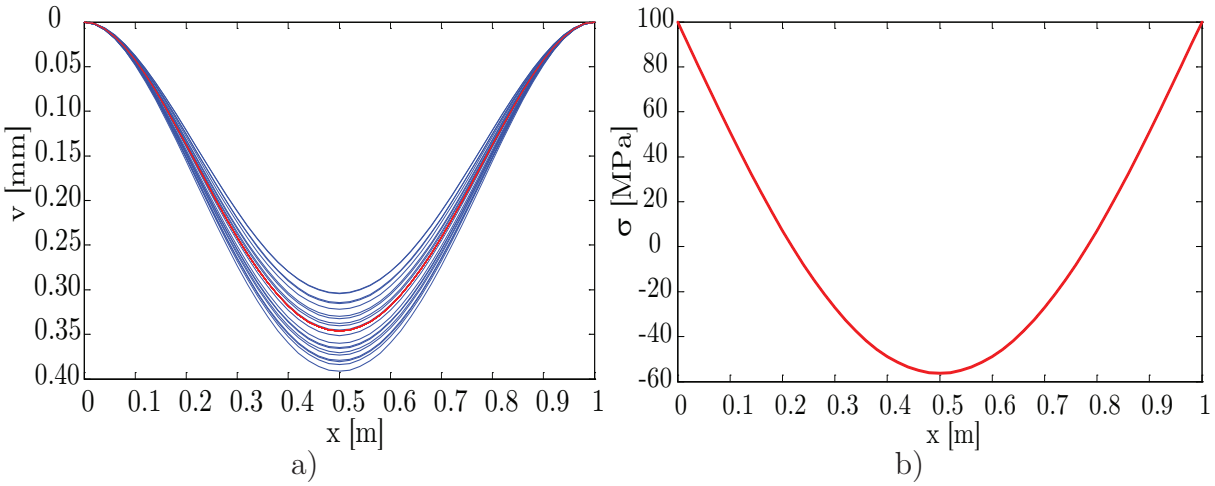


Figure 6.27: a) Deflection $v(\xi, x)$. Red line – EV model, blue lines – EO model, b) Maximum stress (red and blue lines coincide).

Furthermore, we will study model (6.138)-(6.144) from multi-objective point of view the same way as in the previous section. The efficient frontier is again computed by repeated optimization of (6.138)-(6.144). One more constraint enforcing achievement level for one criterion ((6.139) in our case) is added and the second criterion (6.138) is treated as a single-objective function. It means that we add the following constraint:

$$\sum_{s=1}^{R} p_s \frac{E_s a b^3}{12} \geq \vartheta, \tag{6.146}$$

where the parameter ϑ is varying in the range of relevant rigidity values ($\vartheta \in (1.29 \cdot 10^{11}; 1.80 \cdot 10^{12})\,\text{Nmm}^2$). This interval for ϑ is divided into 51 uniformly distributed values and therefore we solve 51 optimization programs with objective function (6.138), constraints (6.140)-(6.144), (6.146) and different values of parameter ϑ.

Efficient frontier of the two-objective problem is plotted in the figure 6.28. A feasible point $(a, b) = (10.7; 90)\,\text{mm}$ graphs as a point $(rigidity, weight) = (1.41 \cdot 10^{11}\,\text{Nmm}^2; 0.0076\,\text{t})$ but it cannot be efficient because it is dominated by a point $(a, b) = (10; 96.5)\,\text{mm}$

producing the point $(rigidity, weight) = (1.61 \cdot 10^{11}\,\mathrm{Nmm}^2; 0.0076\,\mathrm{t})$ with the same weight but superior rigidity. Feasibility of the non-efficient point was verified by fixing the values of the first-stage solution $(a, b) = (10.7; 90)\,\mathrm{mm}$ and solving program with objective function (6.145), constraints (6.140)-(6.144) and arbitrary combination of the weighting coefficients α and β. A point $(a, b) = (19.3; 50)\,\mathrm{mm}$ graphs as the point $(rigidity, weight) = (4.34 \cdot 10^{10}\,\mathrm{Nmm}^2; 0.0076\,\mathrm{t})$ which lies outside the feasible region bounded by efficient frontier and lines $weight = 0.078\,\mathrm{t}$ and $rigidity = 1.29 \cdot 10^{11}\,\mathrm{Nmm}^2$ and therefore is infeasible. The efficient frontier is the useful tool for engineers in practice. If the engineer selects a certain value of e. g. rigidity, then the efficient frontier enables him to find the best (smallest) possible weight. Furthermore, he can see all feasible combinations of rigidity and weight.

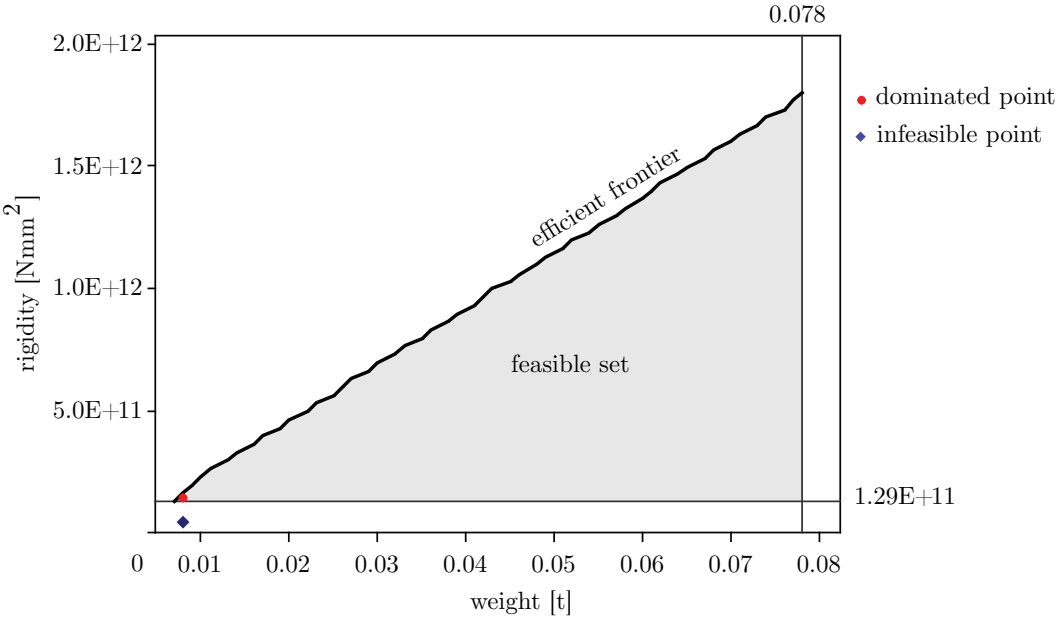


Figure 6.28: Efficient frontier.

The parametric analysis with respect to the weighting coefficients is made. The coefficient α is varied from 0 to 1 using increments of 0.02. The weighting coefficient β is computed as $\beta = 1 - \alpha$. Figure 6.29 shows the effect of the weighting coefficient α on the optimal values of rigidity and weight. Figure 6.30 shows the same for optimal values of cross section dimensions a and b. The effect of weighting coefficients is qualitatively the same for rigidity and weight, i. e. with increasing weighting coefficient α both weight and rigidity are increasing.

We obtained two extreme solutions and one intermediate solution by varying the weighting coefficient α:

- $a = 10\,\mathrm{mm}$, $b = 89.4\,\mathrm{mm}$ for $\alpha \in \langle 0; 0.78 \rangle$.
  This solution corresponds to minimum weight $0.007\,\mathrm{t}$ and minimum rigidity $1.29 \cdot 10^{11}\,\mathrm{Nmm}^2$.

- $a = 10\,\mathrm{mm}$, $b = 100\,\mathrm{mm}$ for $\alpha \in \langle 0.8; 0.9 \rangle$.
  This is the intermediate solution. It corresponds to weight $0.008\,\mathrm{t}$ and rigidity $1.80 \cdot 10^{11}\,\mathrm{Nmm}^2$.

- $a = 100\,\mathrm{mm}$, $b = 100\,\mathrm{mm}$ for $\alpha \in \langle 0.92; 1 \rangle$.
  This solution corresponds to maximum weight $0.079\,\mathrm{t}$ and maximum rigidity $1.80 \cdot 10^{12}\,\mathrm{Nmm}^2$.

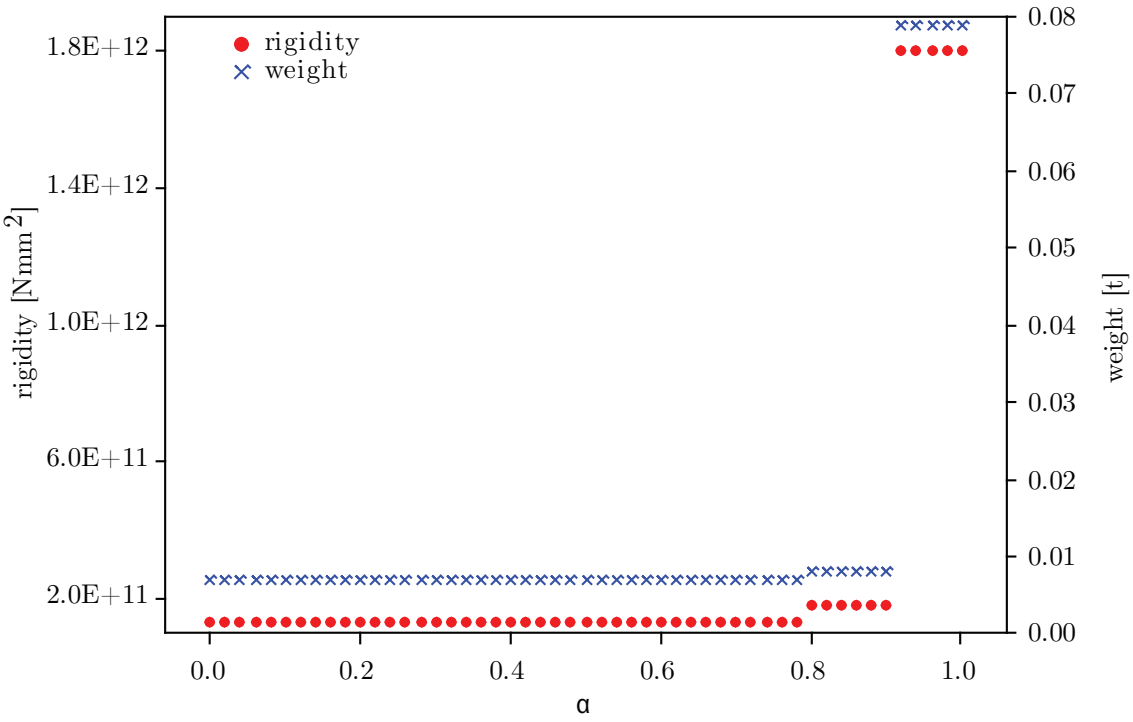


Figure 6.29: Beam rigidity and weight versus weighting coefficient α.

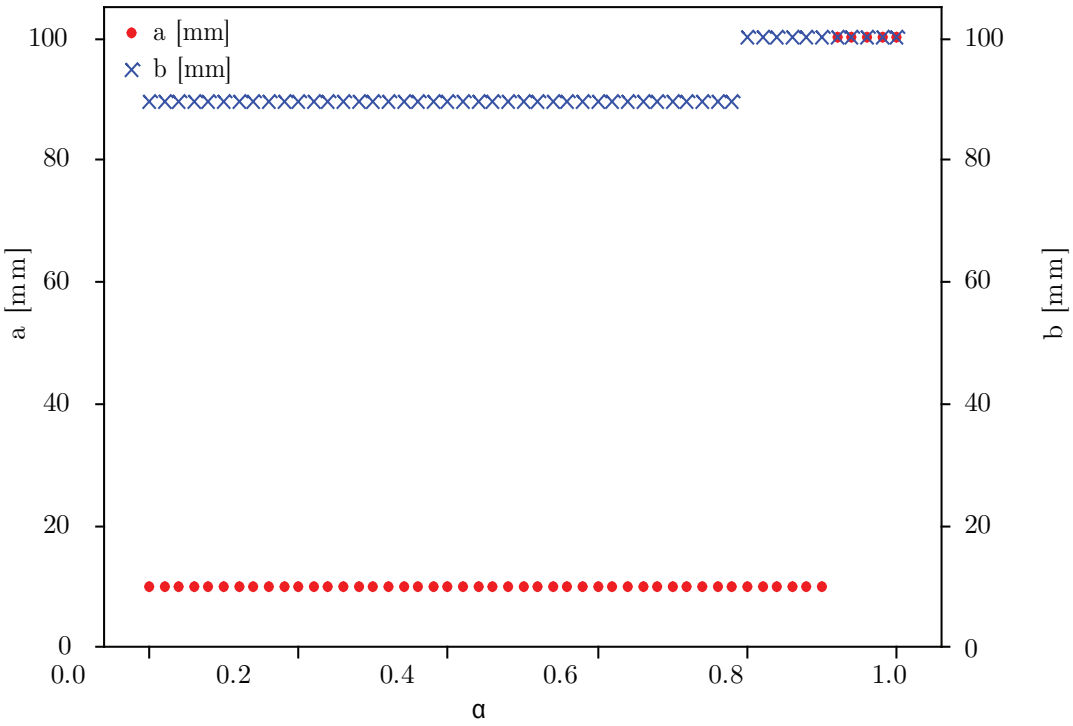


Figure 6.30: Beam dimensions a and b versus weighting coefficient α.

When we compare the results of parametric analysis for this two-objective and previous three-objective problems, we obtain the following conclusions. The longer side of rectangular cross section b can reach only two values 89.4 mm and 100 mm in both problems. But the shorter side a reaches only two values 10 mm and 100 mm given by constraint (6.143) in case of two-objective problem while it can vary in the whole interval $(10; 100)$ mm for three-objective problem. It means that all three solutions of two-objective problem are involved in the set of all solutions of three-objective problem.

6.3.3 Chance constrained stochastic optimization of design of beam cross section dimensions

At the end of this chapter, we will study the alternatives of approximations of the model that includes reliability-related probabilistic terms (the aim of the Ph. D. thesis no. 4).

There can be found various requirements for optimization in applications, e. g. to increase reliability of some equipment (especially in engineering). Therefore, our last reformulation will contain individual chance constraint (see inequality (6.152)) which provides a simple probability measure. Strictly speaking, the probability that the maximum deflection does not exceed a given limiting value has to be greater or equal to a given confidence level. We can imagine that if the deflection of the beam is too large, then the beam can be damaged and its reliability may become worse. Therefore, we compute empirical probability of satisfying the constraint $\max_x v(\xi, x) \leq A$ (notation is explained below) for the previous two-objective model and find out that it is too small. For that reason we add the chance constraint into model to obtain higher reliability. The corresponding underlying program has the following form.

$$\min\ \rho abl \tag{6.147}$$

$$\max\ \frac{E(\xi)ab^3}{12} \tag{6.148}$$

$$\text{s. t. } E(\xi)\frac{ab^3}{12}\frac{\mathrm{d}^4 v}{\mathrm{d}x^4}(\xi, x) = h(x),\ x \in \langle 0, l\rangle,\ \xi \in \Xi, \tag{6.149}$$

$$v(\xi, 0) = 0,\ \frac{\mathrm{d}v}{\mathrm{d}x}(\xi, 0) = 0,\ \xi \in \Xi, \tag{6.150}$$

$$v(\xi, l) = 0,\ \frac{\mathrm{d}v}{\mathrm{d}x}(\xi, l) = 0,\ \xi \in \Xi, \tag{6.151}$$

$$\mathbb{P}(\max_x v(\xi, x) \leq A) \geq 1 - \alpha_p,\ \xi \in \Xi, \tag{6.152}$$

$$\left| E(\xi)\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}(\xi, x)\frac{b}{2} \right| \leq \sigma_{limit},\ x \in \langle 0, l\rangle,\ \xi \in \Xi, \tag{6.153}$$

$$a_{min} \leq a \leq a_{max}, \tag{6.154}$$

$$b_{min} \leq b \leq b_{max}, \tag{6.155}$$

where $\mathbb{P}$ is a probability measure on $(\Xi, \mathcal{B})$, A is a given limiting value for maximum deflection and $1 - \alpha_p$ is a confidence level.

Scenario-based approach and finite difference method described in section 6.3.1 and the weighted sum approach are used to obtain approximating programs.

The first possible approach for solving an optimization problem involving random variable with discrete distribution with R scenarios is a mixed-integer nonlinear reformulation (MINLP) (see [63] for the formulation and [52] for precise theoretical results). But there is a problem with function max in constraint (6.152) in practical implementation.

Therefore, maximum is replaced by summation, i. e.

$$\mathbb{P}(\sum_{x} v(\xi, x) \leq A) \geq 1 - \alpha_p,\ \xi \in \Xi. \tag{6.156}$$

The chance constraint (6.156) is then approximated by two inequalities (6.160) and (6.161). The approximating deterministic mixed-integer program has the following form:

$$\min_{a,b,\mathbf{V}_s,\delta_s} \left(-\alpha \sum_{s=1}^{R} p_s \frac{E_s a b^3}{12 c_{rigid}} + \beta \frac{\rho a b l}{c_{weight}} \right) \tag{6.157}$$

$$\text{s. t.} \quad ab^3 \mathbb{K} E_s \mathbf{V}_s = \mathbf{f},\ s = 1, \ldots, R, \tag{6.158}$$

$$V_{s,0} = 0,\ V_{s,N} = 0,\ s = 1, \ldots, R, \tag{6.159}$$

$$\sum_{i=0}^{N} V_{s,i} - M_p(1 - \delta_s) \leq A,\ s = 1, \ldots, R, \tag{6.160}$$

$$\sum_{s=1}^{R} p_s \delta_s \geq 1 - \alpha_p, \tag{6.161}$$

$$|b \mathbb{C} E_s \mathbf{V}_s| \leq d^2 \sigma_{limit} \mathbf{g},\ s = 1, \ldots, R, \tag{6.162}$$

$$a_{min} \leq a \leq a_{max}, \tag{6.163}$$

$$b_{min} \leq b \leq b_{max}, \tag{6.164}$$

where $\delta_s \in \{0, 1\}$, $s = 1, \ldots, R$ are the binary variables and M_p is a sufficiently large number which bounds $\sum_{i=0}^{N} V_{s,i} - A$ for $s = 1, \ldots, R$ from above and the other notation is the same as in the section 6.3.1.

This model has been implemented in GAMS with solver of mixed-integer programming problems BARON (see section 7.1). This solver has been selected as a suitable solution tool because of the recent computational experience, see [34]. The limiting value has been computed as $A = \varepsilon - 1$, where $\varepsilon = \max_{s} \sum_{i=0}^{N} V_{s,i}$ is computed for program (6.145), (6.140)-(6.144) with corresponding number of scenarios. Hence, $M_p \geq \varepsilon - (\varepsilon - 1) = 1$ and we set its value to $M_p = 2$.

The confidence level has been chosen as $1 - \alpha_p = 0.8$. At first, only $R = 5$ scenarios have been assumed with $A = 9.004\,\text{mm}$. CPU time was 14:46 min for PC with Intel Core i7 CPU 920 2.79 GHz and 12 GB RAM. The objective function value is $z = 0.48$, optimal dimensions are $a = 10\,\text{mm}$ and $b = 92.6\,\text{mm}$. Optimal values of binary variables are $\delta_s = 1$ for all s. After computing $\sum_{i=0}^{N} V_{s,i}$, we find out that the constraint $\sum_{x} v(\xi, x) \leq A$ is satisfied for all five scenarios and therefore the empirical probability of satisfying this constraint is equal to 1. The reason for this situation is the following. The constraint (6.160) is derived from the implication condition $\delta_s = 1 \Rightarrow \sum_{i=0}^{N} V_{s,i} \leq A$ and that is why the inequality $\sum_{i=0}^{N} V_{s,i} \leq A$ has to be satisfied for $\delta_s = 1$ and we cannot say anything about relationship of $\sum_{i=0}^{N} V_{s,i}$ and A for $\delta_s = 0$ (see table 6.4).

$\delta_s = 1$	$\sum_{i=0}^{N} V_{s,i} \leq A$	$\delta_s = 1 \Rightarrow \sum_{i=0}^{N} V_{s,i} \leq A$
1	1	1
1	0	0
0	1	1
0	0	1

Table 6.4: Truth table of implication.

Furthermore, we are also interested in the influence of the value M_p. It can be seen from the table 6.5 that M_p influences solution very much and it is better to choose smaller values to obtain more reliable results corresponding to the results from penalty reformulation. The interesting thing is a decrease of CPU time for the case with $M_p = 1000$, whereas CPU time is by then increasing with increasing M_p. It is probably due to a "step change" in solution from $b \approx 90\,\text{mm}$ to $b = 100\,\text{mm}$. This solution is equal to the initial iteration 100 mm and the solver therefore needs less iterations to obtain optimal solution.

Value M_p	**CPU time** [s]	**Objective value** z	**Optimal solution** a [mm]	b [mm]
2	886	0.48	10.0	92.6
100	1803	0.48	10.0	91.6
500	2296	0.51	11.0	89.0
1000	609	0.52	10.1	100

Table 6.5: Influence of M_p on solution of MINLP reformulation for $R = 5$ scenarios.

CPU time of solving model with $R = 15$ scenarios with $A = 9.328\,\text{mm}$ increased rapidly, it was 27:23:06 h. The objective function value is $z = 0.47$, optimal dimensions are $a = 10\,\text{mm}$ and $b = 91.6\,\text{mm}$. Optimal values of binary variables are $\delta_s = 0$ for $s = 9, 15$ and $\delta_s = 1$ otherwise, and the constraint $\sum_x v(\xi, x) \leq A$ is not satisfied for these two scenarios. Therefore, the empirical probability of satisfying this constraint is equal to $\frac{13}{15}$. The table 6.6 shows the summary of MINLP reformulation results for several number of scenarios. As above, a decrease of CPU time for the case with $R = 20$ scenarios occurs, whereas CPU time is by then increasing with increasing R. The statement of the reasons is the same as before ("step change" in solution). CPU times in the table advert to really large computational intensity and to inapplicability of this approach for models with more scenarios.

No. of scenarios R	**Limiting value** A [mm]	**CPU time** [s]	**Objective value** z	**Optimal solution** a [mm]	b [mm]
5	9.004	886	0.48	10	92.6
10	9.328	12 595	0.48	10	92.8
15	9.328	98 586	0.47	10	91.6
20	9.328	54 713	0.51	10	100

Table 6.6: Summary of MINLP reformulation results.

Therefore, another approach is used for larger problems. It reformulates a chance constrained problem using an appropriate penalty function ν, i. e. the chance constraint (6.156) is incorporated into the objective function as follows.

$$\min_{a,b,v(\xi)} \left[-\alpha \mathbb{E}\left(\frac{E(\xi) a b^3}{12 c_{rigid}} \right) + \beta \frac{\rho a b l}{c_{weight}} + M_{pen} \mathbb{E}\left(\nu \left(\sum_x v(\xi, x) - A \right) \right) \right], \tag{6.165}$$

where $\nu : \mathbb{R} \to \mathbb{R}_0^+$ is an continuous nondecreasing function equal to 0 on $\mathbb{R}_0^-$ and positive otherwise [7] and M_{pen} is an penalty coefficient. Frequently used penalty function is the following one [3] (see figure 6.31):

$$\nu(s) = (\max(0, s))^2. \tag{6.166}$$

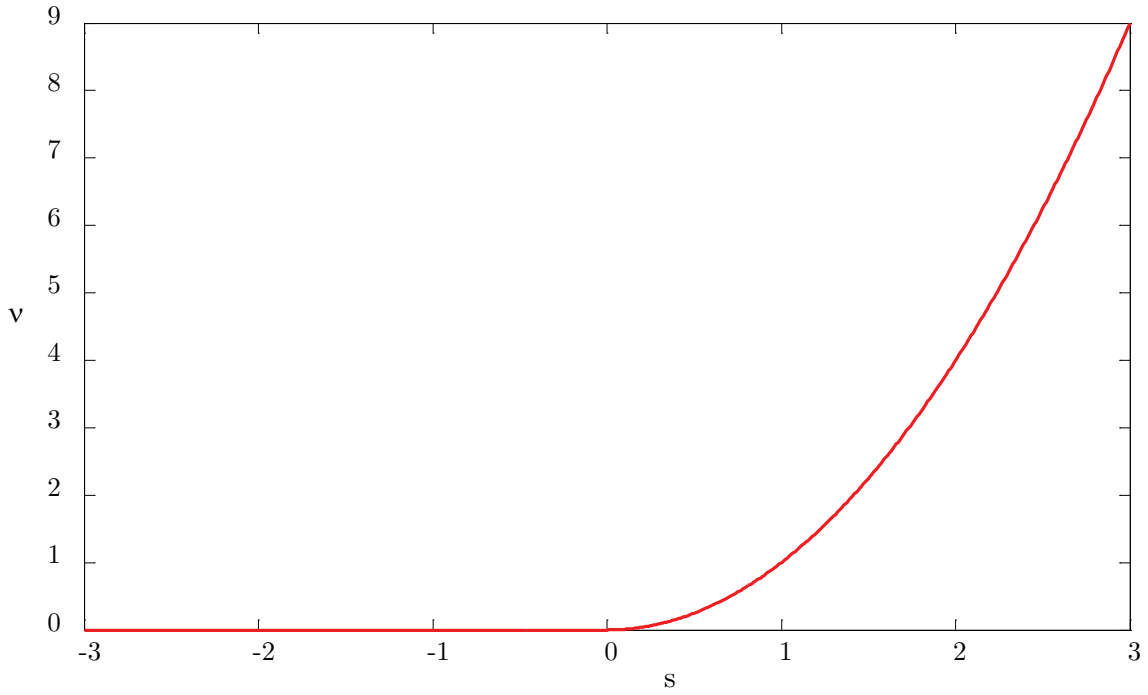


Figure 6.31: Penalty function ν.

Hence, the stochastic program with the objective function (6.165) and constraints (6.107)-(6.112) is approximated by the following deterministic nonlinear program where the penalized chance constraint plays the role of recourse function.

$$\min_{a,b,\mathbf{V}_s} \left[-\alpha \sum_{s=1}^{R} p_s \frac{E_s a b^3}{12 c_{rigid}} + \beta \frac{\rho a b l}{c_{weight}} + M_{pen} \sum_{s=1}^{R} p_s \left(\max\left(0, \sum_{i=0}^{N} V_{s,i} - A \right) \right)^2 \right] \tag{6.167}$$

$$\text{s. t. } a b^3 \mathbb{K} E_s \mathbf{V}_s = \mathbf{f},\ s = 1, \ldots, R, \tag{6.168}$$

$$V_{s,0} = 0,\ V_{s,N} = 0,\ s = 1, \ldots, R, \tag{6.169}$$

$$|b \mathbb{C} E_s \mathbf{V}_s| \leq d^2 \sigma_{limit} \mathbf{g},\ s = 1, \ldots, R, \tag{6.170}$$

$$a_{min} \leq a \leq a_{max}, \tag{6.171}$$

$$b_{min} \leq b \leq b_{max}. \tag{6.172}$$

The limiting value A is chosen the same way as in the section about MINLP reformulation. The results of penalty reformulation depending on different values of penalty coefficient M_{pen} are summarized in the table 6.7. CPU times are negligible (less than 10 s) in comparison to CPU times of MINLP reformulation.

Penalty coeff. M_{pen}	**Total obj. value**	**First-stage obj. value**	**Recourse obj. value**	**Optimal solution** a [mm]	b [mm]	**Empirical probability**
0.001	0.4659	0.4658	$1.3 \cdot 10^{-4}$	10	89.44	0.55
0.01	0.4670	0.4658	$1.3 \cdot 10^{-3}$	10	89.44	0.55
0.05	0.4714	0.4681	$3.3 \cdot 10^{-3}$	10	89.98	0.66
0.1	0.4734	0.4710	$2.3 \cdot 10^{-3}$	10	90.64	0.74
0.2	0.4748	0.4730	$1.8 \cdot 10^{-3}$	10	91.09	0.81
0.5	0.4762	0.4749	$1.3 \cdot 10^{-3}$	10	91.54	0.90
1	0.4769	0.4760	$9.3 \cdot 10^{-4}$	10	91.78	0.92
1.9	0.4775	0.4767	$7.5 \cdot 10^{-4}$	10	91.95	0.95
5	0.4781	0.4776	$5.5 \cdot 10^{-4}$	10	92.14	0.97
10	0.4785	0.4780	$4.8 \cdot 10^{-4}$	10	92.25	0.99
100	0.4789	0.4789	$4.9 \cdot 10^{-5}$	10	92.45	0.99
1000	0.4790	0.4790	$4.9 \cdot 10^{-6}$	10	92.47	0.99
10 000	0.4790	0.4790	$4.9 \cdot 10^{-7}$	10	92.47	0.99
100 000	0.4790	0.4790	$4.9 \cdot 10^{-8}$	10	92.47	0.99

Table 6.7: Summary of penalty reformulation results for $R = 100$ scenarios.

The first-stage objective value is $-\alpha \sum_{s=1}^{R} p_s \frac{E_s a b^3}{12 c_{rigid}} + \beta \frac{\rho a b l}{c_{weight}}$ evaluated in the optimal solutions a and b while the recourse objective value means $M_{pen} \sum_{s=1}^{R} p_s \left(\max \left(0, \sum_{i=0}^{N} V_{s,i} - A \right) \right)^2$ evaluated in the optimal solution $V_{s,i}$. Empirical probability is computed as the relative frequency of event $\sum_x v(\xi, x) \leq A$, i. e. number of scenarios for which the inequality $\sum_{i=0}^{N} V_{s,i} \leq A$ is fulfilled divided by total number of scenarios ($R = 100$ scenarios in our case). It is evident that the dimension b is increasing with the increasing value of M_{pen}. Also the empirical probability of satisfying $\sum_x v(\xi, x) \leq A$ increases which is easily understandable. An increase of b results in rigidity rise and deflection decrease which are obviously related to increasing of the aforementioned empirical probability. Also total objective value together with the first-stage objective value are increasing with the increasing value of M_{pen} while the recourse objective value is decreasing. It is obvious that the smallest probability of satisfying the deflection constraint occurs for the solution of the program (6.124), (6.119)-(6.123) without the corresponding chance constraint. Relationship between the value of penalty coefficient M_{pen} and the probability of satisfying the constraint $\sum_x v(\xi, x) \leq A$ is presented in the figure 6.32. We can easily determine from this graph the value of penalty coefficient needed to achieve the given probability level and vice versa. This procedure is demonstrated for the probability level 0.95 for which the value $M_{pen} = 1.9$ is obtained. It can be seen in the table 6.7 that for $M_{pen} = 1.9$ the empirical probability is indeed 0.95.

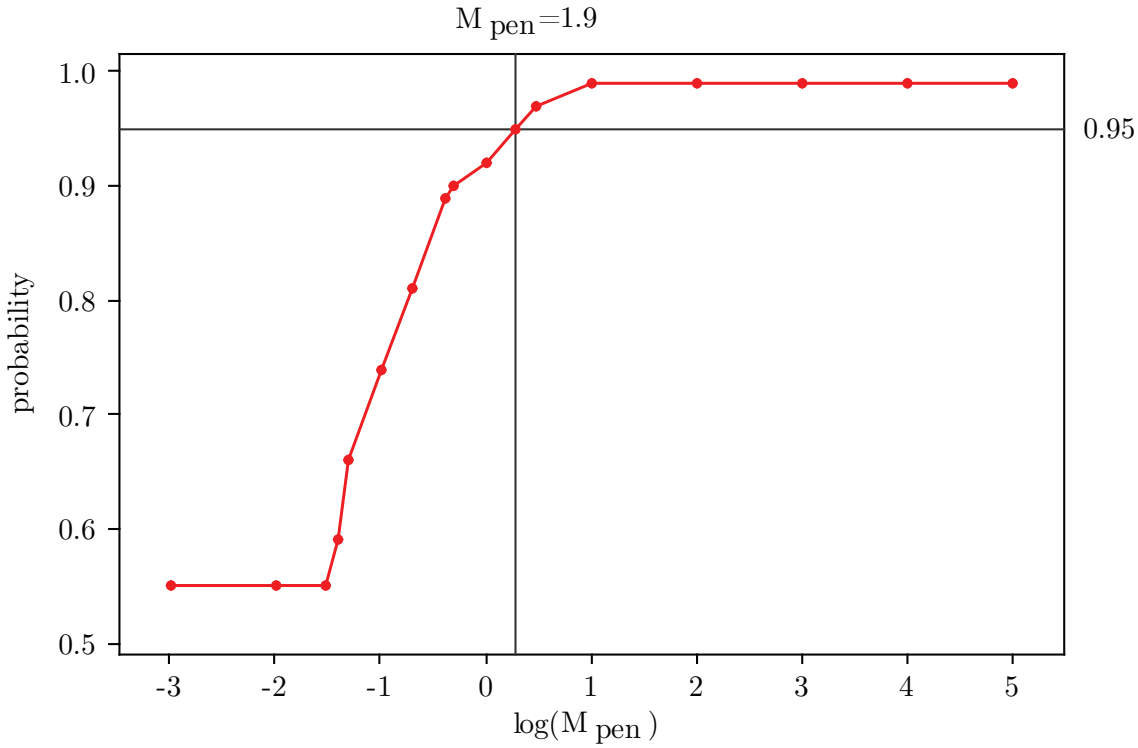


Figure 6.32: Probability of satisfying the constraint $\sum_x v(\xi, x) \leq A$ versus penalty coefficient M_{pen}.

Deflection and maximum stress are compared in the figures 6.33 and 6.34 for models with and without chance constraint (6.156) with 20 scenarios because of the clarity. Maximum deflections 0.39 mm, respectively 0.35 mm for model without, respectively with chance constraint occur in the middle of the beam while it decreases towards the beam's ends. Maximum tensile stresses 100 MPa, respectively 93.4 MPa occur at the ends of the beam while maximum compression stresses -56.3 MPa, respectively -52.6 MPa occurs in the middle of the beam for model without, respectively with chance constraint.

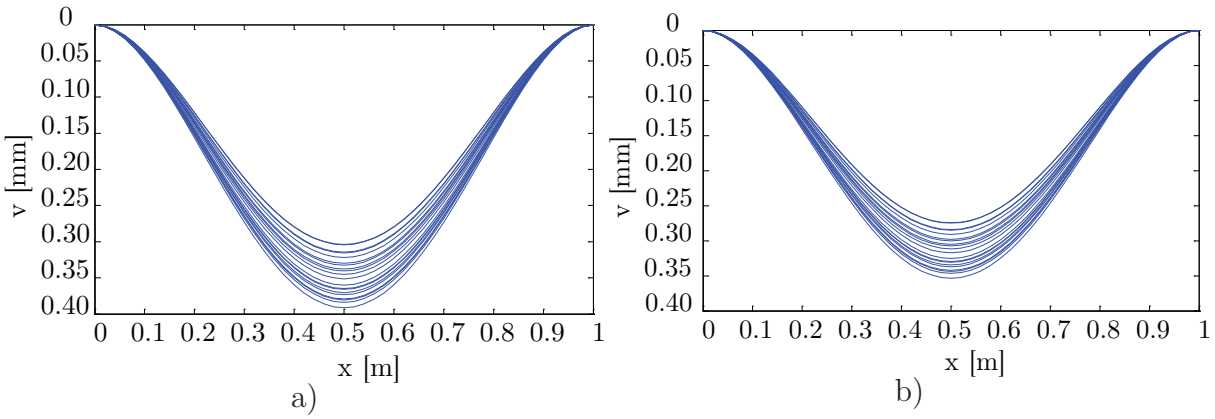


Figure 6.33: Deflection $v(\xi, x)$ for model a) without chance constraint, b) with chance constraint.

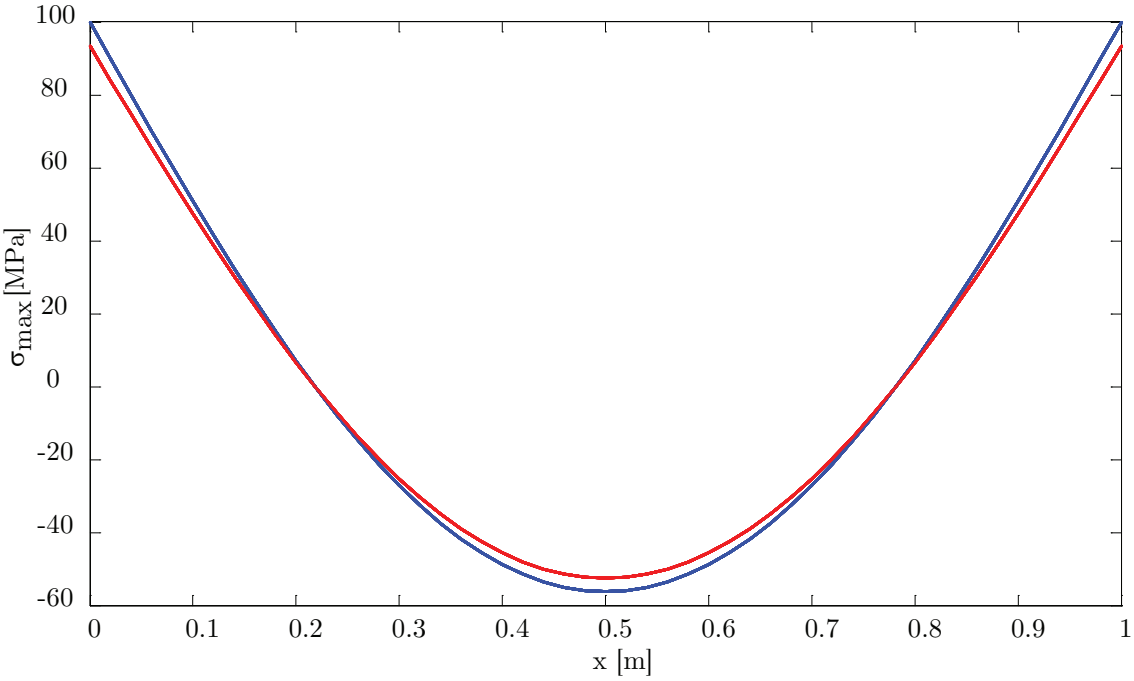


Figure 6.34: Maximum stress $\sigma_{max}(x)$. Blue line – model without chance constraint, red line – model with chance constraint.

Finally, these two described reformulations of the chance constrained problem are compared in the table 6.8 for different numbers of scenarios, $M_p = 2$ and $M_{pen} = 1000$. Their solutions are quite comparable for $R = 5, 10, 15$, for $R = 20$ the dimension b is about 8% bigger in case of MINLP reformulation. But the largest difference is in CPU time which is much higher for MINLP reformulation.

No. of scenarios	**Obj. value**	**Optimal solution**		**Obj. value**	**Optimal solution**	
R	z_{MINLP}	a [mm]	b [mm]	z_{pen}	a [mm]	b [mm]
5	0.48	10.02	92.57	0.48	10.00	92.64
10	0.48	10.00	92.83	0.48	10.00	92.53
15	0.47	10.00	91.56	0.48	10.00	92.53
20	0.51	10.00	99.97	0.48	10.00	92.53

Table 6.8: Comparison of MINLP and penalty reformulations.

Chapter 7

Algorithms and their implementations

This chapter provide the description of a method used to determining the solution quality based on Monte Carlo bounding technique, then the progressive hedging algorithm is introduced together with the description of the practical implementation. And finally, finite difference method and finite element method are compared for deterministic version of ODE constrained problem by using GAMS and ANSYS.

7.1 GAMS

All models have been implemented in optimization software GAMS and several sophisticated solvers have been used depending on the type of model. GAMS (General Algebraic Modelling System) is an example of the algebraic modelling language which provides a high level programming language for the compact representation of large and complex models. It allows to define the problem formally using mathematical notation and also permits model description that is independent of solution algorithms. An optimization problem is expressible independently of the used data which allows a problem to be increased in size without increasing the complexity of the representation [8].

CONOPT is a solver especially for nonlinear models (NLP and also DNLP models) and the algorithm is based on the Generalized Reduced Gradient (GRG) method [13]. BARON (Branch-And-Reduce Optimization Navigator) [53] is a solver for the global solution of nonlinear (NLP) and mixed-integer nonlinear (MINLP) programs. It implements algorithms of the branch-and-bound type enhanced with a variety of constraint propagation and duality techniques for reducing ranges of variables.

We have encountered several difficulties during the practical implementation of our models into GAMS. For example, the constraint (6.121) related to maximum stress limitation in stochastic optimization model of design of beam cross section dimensions involves absolute value. Absolute value is a non-smooth function and may cause numerical problems, especially when the arguments of the function are variables (so called endogenous arguments). Therefore, it should be used only if unavoidable and a special model type called "dnlp" (it means nonlinear programming with discontinuous derivatives) have to be used in addition [8]. That is why, we have replaced above mentioned constraint $|b\mathbb{C}E_s\mathbf{V}_s| \leq d^2\sigma_{limit}\mathbf{g}$, $s = 1, \ldots, R$ by the following two constraints:

$$b\mathbb{C}E_s\mathbf{V}_s \leq d^2\sigma_{limit}\mathbf{g},\ s = 1, \ldots, R \tag{7.1}$$

$$-b\mathbb{C}E_s\mathbf{V}_s \leq d^2\sigma_{limit}\mathbf{g},\ s = 1, \ldots, R. \tag{7.2}$$

Another problems occurred during solving chance constrained program, strictly speaking during solving of its mixed-integer reformulation using solver BARON. This solver cannot handle the function max which is included in the constraint (6.152). Therefore, maximum is replaced by summation as has been explained at the end of the previous chapter. Furthermore, because of its computational intensity the default value of termination option MaxTime meaning maximum CPU time allowed had to be noticeably increased (from 1200 s to 100 000 s).

Every type of graphical result has been obtained using the environment of MATLAB R2007a (The MathWorks, Inc., USA) or statistical software MINITAB 15 (Minitab Inc., USA). MINITAB has been also used for statistical computations that were necessary e. g. in Monte Carlo technique (calculations of point estimates, interval estimates, etc.).

7.2 Monte Carlo bounding technique

Let us assume a stochastic programming problem which is so-called true optimization problem in the following form:

$$z^* = \min_{\mathbf{x}\in C}\{f(\mathbf{x}) = \mathbb{E}[F(\boldsymbol{\xi}, \mathbf{x})]\}, \tag{7.3}$$

where $\mathbf{x}$ is a decision vector (first-stage decision in case of two-stage programs), C is the feasible set, $F(\boldsymbol{\xi}, \mathbf{x})$ is a real-valued function with finite mean and variance and the optimal solution is denoted as $\mathbf{x}^*$. In our studied cases of two-stage programs, $F(\boldsymbol{\xi}, \mathbf{x})$ is the optimal value of the second-stage program plus possible first-stage objective function if included. We suppose that the support of the probability distribution of $\boldsymbol{\xi}$ is $\Xi = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_R\}$, i. e. we have finite number of scenarios with respective probabilities p_s. The problem is that R grows exponentially with dimension K of $\boldsymbol{\xi}$, i. e. $\boldsymbol{\xi} \in \mathbb{R}^K$ with independent components each with m realizations means $R = m^K$ scenarios. Usually, a number of scenarios is really big and it is impossible to solve stochastic program (7.3) exactly. Therefore, Monte Carlo sampling techniques are used to reduce the number of scenarios to a manageable level.

There are two methods of sampling. In interior methods, the sampling is performed inside a chosen algorithm with new samples generated at each iteration. In exterior sampling methods, which we will use, the sample is generated outside of the considered optimization problem, then the approximation problem is constructed and solved by an appropriate deterministic algorithm.

At first, we need to generate a sample of n equally probable replications $\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^n$ of the random vector $\boldsymbol{\xi}$. Monte Carlo sampling method is accomplished by generating a random sequence $U^1, \ldots, U^n$ of numbers independent of each other and uniformly distributed on $\langle 0, 1\rangle$. Then an i. i. d. sample of $\boldsymbol{\xi}$ is constructed by an appropriate transformation [50].

The true problem (7.3) is then approximated by the sample average approximation (SAA) problem [52]:

$$\hat{z}_n = \min_{\mathbf{x}\in C}\{\hat{f}_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} F(\mathbf{x}, \boldsymbol{\xi}^i)\}, \tag{7.4}$$

where the optimal solution is denoted as $\hat{\mathbf{x}}_n$. Once the sample is generated, $\hat{f}_n(\mathbf{x})$ becomes a deterministic function and SAA problem is solved by an appropriate deterministic algorithm. So $\hat{z}_n$, $\hat{\mathbf{x}}_n$ are the statistical estimators of their counterparts of the true problem (7.3).

It can be shown [52] that:

- $\hat{f}_n(\mathbf{x})$ is an unbiased estimator of $f(\mathbf{x})$, i. e. $\mathbb{E}[\hat{f}_n(\mathbf{x})] = f(\mathbf{x})$.
- $\hat{z}_n$ is a downwards biased estimator of z^* (probabilistic lower bound), i. e. $\mathbb{E}[\hat{z}_n] \leq z^*$.
- $\mathbb{E}[\hat{z}_n] \leq \mathbb{E}[\hat{z}_{n+1}] \leq z^*$, i. e. better lower bounds are obtained with increasing sample size.

Consistency of the estimators gives us an assurance that the error of the estimation approaches zero in the limit as the sample size grows to infinity. But it does not give us any information about the magnitude of the error. It can be proved [31] that $\hat{z}_n$ is a strongly consistent estimator of z^*, i. e. $\hat{z}_n$ converges to z^* w.p.1 as $n \to \infty$.

It is important to assess the quality of our solution. This solution $\bar{\mathbf{x}}$ is most probably suboptimal and we need to estimate how far it is from the optimal solution. We use the concept of Monte Carlo bounding technique for determining solution quality proposed by Morton et al. [31] where the optimality gap is estimated as a measure of solution quality.

$$G(\bar{\mathbf{x}}) = f(\bar{\mathbf{x}}) - z^* \geq 0, \tag{7.5}$$

where $\bar{\mathbf{x}} \in C$ is a candidate for an optimal solution of the true problem (7.3). The fact that $\hat{z}_n$ forms the probabilistic lower bound on z^* is used to construct confidence interval on the optimality gap for any candidate solution. The optimality gap given by solving approximation problems with n scenarios is:

$$G_n(\bar{\mathbf{x}}) = \hat{f}_n(\bar{\mathbf{x}}) - \hat{z}_n. \tag{7.6}$$

The gap $G_n(\bar{\mathbf{x}})$ is estimated by averaging, i. e. we have to generate n_g i. i. d. samples $\boldsymbol{\xi}^{1j}, \ldots, \boldsymbol{\xi}^{nj}$ each of size n and compute $\hat{f}_n^j(\bar{\mathbf{x}})$ and $\hat{z}_n^j$, $j = 1, \ldots, n_g$. We use the same set of observations of $\boldsymbol{\xi}$ for computing $\hat{f}_n^j(\bar{\mathbf{x}})$ and $\hat{z}_n^j$ which is the application of the common random numbers technique. Then the point estimate of $G(\bar{\mathbf{x}})$ is:

$$\hat{G}_{n,n_g}(\bar{\mathbf{x}}) = \frac{1}{n_g}\sum_{j=1}^{n_g} G_n^j(\bar{\mathbf{x}}) = \frac{1}{n_g}\sum_{j=1}^{n_g}[\hat{f}_n^j(\bar{\mathbf{x}}) - \hat{z}_n^j]. \tag{7.7}$$

By central limit theorem we get $\sqrt{n_g}(\hat{G}_{n,n_g}(\bar{\mathbf{x}}) - \mathbb{E}[G_n(\bar{\mathbf{x}})]) \Rightarrow N(0, \sigma_{n_g}^2(\bar{\mathbf{x}}))$ as $n_g \to \infty$, where $\Rightarrow$ denotes convergence in distribution, $N(0, \sigma^2)$ is a normal random variable with zero mean and variance σ^2 and $\sigma_{n_g}^2(\bar{\mathbf{x}}) = \mathrm{var}(G_n(\bar{\mathbf{x}}))$. By means of this result we obtain $(1-\alpha)$-level confidence interval for the optimality gap as [31]:

$$G(\bar{\mathbf{x}}) \in \left\langle 0, \hat{G}_{n,n_g}(\bar{\mathbf{x}}) + \frac{t_{1-\alpha}(n_g - 1)s_{n_g}(\bar{\mathbf{x}})}{\sqrt{n_g}} \right\rangle,$$

where $t_{1-\alpha}(n_g - 1)$ is the $(1-\alpha)$-quantile of the Student's distribution with $n_g - 1$ degrees of freedom, $s_{n_g}(\bar{\mathbf{x}}) = \sqrt{\frac{1}{n_g - 1}\sum_{j=1}^{n_g}(G_n^j(\bar{\mathbf{x}}) - \hat{G}_{n,n_g}(\bar{\mathbf{x}}))^2}$ is the sample standard deviation and $\epsilon = \frac{t_{1-\alpha}(n_g-1)s_{n_g}(\bar{\mathbf{x}})}{\sqrt{n_g}}$ is an error estimate.

In the subsequent text we will use aforementioned results to determine solution quality of our three engineering problems formulated in the previous chapter. Models of two PDE constrained optimization problems concerning the vibrations of string and console (see

section 6.1) can be written in the same compact form, and therefore, we will specify the form of optimality gap in all.

The true optimization problems (6.21)-(6.26) and (6.44)-(6.50) can be written as follows:

$$z^* = \min_{e \in X,\, g(\xi),\, v(\xi) \in Y} \mathbb{E}\left[F(\xi, e, g(\xi), v(\xi))\right]. \tag{7.8}$$

These true problems are then approximated by the following SAA problem (see (6.27)-(6.33), (6.60)-(6.66))

$$\hat{z}_n = \min_{\mathbf{e} \in X',\, \mathbf{g}_s,\, \mathbf{v}_s \in Y'} \frac{1}{n} \sum_{s=1}^{n} F(\xi_s, \mathbf{e}, \mathbf{g}_s, \mathbf{v}_s), \tag{7.9}$$

where the optimization over $\mathbf{e} \in X'$, $\mathbf{g}_s$, $\mathbf{v}_s \in Y'$ corresponds to constraints (6.28)-(6.33) and (6.61)-(6.66). Hence, the optimality gap in the candidate solution $\bar{\mathbf{e}}$ is

$$G(\bar{\mathbf{e}}) = \min_{\mathbf{g}(\xi), \mathbf{v}(\xi)} \mathbb{E}\left[F(\xi, \bar{\mathbf{e}}, \mathbf{g}(\xi), \mathbf{v}(\xi))\right] - z^*, \tag{7.10}$$

where $\mathbf{g}(\xi)$ and $\mathbf{v}(\xi)$ denote random vectors with realizations $\mathbf{g}_s$ and $\mathbf{v}_s$, $s = 1, \ldots, n$ and probabilities p_s, $s = 1, \ldots, n$ as discussed in previous chapter (the only exception is in notation of number of scenarios, n corresponds to R here).

Then the point estimate $\hat{G}_{n,n_g}(\bar{\mathbf{e}})$ of $G(\bar{\mathbf{e}})$ is

$$\hat{G}_{n,n_g}(\bar{\mathbf{e}}) = \frac{1}{n_g} \sum_{j=1}^{n_g} \left[\min_{\mathbf{g}(\xi), \mathbf{v}(\xi)} \frac{1}{n} \sum_{s=1}^{n} F(\xi^{sj}, \bar{\mathbf{e}}, \mathbf{g}(\xi^{sj}), \mathbf{v}(\xi^{sj})) - \right. \tag{7.11}$$

$$\left. - \min_{\mathbf{e}, \mathbf{g}(\xi), \mathbf{v}(\xi)} \frac{1}{n} \sum_{s=1}^{n} F(\xi^{sj}, \mathbf{e}, \mathbf{g}(\xi^{sj}), \mathbf{v}(\xi^{sj})) \right]. \tag{7.12}$$

$(1-\alpha)$-level confidence interval for the optimality gap is given as follows:

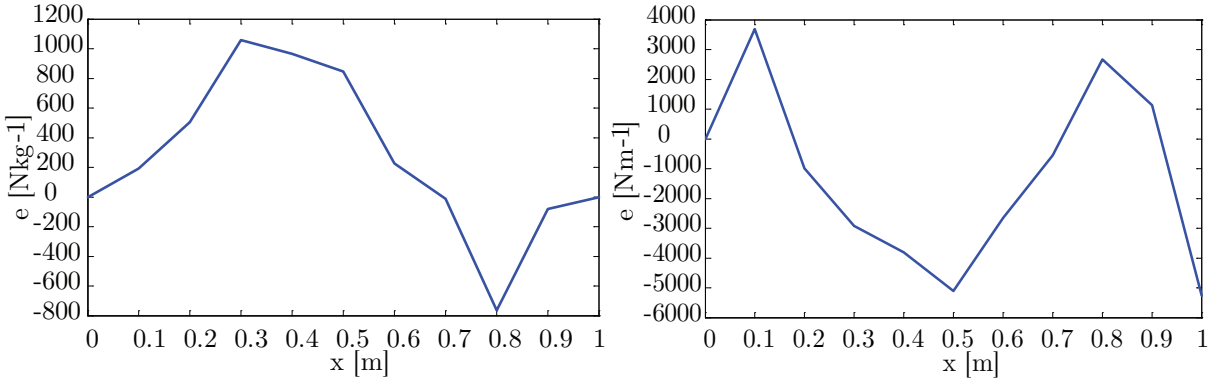
$$G(\bar{\mathbf{e}}) \in \left\langle 0, \bar{G}_{n,n_g}(\bar{\mathbf{e}}) + \frac{t_{1-\alpha}(n_g - 1) s_{n_g}(\bar{\mathbf{e}})}{\sqrt{n_g}} \right\rangle \tag{7.13}$$

with the same meaning of $t_{1-\alpha}(n_g - 1)$ and $s_{n_g}(\bar{\mathbf{e}})$ as before.

Number of batches has been chosen as $n_g = 30$ because this value should be high enough to ensure the normality of optimality gap supposed by central limit theorem. Furthermore, we have iteratively increased sample size ($n = 5, 10, \ldots, 100$, i. e. $n_s = 20$ different sample sizes) to see the behaviour of the optimality gap. Therefore, the total number of $n_g \cdot n_s = 600$ of SAA problems is solved.

Candidate solutions $\bar{\mathbf{e}}$ are computed via approximation models (7.9) with $n = 1000$ scenarios in both cases and are presented in the figure 7.1.

The numerical results related to the characteristics of optimality gaps are summarized in the tables 7.1, 7.2 and 7.3. Confidence level is $1 - \alpha = 0.95$. CPU time was about 8 min for PC with Core 2Duo 2x3GHz and 4GB RAM for console problem.



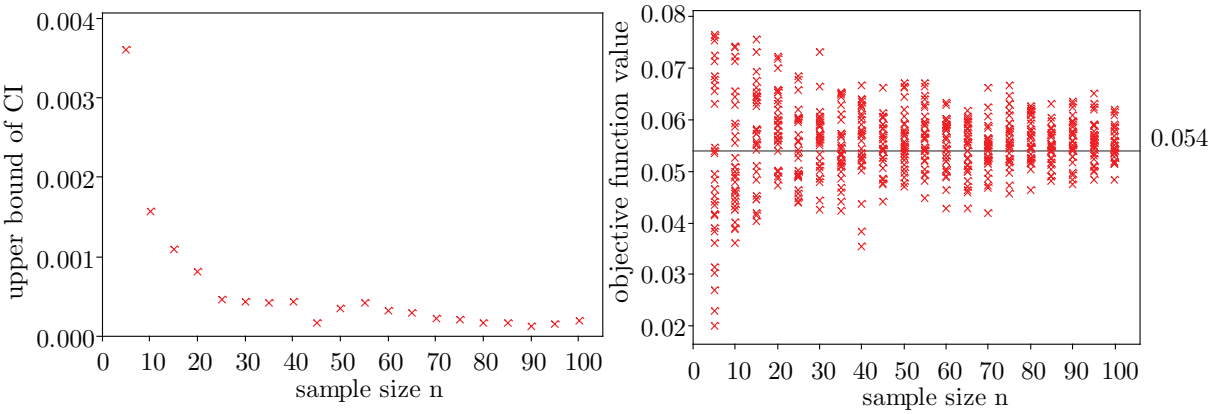
a) b)

Figure 7.1: Candidate solution $\bar{\mathbf{e}}$ for case of optimizing the vibrations of the a) string, b) console.

The point estimate of the objective value $\frac{1}{n_g}\sum_{j=1}^{n_g}\min_{\mathbf{g}(\xi),\mathbf{v}(\xi)\in Y'}\frac{1}{n_o}\sum_{s=1}^{n_o}F(\xi^{sj},\bar{\mathbf{e}},\mathbf{g}(\xi^{sj}),\mathbf{v}(\xi^{sj}))$ based on $n_o = 1000$ scenarios with $n_g = 30$ batches, i. e. based on scenarios independent of those used to find $\bar{\mathbf{e}}$, is 0.0557 with 95% confidence interval half-width equal to 0.0006. Width of confidence interval of the optimality gap then varies from $0,2\%$ to 6.5% of the point estimate of the objective function value [66]. It indicates good quality of candidate solution.

Batch size n	**Point estimate of** $G(\bar{\mathbf{e}})$	**Error estimate** ε	**Confidence interval of** $G(\bar{\mathbf{e}})$
5	0.00242	0.00119	[0;0.0036]
10	0.00120	0.00037	[0;0.0016]
15	0.00074	0.00034	[0;0.0011]
20	0.00053	0.00027	[0;0.0008]
25	0.00033	0.00012	[0;0.0005]
30	0.00028	0.00013	[0;0.0004]
35	0.00030	0.00011	[0;0.0004]
40	0.00032	0.00010	[0;0.0004]
45	0.00011	0.00004	[0;0.0002]
50	0.00024	0.00010	[0;0.0003]
55	0.00027	0.00013	[0;0.0004]
60	0.00022	0.00009	[0;0.0003]
65	0.00019	0.00009	[0;0.0003]
70	0.00013	0.00008	[0;0.0002]
75	0.00014	0.00006	[0;0.0002]
80	0.00011	0.00005	[0;0.0002]
85	0.00011	0.00004	[0;0.0002]
90	0.00008	0.00004	[0;0.0001]
95	0.00008	0.00005	[0;0.0001]
100	0.00013	0.00005	[0;0.0002]

Table 7.1: Summary of the optimality gap estimation – string problem.



a) b)

Figure 7.2: a) Upper bound of confidence interval of optimality gap, b) Variation of objective function values versus sample size – string problem.

The point estimate of the objective value $\frac{1}{n_g}\sum_{j=1}^{n_g} \min_{\mathbf{g}(\xi),\mathbf{v}(\xi)\in Y'} \frac{1}{n_o}\sum_{s=1}^{n_o} F(\xi^{sj}, \bar{\mathbf{e}}, \mathbf{g}(\xi^{sj}), \mathbf{v}(\xi^{sj}))$ based on $n_o = 1000$ scenarios with $n_g = 30$ batches is 0.0147 with 95% confidence interval half-width equal to 0.0002. Width of confidence interval of the optimality gap then varies from $0,2\%$ to 7.3% of the point estimate of the objective function value. It indicates good quality of candidate solution.

Batch size n	**Point estimate of** $G(\bar{\mathbf{e}})$	**Error estimate** ε	**Confidence interval of** $G(\bar{\mathbf{e}})$
5	0.00071	0.00035	[0;0.00107]
10	0.00034	0.00011	[0;0.00045]
15	0.00022	0.00010	[0;0.00032]
20	0.00016	0.00008	[0;0.00024]
25	0.00010	0.00004	[0;0.00013]
30	0.00008	0.00004	[0;0.00012]
35	0.00009	0.00003	[0;0.00012]
40	0.00009	0.00003	[0;0.00012]
45	0.00003	0.00001	[0;0.00004]
50	0.00007	0.00003	[0;0.00010]
55	0.00008	0.00004	[0;0.00012]
60	0.00006	0.00003	[0;0.00009]
65	0.00005	0.00002	[0;0.00008]
70	0.00004	0.00002	[0;0.00006]
75	0.00004	0.00002	[0;0.00006]
80	0.00003	0.00002	[0;0.00005]
85	0.00003	0.00001	[0;0.00004]
90	0.00002	0.00001	[0;0.00003]
95	0.00002	0.00001	[0;0.00004]
100	0.00004	0.00001	[0;0.00005]

Table 7.2: Summary of the optimality gap estimation – console problem.

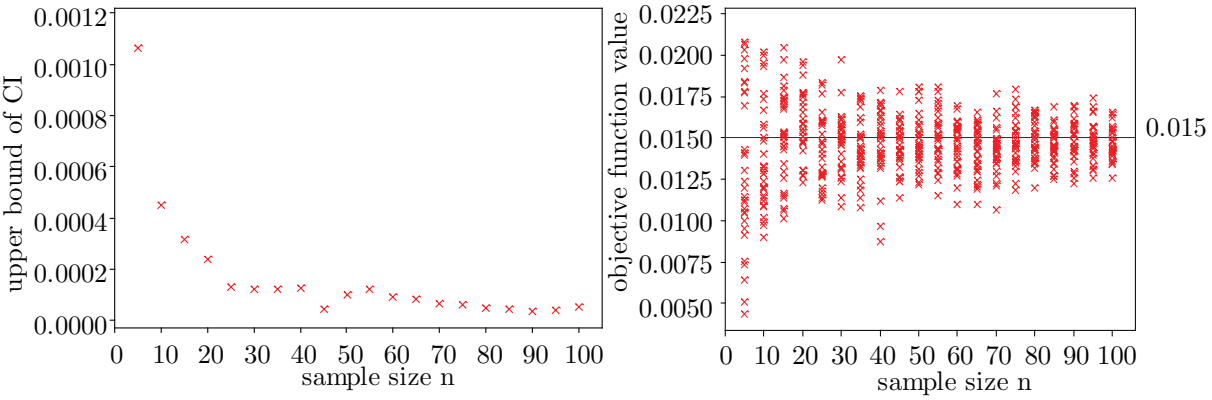


Figure 7.3: a) Upper bound of confidence interval of optimality gap, b) Variation of objective function values versus sample size – console problem.

Now we will present the same analysis for the ODE constrained problem related to optimum beam design (see section 6.3.1). The true optimization problem (6.104)-(6.112) is written as

$$z^* = \min_{\mathbf{a}\in X,\, \mathbf{v}(\xi)\in Y} \mathbb{E}[F(\xi, \mathbf{a}, \mathbf{v}(\xi))]. \tag{7.14}$$

This true problem is approximated by the following SAA problem (see ((6.124), (6.119)-(6.123))

$$\hat{z}_n = \min_{\mathbf{a}\in X',\, \mathbf{v}_s\in Y'} \frac{1}{n}\sum_{s=1}^{n} F(\xi_s, \mathbf{a}, \mathbf{v}_s), \tag{7.15}$$

where the optimization over $\mathbf{a} \in X'$ corresponds to constraints (6.122)-(6.123) and optimization over $\mathbf{v}_s \in Y'$ corresponds to constraints (6.119)-(6.121). Hence, the optimality gap in the candidate solution $\bar{\mathbf{a}}$ is

$$G(\bar{\mathbf{a}}) = \min_{\mathbf{v}(\xi)} \mathbb{E}[F(\xi, \bar{\mathbf{a}}, \mathbf{v}(\xi))] - z^*, \tag{7.16}$$

where $\mathbf{v}(\xi)$ denotes a random vector with realizations $\mathbf{V}_s$, $s = 1, \ldots, n$ and probabilities p_s, $s = 1, \ldots, n$ as discussed in previous section (the only exception is in notation of number of scenarios, n corresponds to R here).

The point estimate $\hat{G}_{n,n_g}(\bar{\mathbf{a}})$ of $G(\bar{\mathbf{a}})$ is:

$$\hat{G}_{n,n_g}(\bar{\mathbf{a}}) = \frac{1}{n_g}\sum_{j=1}^{n_g}\left[\min_{\mathbf{v}(\xi)} \frac{1}{n}\sum_{s=1}^{n} F(\xi^{sj}, \bar{\mathbf{a}}, \mathbf{v}(\xi^{sj})) - \min_{\mathbf{a},\mathbf{v}(\xi)} \frac{1}{n}\sum_{s=1}^{n} F(\xi^{sj}, \mathbf{a}, \mathbf{v}(\xi^{sj}))\right] \tag{7.17}$$

and $(1-\alpha)$-level confidence interval for the optimality gap is

$$G(\bar{\mathbf{a}}) \in \left\langle 0, \hat{G}_{n,n_g}(\bar{\mathbf{a}}) + \frac{t_{1-\alpha}(n_g-1)s_{n_g}(\bar{\mathbf{a}})}{\sqrt{n_g}} \right\rangle \tag{7.18}$$

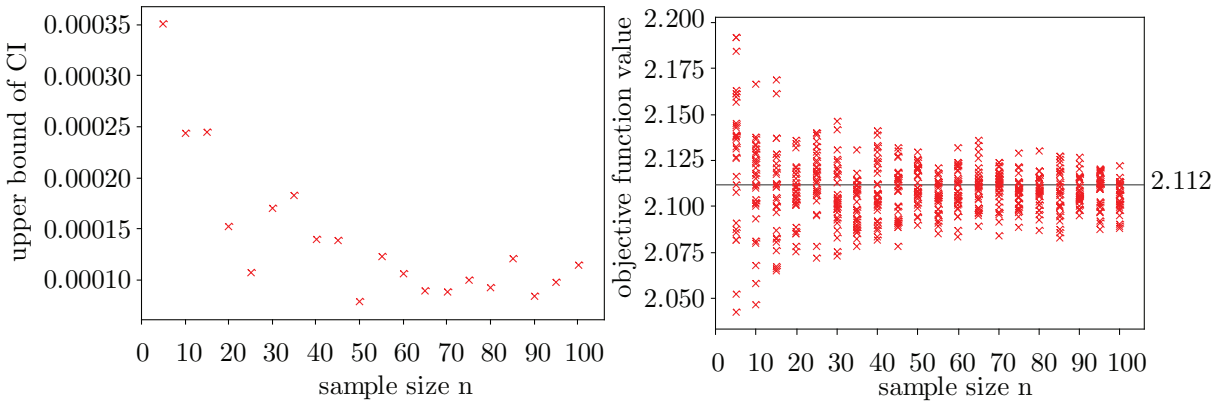
with the same meaning of $t_{1-\alpha}(n_g - 1)$ and $s_{n_g}(\bar{\mathbf{a}})$ as before.

Candidate solution $\bar{\mathbf{a}} = (a, b)^T = (22.4, 100)^T$ mm is computed via approximation model (7.15) with $n = 100$ scenarios. CPU time was about 66 min for PC with AMD Sempron 1.5 GHz and 496 MB RAM for this problem.

Batch size n	Point estimate of $G(\bar{\mathbf{a}})$	Error estimate ε	Confidence interval of $G(\bar{\mathbf{a}})$
5	0.00025	0.00010	[0;0.00035]
10	0.00016	0.00009	[0;0.00024]
15	0.00018	0.00007	[0;0.00025]
20	0.00011	0.00004	[0;0.00015]
25	0.00007	0.00004	[0;0.00011]
30	0.00013	0.00004	[0;0.00017]
35	0.00015	0.00004	[0;0.00018]
40	0.00011	0.00003	[0;0.00014]
45	0.00010	0.00003	[0;0.00014]
50	0.00006	0.00002	[0;0.00008]
55	0.00010	0.00002	[0;0.00012]
60	0.00008	0.00003	[0;0.00011]
65	0.00007	0.00002	[0;0.00009]
70	0.00006	0.00002	[0;0.00006]
75	0.00008	0.00002	[0;0.00010]
80	0.00007	0.00002	[0;0.00009]
85	0.00009	0.00003	[0;0.00012]
90	0.00007	0.00002	[0;0.00008]
95	0.00007	0.00002	[0;0.00010]
100	0.00009	0.00002	[0;0.00012]

Table 7.3: Summary of the optimality gap estimation – beam problem.

The point estimate of the objective function value $\frac{1}{n_g}\sum_{j=1}^{n_g}\min_{\mathbf{v}(\xi)\in Y'}\frac{1}{n_o}\sum_{s=1}^{n_o}F(\xi^{sj},\bar{\mathbf{a}},\mathbf{v}(\xi^{sj}))$ based on $n_o = 1000$ scenarios with $n_g = 30$ batches is 2.108 with 95% confidence interval half-width equal to 0.001. Width of confidence interval of the optimality gap then varies from 0.004% to 0.02% of the point estimate of the objective function value. It indicates very good quality of candidate solution.



a) b)

Figure 7.4: a) Upper bound of confidence interval of optimality gap, b) Variation of objective function values versus sample size – beam problem.

An engineer using the discussed technique can see how the width of confidence interval of the optimality gap roughly decreases with increasing sample size and our candidate solution approaches to the solution of true optimization problem (see figures 7.2 a), 7.3 a) and 7.4 a)). Furthermore, it can be seen from the figures 7.2 b), 7.3 b) and 7.4 b) how the variation of objective function values for fixed sample size also decreases with

increasing sample size and how the objective function values converge to the true objective function value. These true objective function values are approximated by the objective function values $\hat{z}_n$ computed for $n = 2000$ scenarios in case of PDE constrained problems, respectively $n = 1000$ scenarios in case of ODE constrained problem. They are displayed as the lines with associated values in the figures 7.2 b), 7.3 b) and 7.4 b).

7.3 Progressive hedging algorithm

Because complex engineering problems often lead to large optimization models we have tested a possibility of utilization of parallel computational techniques for the beam and console problems. Progressive hedging algorithm (PHA) proposed by Rockafellar and Wets [51], [69] has been chosen.

This algorithm was originally proposed for solving the optimization problems with partial information about unknown parameters. Therefore, at first we will describe its original form and then we will present its adjustment for application to our engineering problems.

When only limited information is available about the distribution of the random elements, stochastic programming approach is not an appropriate tool. In such cases, many practitioners rely on *scenario analysis* [51]. Uncertainty about parameters is modelled by a small number of versions of subproblems derived from an underlying optimization problem. These subproblems correspond to different *scenarios* s:

$$\min_{\mathbf{x}} f(\mathbf{x}, s) \text{ s. t. } \mathbf{x} \in C(s) \subset \mathbb{R}^n, \tag{7.19}$$

where $s \in S$. We assume that the optimal solution exists for all $s \in S$, denote it as $\mathbf{x}_s$ and call it scenario solution.

The purpose of studying the different subproblems and their optimal solutions is to discover trends and arrive at a "well hedged" solution to the underlying problem. This solution (called average solution) is computed by assigning weights p_s to scenario solutions (in fact it is their convex combination):

$$\hat{\mathbf{x}} := \sum_{s \in S} p_s \mathbf{x}_s, \tag{7.20}$$

where $p_s \geq 0$ and $\sum_{s \in S} p_s = 1$.

This average solution is expected to perform better under all scenarios than any particular scenario solution $\mathbf{x}_s$. But the solution that hedges against all eventualities should be obtained by solving the following stochastic optimization problem:

$$\min_{\mathbf{x}} \sum_{s \in S} p_s f(\mathbf{x}, s) \text{ s. t. } \mathbf{x} \in \bigcap_{s \in S} C(s), \tag{7.21}$$

where scenario s is assigned probability p_s. The optimal solution of this problem is denoted as $\mathbf{x}^*$.

The trouble is that problem (7.21) can be much larger and therefore much harder to solve than the scenario subproblems (7.19). Therefore, the progressive hedging algorithm, based on the principle of scenario aggregation, was developed [51]. This method generates a sequence of estimates $\{\hat{\mathbf{x}}^k,\ k = 1, \ldots\}$ which should converge to the optimal solution $\mathbf{x}^*$. The estimate at certain iteration is obtained by solving the modified version of (7.19) and aggregating these solutions into "compromise" solution whose robustness in the face of all eventualities is increasingly demanded.

The structure of the algorithm for one-stage models is the following [46], [69]:
Step 0: Initialize: set a price vector $\mathbf{w}_s^0 = \mathbf{0}$, the initial estimate $\hat{\mathbf{x}}^0 = \mathbf{0}$, choose a penalty parameter $\rho > 0$ and set $k = 1$.
Step 1: For each $s \in S$ solve the approximation problem (modified version of (7.19)):

$$\min_{\mathbf{x}} f(\mathbf{x}, s) + (\mathbf{w}_s^{k-1})^T \mathbf{x} + \frac{\rho}{2} \|\mathbf{x} - \hat{\mathbf{x}}^{k-1}\|^2 \text{ s. t. } \mathbf{x} \in C(s).$$

and denote the optimal solution as $\mathbf{x}_s^k$. $\|\cdot\|$ means the Euclidean norm on $\mathbb{R}^n$.
Step 2: Calculate estimate:

$$\hat{\mathbf{x}}^k = \sum_{s \in S} p_s \mathbf{x}_s^k$$

and update the perturbation term:

$$\mathbf{w}_s^k = \mathbf{w}_s^{k-1} + \rho(\mathbf{x}_s^k - \hat{\mathbf{x}}^k).$$

Return to Step 1 with $k = k + 1$.

The solution $\hat{\mathbf{x}}^k$ is optimal as soon as $\mathbf{x}_s^k = \hat{\mathbf{x}}^k$ for all s. Therefore we can use the distance

$$\sum_{s \in S} p_s \|\mathbf{x}_s^k - \hat{\mathbf{x}}^k\|^2 = \theta_k$$

as the terminating criterion. The procedure is repeated until $\theta_k < \varepsilon$ is satisfied, where ε is our chosen limit. This distance can be taken as a measure of how close we are from satisfying all constraints and it will tend to 0 for the convex case.

The advantage of this approach is that except for two simple calculations ($\mathbf{w}^k$ and $\hat{\mathbf{x}}^k$), the algorithm only requires the capability of solving individual scenario problems and their modified versions.

The progressive hedging algorithm is used not only for the optimization problems with partial information about unknown parameters but also for stochastic programming problems, where a probabilistic description of the unknown elements is available.

It is very useful for two-stage programs, which we are interested in, because it incorporates the nonanticipativity constraints $\mathbf{x}_s = \mathbb{E}(\mathbf{x}_s)$ for all $s \in S$ (see (5.25)) in the form of a penalty term into the objective function from (7.19). More precisely, it assigns some Lagrangian multipliers to these constraints and is slightly based on the augmented Lagrangian method. Strictly speaking, the progressive hedging algorithm is the version of dual decomposition methods and the advantage of this algorithm is that we obtain a separable program whose independent scenario subprograms can be solved in parallel way.

Let us assume the three-objective beam problem formulated in the section 6.3.1. Denote the first-stage decision variable (i. e. dimensions of the cross section) as $\mathbf{a} = (a, b)^T = = (a_1, a_2)^T$.
The structure of PHA for our beam problem is the following:
Step 0: Set $\mathbf{w}_s^{(0)} = \mathbf{0}$, choose $\hat{\mathbf{a}}^{(0)}$, penalty parameter $\rho > 0$ and tolerance ε, set $k = 1$.
Step 1: For all $s = 1, \ldots, R$ solve the approximation program:

$$\min_{\mathbf{a}, \mathbf{V}_s} F_s(\xi_s, \mathbf{a}, \mathbf{V}_s) + (\mathbf{w}_s^{(k-1)})^T \mathbf{a} + \frac{\rho}{2} \|\mathbf{a} - \hat{\mathbf{a}}^{(k-1)}\|^2,$$

where $F_s(\xi_s, \mathbf{a}, \mathbf{V}_s)$ is the objective function value of sth scenario subprogram of (6.124), (6.119)-(6.123). Denote optimal solution as $\mathbf{a}_s^{(k)}$.

Step 2: Compute the estimate:

$$\hat{\mathbf{a}}^{(k)} = \sum_{s=1}^{R} p_s \mathbf{a}_s^{(k)}$$

and update the weight vector:

$$\mathbf{w}_s^{(k)} = \mathbf{w}_s^{(k-1)} + \rho(\mathbf{a}_s^{(k)} - \hat{\mathbf{a}}^{(k)}).$$

Step 3: If the termination inequality $\|\hat{\mathbf{a}}^{(k)} - \hat{\mathbf{a}}^{(k-1)}\|^2 + \sum_{s=1}^{R} p_s \|\mathbf{a}_s^{(k)} - \hat{\mathbf{a}}^{(k)}\|^2 \leq \varepsilon$ defined by [21] is satisfied, then the solution $\hat{\mathbf{a}}^{(k)}$ is optimal with given tolerance ε, otherwise set $k = k + 1$ and return to step 1.

The termination criterion tries to minimize the difference between successive estimates and forces the solutions for different scenarios to be the same.

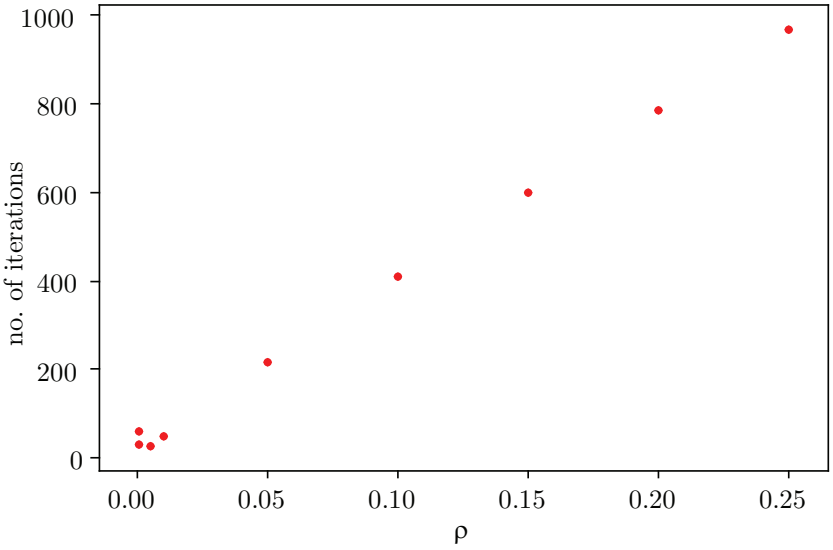
The values of parameters are the same as in section 6.3.1 excepting the number of scenarios. We have tested PHA with $R = 10$ scenarios instead of 100 scenarios because of bigger computational complexity for the test non-parallel implementation. The initial estimate for dimensions is $\hat{\mathbf{a}}^{(0)} = (100; 100)\,\text{mm}$ which corresponds to maximum rigidity. Tolerance is set to $\varepsilon = 10^{-6}$ because it roughly conforms to the accuracy of one decimal place in length which is fully sufficient in engineering practice.

The optimal dimensions are $a = 22.5\,\text{mm}$, $b = 100\,\text{mm}$. It can be seen from the figures 7.5-7.8 and table 7.4 that the penalty parameter ρ plays the key role for the computational process convergence properties of the algorithm. Unfortunately, there is no exact rule how to determine the best value of this parameter ρ. We have estimated that for our example the best value lies within the interval $(0.001; 0.01)$ (see table 7.4). For larger values of ρ, the convergence process will take much more time. Program was implemented in GAMS and ran on a notebook with Intel Core 2Duo 2GHz and 2GB RAM and corresponding CPU times given in the table 7.4.

Parameter ρ	**No. of iterations**	**CPU time** [min]
0.0005	59	1
0.001	32	0.7
0.005	28	0.5
0.01	50	1
0.05	215	4
0.1	410	9
0.15	599	15
0.2	784	22
0.25	966	38

Table 7.4: Convergence properties of PHA.

Convexity assumptions from the convergence theorem A1 are not fulfilled for this problem though the constraints (6.119)-(6.123) are linear, and hence, the feasible set is convex. It is because of that the objective function (6.124) is not convex as is shown in the page 51. But the achieved results demonstrate that the PHA can be used even when the mathematical conditions for convergence are not respected.



There is certain value of parameter ρ for which the algorithm converges fastest (see figure 7.5). For smaller and larger values the convergence process is slowing down.

Figure 7.5: Number of iterations versus penalization parameter ρ.

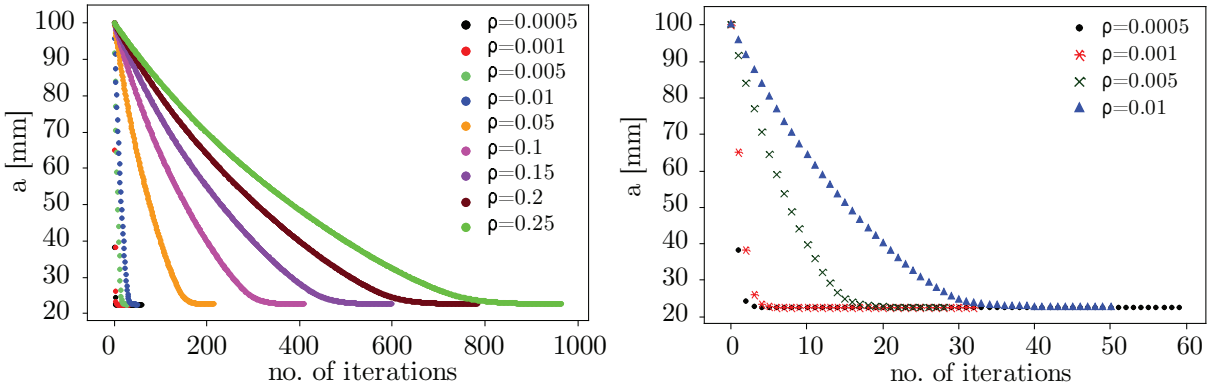


Figure 7.6: Convergence of the beam dimension a.

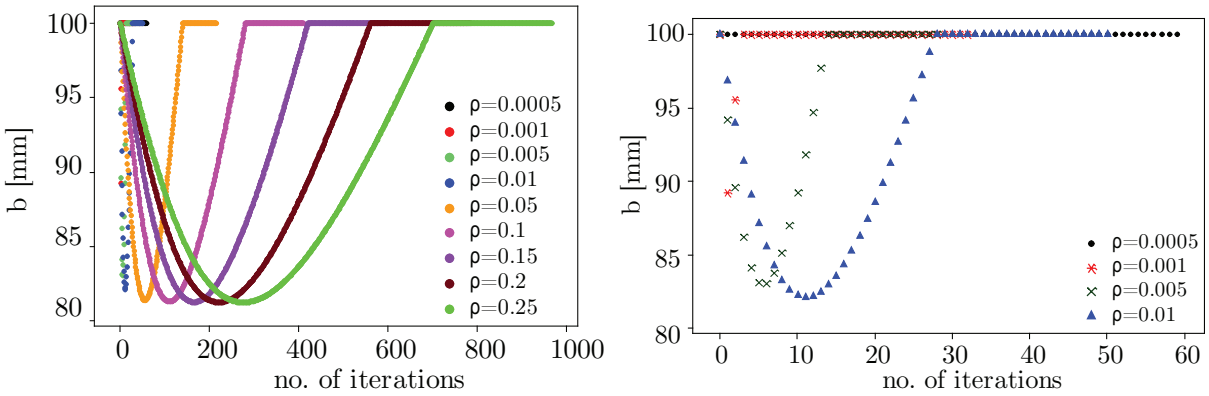


Figure 7.7: Convergence of the beam dimension b.

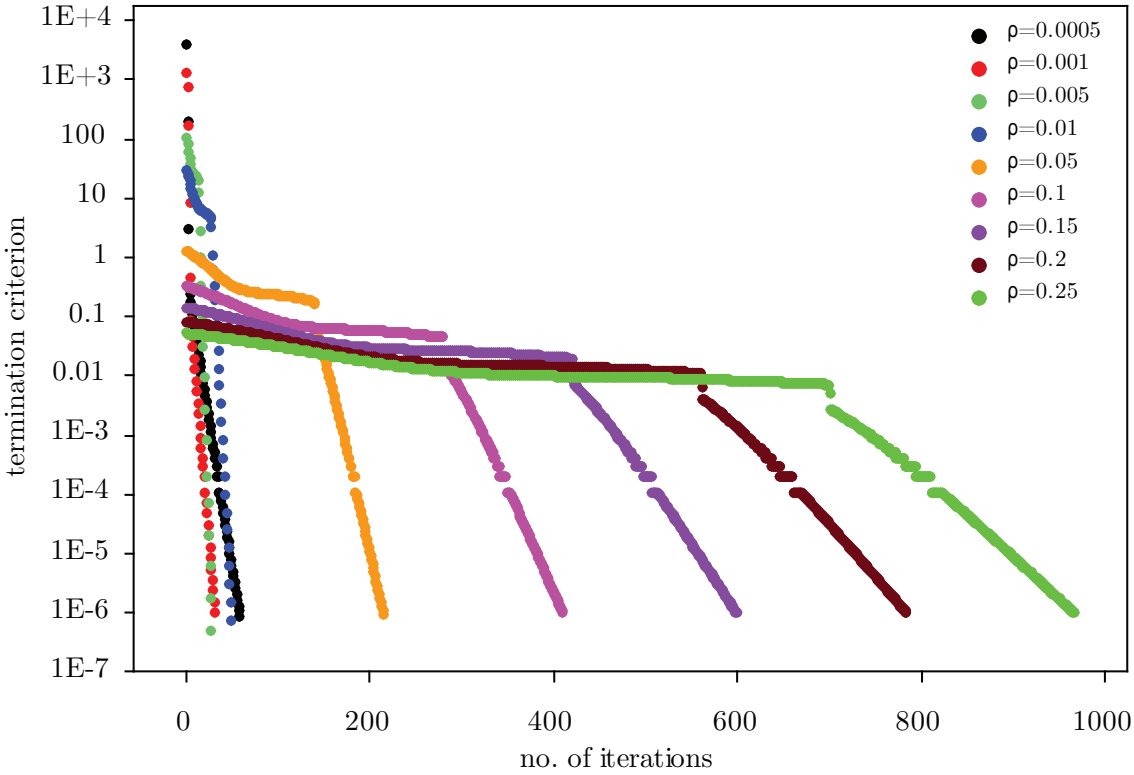


Figure 7.8: Convergence of the termination criterion value – beam problem. The scale of the y-axis is logarithmic.

We have also tested PHA for the PDE constrained stochastic programming problem concerning the vibrations of the console (see the section 6.1.2). Denote the first-stage variable as $\mathbf{e} = (e_0, \ldots, e_N)^T$. The structure of PHA for this problem is the following:
Step 0: Set $\mathbf{w}_s^{(0)} = \mathbf{0}$, choose $\hat{\mathbf{e}}^{(0)}$, penalty parameter $\rho > 0$ and tolerance ε, set $k = 1$.
Step 1: For all $s = 1, \ldots, R$ solve the approximation program:

$$\min_{\mathbf{e},\mathbf{g}_s,\mathbf{V}_s} F_s(\xi_s, \mathbf{e}, \mathbf{g}_s, \mathbf{V}_s) + (\mathbf{w}_s^{(k-1)})^T\mathbf{e} + \frac{\rho}{2}\|\mathbf{e} - \hat{\mathbf{e}}^{(k-1)}\|^2$$

where $F_s(\xi_s, \mathbf{e}, \mathbf{g}_s, \mathbf{V}_s)$ is the objective function value of sth scenario subprogram of (6.60)-(6.66). Denote optimal solution as $\mathbf{e}_s^{(k)}$.
Step 2: Compute the estimate:

$$\hat{\mathbf{e}}^{(k)} = \sum_{s=1}^{R} p_s \mathbf{e}_s^{(k)}$$

and update the weight vector:

$$\mathbf{w}_s^{(k)} = \mathbf{w}_s^{(k-1)} + \rho(\mathbf{e}_s^{(k)} - \hat{\mathbf{e}}^{(k)}).$$

Step 3: If the termination inequality $\|\hat{\mathbf{e}}^{(k)} - \hat{\mathbf{e}}^{(k-1)}\|^2 + \sum_{s=1}^{R} p_s\|\mathbf{e}_s^{(k)} - \hat{\mathbf{e}}^{(k)}\|^2 \leq \varepsilon$ is satisfied then the solution $\hat{\mathbf{e}}^{(k)}$ is optimal with given tolerance ε, otherwise set $k = k+1$ and return to step 1.

The algorithm has been again implemented in GAMS with the same values of parameters as in the section 6.1.2 excepting the number of scenarios which is $R = 10$ scenarios.

But the results are not so satisfying as in the aforementioned beam problem. We have come to the following conclusions [76]. The choice of the penalty parameter ρ again plays the key role for the convergence properties of the algorithm. Furthermore, we have found out that for this certain example the ρ value has to be bounded from above, otherwise the solution is infeasible. In addition, this bounding value depends on the choice of the initial iteration $\hat{\mathbf{e}}^0$. For decreasing value of ρ and the fixed number of iterations, the achieved accuracy also decreases. Furthermore, the value of the termination criterion is constant for fixed ρ value. Therefore, we have achieved the largest accuracy ($3 \cdot 10^{-18}$ in 8 iterations) by using heuristic (the part of the aim no. 3) in the form $\rho^k = 2\rho^{k-1}$, $\rho^0 = B$. For bigger values of B, $B < 10$, less iterations are needed. The progress of termination criterion values is presented in the figure 7.9 for $B = 9$ and $\hat{\mathbf{e}}^0 = 0$.

The convergence of PHA with this heuristic is conserved due to the fact that from the certain iteration k^* the ρ value is fixed (it has to be bounded from above because of the feasibility) and hence the theorem A1 of Rockafellar and Wets with fixed value of ρ can be used. Convexity assumptions from this theorem are fulfilled for this console problem, because the objective function (6.60) is quadratic and therefore convex and the constraints (6.61)-(6.66) are linear, hence the feasible set is also convex.

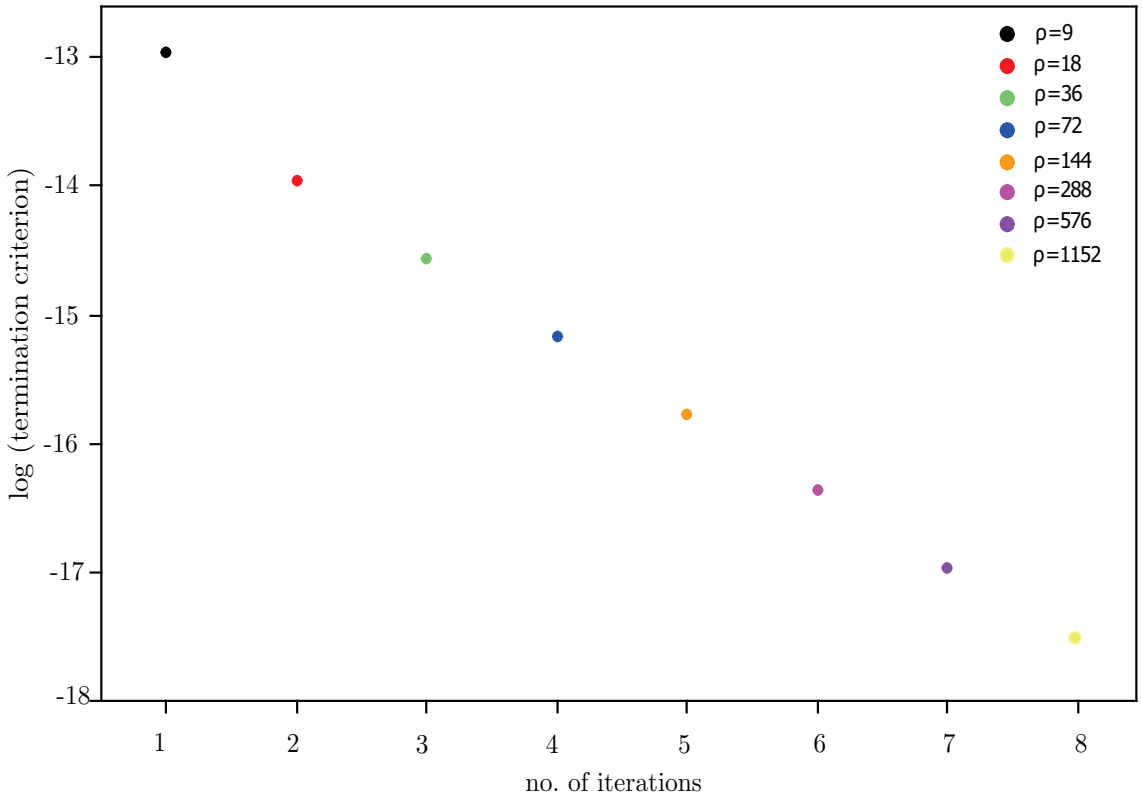


Figure 7.9: Convergence of the termination criterion value – console problem.

7.4 Comparison of FDM and FEM – case study

We have been asked by potential users of the proposed computational scheme, whether our approach with simple discretization method and algebraic modelling system GAMS provides results comparable with results from "black-box" like systems widely used by

engineers. We have compared GAMS implementation involving finite difference method (FDM) and ANSYS 11.0 (Ansys Inc., Canonsburg, PA, USA) model based on finite element method (FEM). This comparison is made for the deterministic version of our civil engineering optimization problem (6.145), (6.140)-(6.144) with $\alpha = \beta = 0.5$ and $E = 2.1 \cdot 10^5$ MPa.

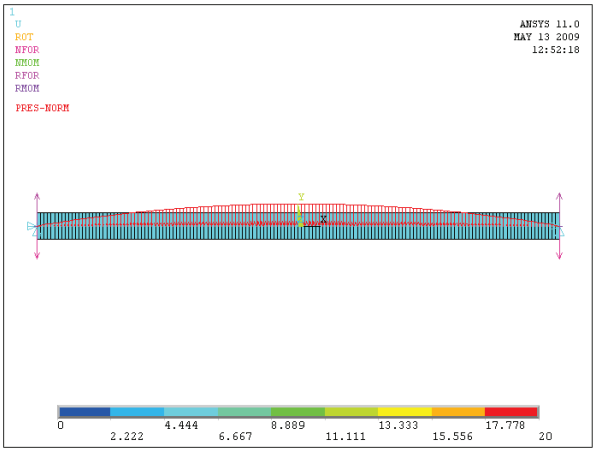


Figure 7.10: Load and elements for FEM.

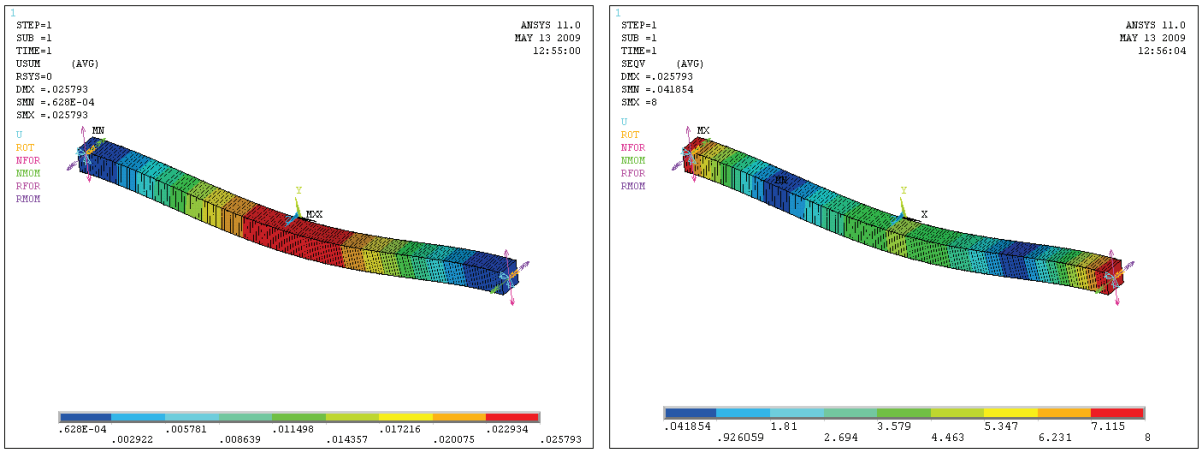
The optimal solution obtained by GAMS with FDM is $a = 10\,\text{mm}$, $b = 89.4\,\text{mm}$, $z = 0.47$ and the optimal solution from ANSYS with FEM is $a = 11.2\,\text{mm}$, $b = 84.5\,\text{mm}$, $z = 0.49$. Convergence of the optimization process in ANSYS is presented in the figure 7.11, where the optimal solution is obtained in the third iteration. The results are slightly better from GAMS but there is only a small difference between them and the results computed by ANSYS.

```
LIST OPTIMIZATION SETS FROM SET 1 TO SET 4 AND SHOW
ONLY OPTIMIZATION PARAMETERS. (A '*' SYMBOL IS USED TO
INDICATE THE BEST LISTED SET)

                      SET 1         SET 2       *SET 3*      SET 4
                 (INFEASIBLE)   (FEASIBLE)   (FEASIBLE)   (FEASIBLE)
MAX_NAPETI(SV) > 16.000          99.000       100.00       96.000
A          (DV)   50.000         11.301       11.184       11.630
B          (DV)   100.00         84.456       84.451       84.450
UCEL       (OBJ)  2.5357         0.49771      0.49252      0.51217
```

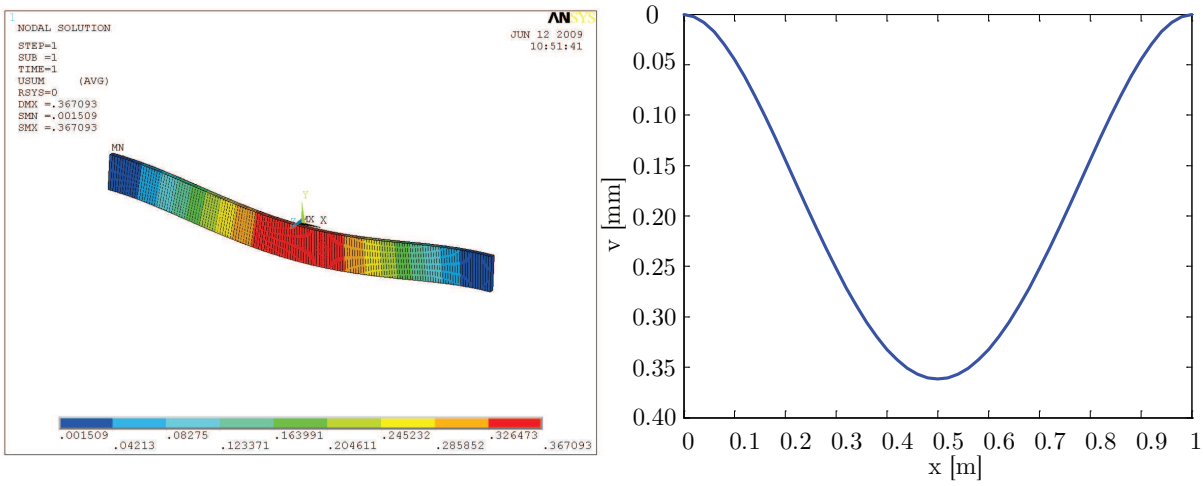
Figure 7.11: Convergence of the optimization process in ANSYS.

The deflection and maximum stress computed by ANSYS in nonoptimized case are presented in the figure 7.12. It can be seen that maximum stress is only 8 MPa at the end points while 100 MPa is permitted. Furthermore, maximum deflection is only 0.03 mm in the middle of the beam which is uselessly small. Therefore, the initial square cross-section is very inconvenient and the optimization to rectangular cross-section is needed. The deflection in optimized cases is quantitatively and qualitatively same for both computing systems and discretization methods (see figure 7.13). Maximum deflection of 0.37 mm occurs in the middle of the beam while it decreases towards the beam's ends. Also the maximum stress in optimized cases is quantitatively and qualitatively nearly the same for both computing systems and discretization methods (see figure 7.14). Difference is only in signs – absolute value of stress is plotted in ANSYS while both positive and negative values are plotted in GAMS. Maximum tensile stress of about 100 MPa occurs at the ends of the beam while maximum compression stress of about -54 MPa occurs in the middle of the beam.



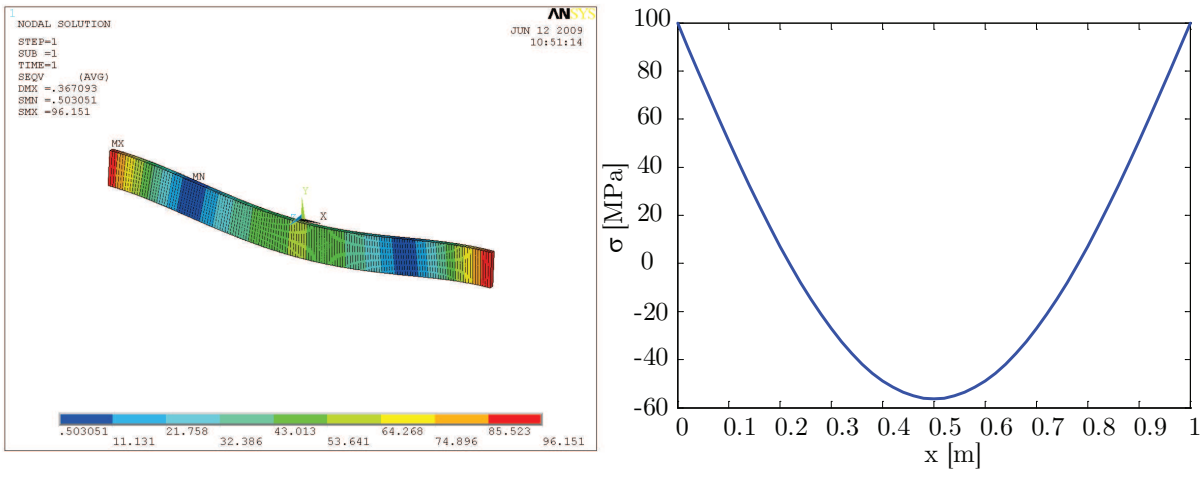
a) b)

Figure 7.12: a) Deflection and b) Maximum stress computed by ANSYS in nonoptimized case.



a) b)

Figure 7.13: Deflection computed by a) ANSYS and b) GAMS.



a) b)

Figure 7.14: Maximum stress computed by a) ANSYS and b) GAMS.

Chapter 8

Conclusions

The applicability of two-stage stochastic programming approach to three engineering problems with random parameters involved has been discussed. The first problem has been concerned in the vibrations of the string loaded by random force, the second problem has been related to the vibrations of the console also loaded by random force and the last problem has been concerned with the optimal design of beam dimensions with random Young's modulus. The first two cases have led to the PDE constrained stochastic quadratic programs while the third one has made for the ODE constrained multi-objective stochastic nonlinear program.

In general, the proposed computational scheme consisting of scenario-based two-stage stochastic program, modelling language implementation, parallelism, solution quality evaluation and verification of results by the FEM solver, seems robust enough for future applications to similar and advanced optimum design problems. There is also a future challenge to motivate engineers to use the proposed approach, because they may still prefer "black-box" like computing system where they choose appropriate preprogrammed mathematical model.

The modelling-based approximation approach focusing on suboptimal solution search has allowed us to avoid difficulties with the huge amounts of input data required and problems with implementation of various algorithms that often appear in the case of real-world applications of stochastic optimal control related models. The choice of the models has been suitable for the implementations in modelling languages such as GAMS. It has been also proved for which type of problems stochastic programming approach (EO reformulation) should be used and when it is sufficient to solve simpler deterministic problem (EV reformulation). This fact has the big importance in practice in term of computational intensity of large scale problems.

The solution quality has been tested by presented Monte-Carlo bounding technique and satisfactory results have been obtained. The progressive hedging algorithm as a representative of scenario decomposition methods has been implemented and tested with respect to future possibilities of parallel computing of large engineering problems. The implementation of this algorithm has shown that it can be used even when the mathematical conditions for the convergence are not fulfilled.

Finally, the results of model approximated by finite difference method and implemented in GAMS have been compared with model using finite element method implemented in ANSYS and the quite comparable results have been obtained.

It is possible to express chance constraints with corresponding random variable with finite support in any algebraic modelling language, but without modelling language support, it requires reformulating problem by introducing extra constraints and binary vari-

ables as was described in the section 6.3.3 or by using the appropriate penalty function (see also the section 6.3.3 for very interesting results of comparison between MINLP and penalty reformulation). But the extensions of classical algebraic modelling language have been originating recently, e. g. stochastic AMPL (SAMPL) as an extension of AMPL for stochastic programming [63]. These extensions support formulation of scenario-based models and allow direct solution of chance constrained problems without reformulation leading to really big computational intensity. Therefore, it could be useful to try to implement the studied engineering problems with more scenarios in this type of extended modelling languages in further research.

The progressive hedging algorithm has been implemented and tested with respect to the future possibilities of parallel computing of large engineering problems. Its main disadvantage is that the performance of the algorithm is very sensitive to the choice of the penalty parameter ρ and that there is no general rule how to determine the best value of this parameter. This problem could be at least partially solved by the suggestions from Watson and Woodruff in [67]. They have proposed that the convergence performance of the PHA can be improved by having separate ρ for each first-stage variable with values proportional to element unit cost. The advantages of their approach are its problem-independent nature and that it is parameter-free. Therefore, the repeated executions of the PHA because of the search for appropriate value of ρ are not needed.

Another future research aim could concern in the practical parallel implementation of the PHA. New version of GAMS offers so called grid computing facility [8] which allows to handle the multiple CPUs and enables to divide large scale problems into several subproblems and solve them in parallel in reasonable times.

Bibliography

[1] AMES, W. F.: *Numerical Methods for Partial Differential Equations (Computer Science and Scientific Computing).* 2nd ed. New York: Academic Press, 1977. 365 p. ISBN 0-12-056760-1.

[2] BAŤA, M. TRÁVNÍČEK, F., PLACHÝ, V.: *Dynamics of structures* (in Czech). 1st ed. Praha: SNTL, 1987. 448 p.

[3] BAZARAA, M. S., SHERALI, H. D., SHETTY, C. M.: *Nonlinear Programming: Theory and Algorithms.* 2nd ed. New York: Wiley and Sons, 1993. 638 p. ISBN 0-471-55793-5.

[4] BETTS, J. T.: *Practical Methods For Optimal Control Using Nonlinear Programming.* London: SIAM, 2001. 190 p. ISBN 978-0898714883.

[5] BIEGLER, L.T., GHATTAS, O. et al.: *Real-Time PDE-Constrained Optimization.* 1st ed. SIAM, 2007. 312 p. ISBN 978-0898716214.

[6] BIRGE, J., LOUVEAUX, F. *Introduction to Stochastic Programming.* 1st ed. New York: Springer, 1997. 421 p. ISBN 0-387-98217-5.

[7] BRANDA, M., DUPAČOVÁ, J.: Approximations and contamination bounds for probabilistic programs. *Stochastic Programming E-Print Series (SPEPS) 13* (2008), 1–25.

[8] BROOKE, A., KENDRICK, D., MEERAUS, A., RAMAN, R.: *GAMS – A User's Guide.* Washington, DC: GAMS Development Corporation, 2008. 281 p.

[9] CHOBOT, K., BENDA, J., HÁJEK, V., NOVOTNÁ, H.: *Statics of structures II* (in Czech). Praha: SNTL, 1983. 344 p. ISBN 04-714-83.

[10] CONTI, S., HELD, H., PACH, M., RUMPF, M., SCHULTZ, R.: Shape Optimization Under Uncertainty – A Stochastic Programming Perspective. *SIAM J. Optim. 19*, 4 (2009), 1610–1632. ISSN 1052-6234.

[11] CRESPO, L. G., SUN, J. Q.: *Stochastic Optimal Control via Bellman's Principle.* NASA/CR-2003-212419, NIA Report No. 2003-04, USA, 2003.

[12] DANTZIG, G. B.: Linear programming under uncertainty. *Management Science 1* (1955), 197–206. ISSN 0025-1909.

[13] DRUD, A. *CONOPT.* Bagsvaerd: ARKI Consulting and Development A/S. [cited 2008-04-30]. Available at: http://www.gams.com/dd/docs/solvers/conopt.pdf

[14] DUPAČOVÁ, J.: The minimax problem of stochastic linear programming and the moment problem. *Ekonom.-Mater. Obzor 13* (1977), 279–307.

[15] DUPAČOVÁ, J., HURT, J., ŠTĚPÁN, J.: *Stochastic modelling in economics and finance.* Dordrecht: Kluwer Academic Publishers, 2002. 386 p. ISBN 1-4020-0840-6.

[16] FELLER, W.: *An Introduction to Probability Theory and Its Applications, Vol. 1.* 3rd ed. Wiley, 1968. 509 p. ISBN 978-0471257080.

[17] FLEMING, W. H., RISHEL, R. W.: *Deterministic and Stochastic Optimal Control (Stochastic Modelling and Applied Probability, vol. 1).* Springer, 1982. 236 p. ISBN 978-0387901558.

[18] GUŠTAR, M.: *Modern methods of system simulation: SBRA method* (in Czech). [PhD Thesis], Prague: Technical University of Ostrava, 2002.

[19] HASLINGER, J., MÄKINEN, R. A. E.: *Introduction to Shape Optimization: Theory, Approximation, and Computation (Advances in Design and Control).* SIAM, 2003. 273 p. ISBN 0898715369.

[20] HELD, H.: *Shape Optimization under Uncertainty from a Stochastic Programming Point of View.* [PhD Thesis], Duisburg: Universität Duisburg-Essen, 2009.

[21] HELGASON, T., WALLACE, S. W.: Approximate scenario solutions in the progressive hedging algorithm. *Annals of Operations Research 31*, 1 (1991), 425–444. ISSN 0254-5330.

[22] ISSMO working group: Optimization in biomechanics. [cited 2009-09-25]. Available at: http://biopt.ippt.gov.pl

[23] JAMESON, A., MARTINELLI, L., PIERCE, N. A.: Optimum Aerodynamic Design Using the Navier-Stokes Equations. *Theoretical and Computational Fluid Dynamics 10*, 1–4 (1998), 213–237. ISSN 0935-4964.

[24] JANÍČEK, P.: *Mechanics of materials I* (in Czech). Brno: CERM, 2004. 287 p. ISBN 80-214-2592-X.

[25] KALL, P.: *Stochastic linear programming.* Berlin: Springer-Verlag, 1976. 95 p.

[26] KALL, P., WALLACE, S. W.: *Stochastic Programming.* 1st ed. New York: Wiley and Sons, 1995. 320 p. ISBN 978-0471951087.

[27] KAVIČKA, F., ŠTĚTINA, J., SEKANINA, B., STRÁNSKÝ, K., DOBROVSKÁ, J., HEGER, J.: The optimization of a concasting technology by two numerical models. *Journal of Materials Processing Technology 185*, 1–3 (2007), 152–159. ISSN 0924-0136.

[28] KIRSCH, U.: *Structural Optimization.* Springer-Verlag, 1993. 302 p. ISBN 978-3540559191.

[29] LANÍKOVÁ, I., ŠTĚPÁNEK, P.: Optimisation of prestressed poles from spun concrete. In Proceedings of the International Conference *Life cycle assessment, optimisation, behaviour and properties of concrete structures*, Brno: CERM, 2008, pp. 180–185. ISBN 978-80-214-37-9.

[30] LEE, E. B., MARKUS, L.: *Foundations of Optimal Control Theory.* 1st ed. New York: Wiley and Sons, 1967. 576 p.

[31] MAK, W., MORTON, D., WOOD, R. K.: Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters 24* (1999), 47–56. ISSN 0167-6377.

[32] MAREK, P., BROZETTI, J., GUŠTAR, M. (eds.): *Probabilistic Assessment of Structures using Monte Carlo Simulation: Background, Excercises and Software.* Prague: ÚTAM AS CR, 2001. 471 p. ISBN 978-8086246086.

[33] MATHEWS, J. H., FINK, K. D.: *Numerical Methods Using Matlab.* 4th ed. New Jersey: Pearson Prentice Hall, 2004. 680 p. ISBN 0-13-065248-2.

[34] MATOUŠEK, R., ŽAMPACHOVÁ, E.: Promising GAHC and HC12 algorithms in global optimization tasks. Accepted in *Optimization methods and software.* 2010.

[35] MAUDER, T.: Optimization methods for the secondary cooling zone of a continuous casting process of steel slabs. *Strojárstvo/Strojírenství* (2009), 175–176. ISSN 1335-2938.

[36] MAUDER, T., KAVIČKA, F., ŠTĚTINA, J., FRANĚK, Z., MASARIK, M.: A mathematical & stochastic modelling of the concasting of steel slabs. In 18th International Conference on Metallurgy and Materials, Tanger, 2009, pp. 41–48. ISBN 978-80-87294-10-9.

[37] MRÁZEK, M., FLORIAN, Z., VESELÝ, R., BORÁK, L., MARCIÁN, P.: Strain-Stress Analysis of Lower Limb with Applied Fixator. In Book of Extended Abstracts from 25th Conference *Computational Mechanics*, Pilsen: University of West Bohemia, 2009, pp. 97–98. ISBN 978-80-7043-824-4.

[38] NEMEC, M., ZINGG, D. W.: Multi-Point and Multi-Objective Aerodynamic Shape Optimization. In 9th AIAA/ISSMO Symposium on *Multidisciplinary Analysis and Optimization*, Atlanta, 2002.

[39] NOVÁK, D., BERGMEISTER, K., PUKL, R., ČERVENKA, V.: Structural assessment and reliability analysis for existing engineering structures, theoretical background. *Structure and infrastructure engineering 5*, 4 (2009), 267–275. ISSN 1573-2479.

[40] PFLUG, G. Ch.: Stochastic Optimization in Finance. Lecture notes from *Stochastic programming school* in Bergamo. 2009.

[41] PLŠEK, J., ŠTĚPÁNEK, P.: Optimization of design of cross-section in concrete structures. In Proceedings of the 4th International Conference *Concrete and Concrete Structures*. Žilina: University of Žilina, 2005, pp. 325–330. ISBN 80-8070-462-7.

[42] PLŠEK, J., ŠTĚPÁNEK, P.: Optimisation of reinforced concrete cross-section. *Engenharia Estudo e Pesquisa 1* (2007), 24–36. ISSN 1415-3025.

[43] PLŠEK, J., ŠTĚPÁNEK, P., POPELA, P.: Deterministic and reliability based structural optimization of concrete cross-section. *Journal of Advanced Concrete Technology 5*, 1 (2007), 63–74. ISSN 1346-8014.

[44] POLYANIN, A. D.: *Handbook of Linear Partial Differential Equations for Engineers and Scientists.* 1st ed. Chapman & Hall/CRC, 2001. 800 p. ISBN 978-1584882992.

[45] POPELA, P.: *An Objected-Oriented Approach to Multistage Stochastic Programming.* [PhD Thesis], Prague: Charles University, 1998.

[46] POPELA, P.: Numerical Techniques and Available Software. In *Stochastic Modeling in Economics and Finance* (authors: Dupačová, J., Hurt, J., Štěpán, J.), Chapter 8 in Part II, Kluwer Academic Publishers, 2002, pp. 206–227.

[47] POPELA, P.: Stochastic programming. Lecture notes from University of Malta. 2004.

[48] POPELA, P.: Integer and dynamic programming. Lecture notes from University of Malta. 2004.

[49] RARDIN, R. L.: *Optimization in operations research.* New Jersey: Prentice Hall, 1998. 919 p. ISBN 0-02-398415-5.

[50] ROBERT, C. P., CASELLA, G.: *Monte Carlo Statistical Methods.* 2nd ed. New York: Springer, 2005. 645 p. ISBN 978-0387212395.

[51] ROCKAFELLAR, R. T., WETS, R. J-B: *Scenarios and Policy Aggregation in Optimization Under Uncertainty.* IIASA Working Paper WP-87-119, Austria, 1987.

[52] RUSZCZYŃSKI, A., SHAPIRO, A. (ed.): *Handbooks in Operations Research and Management Science, vol. 10: Stochastic Programming.* 1st ed. Amsterdam: Elsevier, 2003. 688 p. ISBN 0-444-50854-6.

[53] SAHINIDIS, N. V.: *BARON: Branch And Reduce Optimization Navigator, User's Manual.* University of Illinois at Urbana-Champaign, USA, [cited 2009-03-26]. Available at: http://www.andrew.cmu.edu/user/ns1b/baron/manuse.pdf

[54] SMITH, G. D.: *Numerical solution of partial differential equations: finite difference methods.* 3rd ed. Oxford: Oxford University Press, 1985. 337 p. ISBN 0-19-859650-2.

[55] STEUER, R. E.: *Multiple Criteria Optimization: Theory, Computation and Application.* New York: John Wiley, 1986. 568 p. ISBN 978-0471888468.

[56] ŠARLEJ, M., PETR, P., HÁJEK, J., STEHLÍK, P.: Computational support in experimental burner design optimisation. *Applied Thermal Engineering 27*, 16 (2007), 2727–2731. ISSN 1359-4311.

[57] ŠTĚPÁNEK, P., LANÍKOVÁ, I.: Principles of optimization of reinforcement design of RC structure (in Czech). *Stavební obzor 9*, 6 (2000), pp. 177–184. ISSN 1210-4027.

[58] ŠTĚPÁNEK, P.: Optimisation of longitudinal reinforcement of concrete framed structures using stochastic finite element method. In proceedings of the 8th conference *Structural safety and reliability.* Colorado, 2001, pp. 278–285.
ISBN: 90-5809-197-X.

[59] ŠTĚPÁNEK, P.: Optimization of concrete structures. In 33rd Conference on *Our World in Concrete & Structures.* Singapore: CI-Premier Conference, 2008,
pp. 81–96. ISBN: 978-981-08-0412-1.

[60] ŠTĚPÁNEK, P., POPELA, P., PLŠEK, J.: Structural Optimization Of Concrete Structures By Stochastic Programming. In *SimTecT 2008* Conference Proceedings. Melbourne: Simulation Industry Association of Australia, 2008, ISBN 0-9775257-4-0.

[61] ŠTĚTINA, J., KAVIČKA, F., DOBROVSKÁ, J., CAMEK, L., MASARIK, M.: Optimization of a concasting technology via a dynamic solidification model of a slab caster. *Materials Science Forum 5*, 475-479 (2005), 3831–3834. ISSN 0255-5476.

[62] TABAK, D., KUO, B. C.: *Optimal Control by Mathematical Programming.* Prentice-Hall, 1971. 237 p. ISBN 978-0136381068.

[63] VALENTE, C., MITRA, G., SADKI, M., FOURER, R.: Extending algebraic modelling languages for stochastic programming. *Informs Journal on Computing 21* (2009), 107–122.

[64] VARAIYA, P., WETS, R. J-B.: Stochastic Dynamic Optimization Approaches and Computation. In Proc. 13th Int. Symp. *Mathematical programming.* Tokyo: KTK Scientific Publisher, 1989, pp. 309–331.

[65] VOŘECHOVSKÝ, M., NOVÁK, D.: Simulation of random fields for stochastic finite element analysis. In *Safety and Reliability of Engineering Systems and Structures.* Netherlands: Millpress Rotterdam, 2005, pp. 436–443. ISBN: 90-5966-040-4.

[66] WATKINS, D. W., MCKINNEY, D. C., MORTON, D. P.: Groundwater pollution control. In *Applications of stochastic programming. MPS-SIAM series on optimization – volume 5* (eds. Wallace, S. W., Ziemba, W. T.), Chapter 21, SIAM, 2005, pp. 409–424. ISBN 978-089-87-1555-2.

[67] WATSON, J.-P., WOODRUFF, D. L., STRIP, D. R.: Progressive hedging innovations for a class of stochastic resource allocation problems. UC Davis Graduate School of Management Research Paper No. 05-08, 2008.[cited 2009-10-07]. Available at: http://ssrn.com/abstract=1268385

[68] WESTMANN, J. J., HANSON, F. B.: Computational method for nonlinear stochastic optimal control. In Proceedings of *American Control Conference.* San Diego, 1999, pp. 2798–2802. ISBN 0-7803-4990-3.

[69] WETS, R. J-B: The Aggregation Principle in Scenario Analysis and Stochastic Optimization. In *Algorithms and Model Formulations in Mathematical Programming.* New York: Springer-Verlag, 1989, pp. 91–113.

[70] WETS, R. J-B: *Challenges in Stochastic Programming.* IIASA Working Paper WP-94-32, Austria, 1994.

[71] Wikipedia, the free encyclopedia [cited 2009-11-07]. Available at: http://en.wikipedia.org/wiki/

[72] YIP, S. (ed.): *Handbook of Materials Modeling.* 1st ed. Dordrecht: Springer, 2005. 2965 p. ISBN 978-1402032875.

[73] ŽAMPACHOVÁ, E.: Approximation of the continuous stochastic programs via mathematical programming (in Czech). In Proceedings from 16th seminar *Modern mathematical methods in engineering.* Ostrava, 2007, pp. 334–338.
ISBN 978-80-248-1649-4.

[74] ŽAMPACHOVÁ, E., POPELA, P.: The selected PDE constrained stochastic programming problem. In Proceedings of the *Risk, Quality and Reliability Conference*. Ostrava, 2007, pp. 233–237. ISBN 978-80-248-1575-6.

[75] ŽAMPACHOVÁ, E:. Determining solution's quality of the selected stochastic programming problem via Monte Carlo method (in Czech). In Proceedings of conference CQR – *REQUEST '08*. Brno, 2008, pp. 244–249. ISBN 978-80-214-3774-6.

[76] ŽAMPACHOVÁ, E.: Approximate solution of PDE constrained stochastic optimization problems (in Czech). In Proceedings of *FME Junior conference*. Brno, 2009, pp. 16–23. ISBN 978-80-214-3834-7.

[77] ŽAMPACHOVÁ, E., POPELA, P.: Different reformulations of stochastic optimization of the transverse vibrations. Accepted in *Engineering mechanics*. 2009.

Notation

Only the most widely used abbreviations and symbols are presented. The others are explained when they are used.

SP	stochastic programming
WS	wait-and-see
HN	here-and-now
EV	expected value
EO	expected objective
VSS	value of stochastic solution
EVPI	expected value of perfect information
ODE	ordinary differential equation
PDE	partial differential equation
FDM	finite difference method
FEM	finite element method
NLP	nonlinear programming
MINLP	mixed-integer nonlinear programming
PHA	progressive hedging algorithm
s. t.	subject to
$\mathbb{R}^n$	n dimensional Euclidean space
C	feasible set
$\mathbf{x}$	first-stage decision variable
$\mathbf{y}$	second-stage decision variable
$z^\bullet$	optimal objective function value for reformulation $\bullet$
ω	random element
Ω	set of all random elements

ω_s	realization of a random element
$\boldsymbol{\xi}$	random vector
Ξ	support of the random vector $\boldsymbol{\xi}$
$\boldsymbol{\xi}_s$	realization of the random vector
$\mathbb{P}$	probability distribution on $(\Xi, \mathcal{B})$
$\mathbb{E}(\cdot)$	mean value
p_s	scenario probability, i. e. $p_s = \mathbb{P}(\boldsymbol{\xi} = \boldsymbol{\xi}_s)$
x	space coordinate
t	time coordinate
i	space index, $i = 0, \ldots, N$
j	time index, $j = 0, \ldots, M$
s	scenario index, $s = 1, \ldots, R$
d	spatial step, $d = \frac{l}{N}$
τ	time step, $\tau = \frac{T}{M}$
l	length
T	time of observation
E	Young's modulus
J	second moment of the cross section
ρ	material density or penalty parameter
$M(x)$	bending moment
$\sigma_{max}(x)$	maximum stress
$h(x)$	external static load
$h(\xi, x, t)$	random external dynamic load
$v(\xi, x, t)$ or $v(\xi, x)$	displacement or deflection
$e(x)$	first-stage decision variable
$g(\xi, x, t)$	second-stage decision variable
a, b	dimensions of the cross section
α, β, γ	weighting coefficients

Appendix

A.1 Convergence theorem of PHA

Theorem A1. *[Rockafellar, Wets, 1987] Consider the PHA defined in the page 75. Let the sets $C(s)$ and the functions $f(\mathbf{x}, s)$ are convex for every $s \in S$. Let $\{\hat{\mathbf{x}}^k\}_{k=1}^{\infty}$ and $\{\mathbf{w}^k\}_{k=1}^{\infty}$ be the sequences generated by the algorithm from an arbitrary initial choice of $\hat{\mathbf{x}}^0$ and $\mathbf{w}_s^0$. These sequences will be bounded if and only if optimal solutions $\mathbf{x}^*$ and $\mathbf{w}^*$ of (7.21) and its dual problem exist. Then*

$$\hat{\mathbf{x}}^k \to \mathbf{x}^* \text{ and } \mathbf{w}^k \to \mathbf{x}^*.$$

In every iteration $k = 1, 2, \ldots$ one will have that

$$\left\| \left(\hat{\mathbf{x}}^{k+1}, \frac{\mathbf{w}^{k+1}}{\rho} \right) - \left(\mathbf{x}^*, \frac{\mathbf{w}^*}{\rho} \right) \right\|_{\mathbb{E}} \leq \left\| \left(\hat{\mathbf{x}}^{k}, \frac{\mathbf{w}^{k}}{\rho} \right) - \left(\mathbf{x}^*, \frac{\mathbf{w}^*}{\rho} \right) \right\|_{\mathbb{E}}$$

with strict inequality unless $(\hat{\mathbf{x}}^k, \mathbf{w}^k) = (\mathbf{x}^*, \mathbf{w}^*)$.
Thus every iteration of the algorithm makes a definite improvement until solutions are attained (if that occurs in finitely many steps). One will also have in every iteration that

$$\left\| \left(\hat{\mathbf{x}}^{k+1}, \frac{\mathbf{w}^{k+1}}{\rho} \right) - \left(\hat{\mathbf{x}}^{k}, \frac{\mathbf{w}^{k}}{\rho} \right) \right\|_{\mathbb{E}} \leq \left\| \left(\hat{\mathbf{x}}^{k}, \frac{\mathbf{w}^{k}}{\rho} \right) - \left(\hat{\mathbf{x}}^{k-1}, \frac{\mathbf{w}^{k-1}}{\rho} \right) \right\|_{\mathbb{E}},$$

where $\|(\mathbf{u}, \mathbf{v})\|_{\mathbb{E}} = \sqrt{\mathbb{E}(\|\mathbf{u}(s)\|^2) + \mathbb{E}(\|\mathbf{v}(s)\|^2)} = \sqrt{\sum_{s \in S} p_s(\|\mathbf{u}(s)\|^2) + \sum_{s \in S} p_s(\|\mathbf{v}(s)\|^2)}.$

Proof. The proof can be found in [51] or [6] and it is based on Rockafellar's proximal point method. □

A.2 Excerpts from GAMS source codes

```
$title Optimality gap by Monte Carlo method

$offlisting;

option solprint = off;
option limrow = 0,limcol = 0;

Scalars          N             number of grid points                    /50/
                 l             beam length [mm]                         /1000/
                 bb                                                     /20/
                 ro            steel density [tmm-3]                    /7.85E-9/
                 alfa          weighting coefficient for rigidity       /0.3/
                 beta          weighting coefficient for weight         /0.45/
                 gamma         weighting coefficient for deflection     /0.25/
                 sigma         stress limitation [MPa]                  /100/
                 rigidity      normalization constant                   /1.8E+12/
                 weight        normalization constant                   /0.007/
                 deflection    normalization constant                   /0.7/;

Sets             i             spatial index                            /0*50/
                 no_s          number of scenarios (5..100)             /1*20/
                 ss            all scenarios                            /1*2000/
                 s(ss)         selected scenarios
                 sb            batches                                  /1*30/;

Parameter        d             spatial step;
                 d=l/N;

Parameter        range(no_s)   different numbers of scenarios in batches;
                 range(no_s) = 0+5*ord(no_s);

Parameters       obj(no_s,sb)  objective function values
                 gap(no_s,sb)  optimality gap
                 prob(ss)      scenario probability
                 E(ss)         Young's modulus [MPa];

Parameter        h(i)          deterministic static load;
                 h(i)=-4*bb*((ord(i)-1)*d)*((ord(i)-1)*d)/l**2+4*bb*((ord(i)-1)*d)/l;

Parameter        acand         cand.solution from EO reform. with 100 scenarios;
                 acand=22.4;

Parameter        bcand         cand.solution from EO reform. with 100 scenarios;
                 bcand=100;

Variables        zcan          objective function with fixed candidate solutions
                 vcan(ss,i)    deflection [mm]

                 zEO           objective function - EO reformulation
                 v(ss,i)       deflection [mm];

Positive variables
                 acan          dimension of cross section
                 bcan          dimension of cross section

                 a             dimension of cross section in EO model
                 b             dimension of cross section in EO model;
```

```
Equations       objcan                  objective function with fixed cand.solutions
                BCacan(ss)              left boundary condition
                BCbcan(ss,i)            right boundary condition
                ODE_1can(ss,i)          deflection in i=1
                ODE_2N_2can(ss,i)       deflection in i=2..N-2
                ODE_N_1can(ss,i)        deflection in i=N-1
                maxstresscanup(ss,i)    maximal stress in i=1..N-1
                maxstresscanlo(ss,i)    maximal stress in i=1..N-1
                maxstresscanup_0(ss)    maximal stress in i=0
                maxstresscanlo_0(ss)    maximal stress in i=0
                maxstresscanup_N(ss,i)  maximal stress in i=N
                maxstresscanlo_N(ss,i)  maximal stress in i=N

                 objEO                  objective function - EO reformulation
                 BCa(ss)                left boundary condition
                 BCb(ss,i)              right boundary condition
                 ODE_1(ss,i)            deflection in i=1
                 ODE_2N_2(ss,i)         deflection in i=2..N-2
                 ODE_N_1(ss,i)          deflection in i=N-1
                 maxstressup(ss,i)      maximal stress in i=1..N-1
                 maxstresslo(ss,i)      maximal stress in i=1..N-1
                 maxstressup_0(ss)      maximal stress in i=0
                 maxstresslo_0(ss)      maximal stress in i=0
                 maxstressup_N(ss,i)    maximal stress in i=N
                 maxstresslo_N(ss,i)    maximal stress in i=N;


*---------------------- model with candidate solution --------------------------*
objcan.. zcan =e= -alfa*sum(s,prob(s)*E(s))*acan*bcan**3/(12*rigidity)
                 +beta*ro*acan*bcan*l/weight
                 +gamma*sum((s,i),prob(s)*vcan(s,i))/deflection;

BCacan(s).. vcan(s,'0')=e= 0;

BCbcan(s,i)$(ord(i) eq card(i)).. vcan(s,i)=e= 0;

ODE_1can(s,i)$(ord(i) eq 2)..
(E(s)*acan*bcan**3/12)*(vcan(s,i+2)-4*vcan(s,i+1)+7*vcan(s,i)) =e= h(i)*d**4;

ODE_2N_2can(s,i)$((ord(i) ne 1)and(ord(i) ne 2)and(ord(i) ne (card(i)-1))and
(ord(i) ne card(i)))..
(E(s)*acan*bcan**3/12)*(vcan(s,i+2)-4*vcan(s,i+1)+6*vcan(s,i)
-4*vcan(s,i-1)+vcan(s,i-2)) =e= h(i)*d**4;

ODE_N_1can(s,i)$(ord(i) eq (card(i)-1))..
(E(s)*acan*bcan**3/12)*(7*vcan(s,i)-4*vcan(s,i-1)+vcan(s,i-2)) =e= h(i)*d**4;

maxstresscanup(s,i)$((ord(i) ne 1)and(ord(i) ne card(i)))..
E(s)*bcan*(vcan(s,i+1)-2*vcan(s,i)+vcan(s,i-1))/(2*d**2) =l= sigma;

maxstresscanup_0(s).. E(s)*bcan*vcan(s,'1')/d**2 =l= sigma;

maxstresscanup_N(s,i)$(ord(i) eq card(i)).. E(s)*bcan*vcan(s,i-1)/d**2 =l= sigma;

maxstresscanlo(s,i)$((ord(i) ne 1)and(ord(i) ne card(i)))..
-E(s)*bcan*(vcan(s,i+1)-2*vcan(s,i)+vcan(s,i-1))/(2*d**2) =l= sigma;

maxstresscanlo_0(s).. -E(s)*bcan*vcan(s,'1')/d**2 =l= sigma;

maxstresscanlo_N(s,i)$(ord(i) eq card(i)).. -E(s)*bcan*vcan(s,i-1)/d**2 =l= sigma

*---------------------------- EO reformulation ---------------------------------*
objEO.. zEO =e= -alfa*sum(s,prob(s)*E(s))*a*b**3/(12*rigidity)
               +beta*ro*a*b*l/weight+gamma*sum((s,i),prob(s)*v(s,i))/deflection;

BCa(s)..                        v(s,'0')=e= 0;

BCb(s,i)$(ord(i) eq card(i)).. v(s,i)=e= 0;

ODE_1(s,i)$(ord(i) eq 2)..(E(s)*a*b**3/12)*(v(s,i+2)-4*v(s,i+1)+7*v(s,i))=e=h(i)*d**4;
```

```
ODE_2N_2(s,i)$((ord(i) ne 1)and(ord(i) ne 2)and(ord(i) ne (card(i)-1))and
(ord(i) ne card(i)))..
(E(s)*a*b**3/12)*(v(s,i+2)-4*v(s,i+1)+6*v(s,i)-4*v(s,i-1)+v(s,i-2)) =e= h(i)*d**4;

ODE_N_1(s,i)$(ord(i) eq (card(i)-1))..
(E(s)*a*b**3/12)*(7*v(s,i)-4*v(s,i-1)+v(s,i-2)) =e= h(i)*d**4;

maxstressup(s,i)$((ord(i) ne 1)and(ord(i) ne card(i)))..
E(s)*b*(v(s,i+1)-2*v(s,i)+v(s,i-1))/(2*d**2) =l= sigma;

maxstressup_0(s).. E(s)*b*v(s,'1')/d**2 =l= sigma;

maxstressup_N(s,i)$(ord(i) eq card(i)).. E(s)*b*v(s,i-1)/d**2 =l= sigma;
maxstresslo(s,i)$((ord(i) ne 1)and(ord(i) ne card(i)))..
-E(s)*b*(v(s,i+1)-2*v(s,i)+v(s,i-1))/(2*d**2) =l= sigma;

maxstresslo_0(s).. -E(s)*b*v(s,'1')/d**2 =l= sigma;

maxstresslo_N(s,i)$(ord(i) eq card(i)).. -E(s)*b*v(s,i-1)/d**2 =l= sigma;
*-------------------------------------------------------------------------------*

model candidate /objcan,BCacan,BCbcan,ODE_1can,ODE_2N_2can,ODE_N_1can,
                 maxstresscanup,maxstresscanup_0,maxstresscanup_N,
                 maxstresscanlo,maxstresscanlo_0,maxstresscanlo_N/;

model EOrefor   /objEO,BCa,BCb,ODE_1,ODE_2N_2,ODE_N_1,
                 maxstressup,maxstressup_0,maxstressup_N,maxstresslo,
                 maxstresslo_0,maxstresslo_N/;

loop(no_s,
    loop(ss,if(ord(ss) le range(no_s), s(ss) = yes););

prob(s)= 1/card(s);

loop(sb,
E(s)=200E3+uniform(-10E3,50E3);
acan.up=acand;
acan.lo=acand;
bcan.up=bcand;
bcan.lo=bcand;
vcan.l(s,i)=0;
solve candidate using nlp minimizing zcan;

a.lo=10;
a.up=100;
b.lo=10;
b.up=100;
a.l=100;
b.l=100;
v.l(s,i)=0;
solve EOrefor using nlp minimizing zEO;

gap(no_s,sb)=zcan.l-zEO.l;
obj(no_s,sb)=zEO.l
););

file opt_gap /opt_gap.dat/, obj_func /obj_func.dat/;
put opt_gap;
put /;
loop((no_s,sb), put no_s.tl,@6, sb.tl,@14, gap(no_s,sb):14:8 /);

put obj_func;
put /;
loop((no_s,sb), put no_s.tl,@6, sb.tl,@14, obj(no_s,sb):14:8 /);
```