STOCHASTIC PROGRAMMING

Pavel Popela

ppope01@um.edu.mt

Department of Statistics and Operations Research
Faculty of Science
University of Malta
Malta

and

popela@um.fme.vutbr.cz

Division of Stochastic and Optimization Methods
Institute of Mathematics
Faculty of Mechanical Engineering
Brno University of Technology
Czech Republic

LECTURE 1: BASIC CONCEPTS AND IDEAS

1 Introduction

1.1 Motivation

Remark 1 (Modelling of decision making) The average human life is based on permanent decision making. Certain decision situations are so important that they require optimal decisions based on precise thinking. During the last fifty years, mathematical programming has been developed as an important supporting tool. It involves deep theoretical results, fast numerical algorithms, advanced models, user-friendly software, and successful applications. However, its main branches are based on deterministic models while real-world decision situations are full of uncertainties. So, our purpose is to study the basic ideas and principal concepts as well as the considerable findings of the stochastic programming (SP) approach to decision making under uncertainty. At first, we review some concepts.

Remark 2 (Historical remarks) Abstract ideas and one-dimensional examples of SP may be found in early papers on inventory problems. SP *originated* in the fifties, and problems involving uncertainty and decisions over time periods have been studied since 1955 (Dantzig aircraft allocation). The first important *theoretical results* were published in the sixties by Madansky, Wets, Prékopa, etc. Special chapters on the subject are contained in the books on MP (e.g., Hadley). Lately, the first monographs appeared (e.g., Kall). The seventies brought deep theoretical results, (e.g., Rockafellar) and specialized conferences on SP. A *remarkable progress* was made in the eighties with the development of algorithms and multistage SP (e.g., Birge, Ermoliev). New areas of interest were integer SP and SP networks (e.g., Wallace). In the nineties, the questions related to the availability of *modelling and algorithmic tools* and parallel implementation of algorithms have become more and more important.

Remark 3 (Applications) Many real problems involve uncertainties:

- **Finance** – interest rates: portfolio and asset liability management, and insurance policies.
- **Energy management**: capacity expansion problems, L-S hydroelectric system, and Michigan power system scheduling.
- **Water management** – uncertain rain: reservoir design, Balaton lake regulation, uncertain risk of floods, and irrigation network design.
- **Engineering** + nonlinearities: *equipment design*: measurements, *structural design*: reliability estimated, *robot control*: stochastic disturbances, *melt control*: random losses.
- **Various applications:** *in production* – uncertain costs, *in transport*: uncertain demands – Dantzig, *in telecommunications*: line requests, *in forest harvesting*: frequency of fires in Canada, and *in fishery management*: fish occurrence near Norway. For more and details, see Prekopa + Birge books on SP.

Remark 4 (Notation — randomness) We assume that randomness of scalars and vectors is emphasized by ($\boldsymbol{\xi}$) symbols following the letters denoting them. Use this notation correctly i.e. $\mathbf{A}(\boldsymbol{\xi})$ means that matrix $\mathbf{A}$ has random elements. Having doubts – write about your doubts in the exam paper – I can accept (without problems) your changes of the notation when they are well explained.

Remark 5 (Notation — abbreviations) We further use the following abbreviations that are common in stochastic programming: SProg for stochastic program or programming, WS for Wait-and-See, HN for Here-and-Now, SB for scenario-based, TS - for two-stage, MS for multistage, EV for Expected Value, IS for individual scenario, and MM for defensive (Min-Max objective) approaches.

Further used abbreviations are: EO - expected objective, VO - variance objective, PO - probabilistic objective, QO - quantile objective, UO - utility objective, PF - permanently feasible, AS - almost surely feasible, PC - probabilistic constraints, SP - separable probabilistic constraints (like in the knapsack example; do not mix with SProg abbreviation), JP - joint probabilistic constraints, RF - recourse function (see two-stage SProg with recourse). All related concepts were verbally explained during lectures.

Remark 6 (Notation — comparison of models) EEV - is the Expectation of the objective function value for the EV solution, VSS - identifies the Value of Stochastic Solution, EVPI - denotes the Expected Value of Perfect Information. Remember that if you want you may get Birge-Louveaux book copies that may help you with some details on EVPI and VSS.
You may work with or without these abbreviations or you can introduce (clearly — please) your own.

1.2 Basic concepts of deterministic programs

Definition 1 (Mathematical program) We define *mathematical program* (MP) as:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}) \mid \mathbf{x} \in C\}, \tag{1}$$

where:

$C \subseteq \mathbb{R}^n$ is called a *feasible set*, $n \in \mathbb{N}$,

$f : C \longrightarrow \mathbb{R}$ is called an *objective function*, and

$\mathbf{x} \in C$ describes a *decision (vector) variable.*

Remark 7 (Importance of MPs) Many decision problems are modelled as MPs. MPs (1) often involve important constant (deterministic) parameters. We may emphasize this fact writing parameters explicitly in MP.

Definition 2 (Parametric MP) We define *parametric MP* (PMP)[1] as:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}, \mathbf{a}) \mid \mathbf{x} \in C(\mathbf{a})\}, \tag{2}$$

where:

[1]parametric form of mathematical program

$\mathbf{a} \in \mathbb{R}^K$ is a *constant parameter*, $K \in \mathbb{N}$.

Exercise 1 (Review!) *Definitions of minima $\mathbf{x}_{min}$ (global and local), optimal objective function value $z_{min} = f(\mathbf{x}_{min})$, Weierstrass' theorem, convexity and its consequences (local $\Rightarrow$ global). Design examples of MPs.*

Definition 3 (Linear program) We define a *linear program* (LP) as:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{\mathbf{c}^\top \mathbf{x} \mid \mathbf{A}\mathbf{x} \diamondsuit \mathbf{b}, \mathbf{l} \le \mathbf{x} \le \mathbf{u}\}, \tag{3}$$

where:

$\mathbf{c} = (c_j)_{j \in \mathcal{J}} \in \mathbb{R}^n$ are coefficients in the objective function, $\mathcal{J}$ is a set of indices of variables, usually $\mathcal{J} = \{1, \ldots, n\}$.

$\mathbf{b} = (b_i)_{i \in \mathcal{I}} \in \mathbb{R}^m$, $\mathbf{A} = (a_{ij})_{i \in \mathcal{I}, j \in \mathcal{J}} \in \mathbb{R}^{m \times n}$ are coefficients for *constraints*, $\mathcal{I}$ is a set of indices of constraints, usually $\mathcal{I} = \{1, \ldots, m\}$.

$\mathbf{l} = (l_j)_{j \in \mathcal{J}} \in \mathbb{R}^n$, $\mathbf{u} = (u_j)_{j \in \mathcal{J}} \in \mathbb{R}^n$ are coefficients for *lower and upper bounds* (usually $\forall j \in \mathcal{I} : l_i = 0$ and $u_j = \infty$, often can be included to constraints).

$\diamondsuit$ symbol is further systematically used to express various relations ($\diamondsuit = (\diamond_i)_{i \in \mathcal{I}} \in \{\le, \ge, =\}^m$) between LHS and RHS in constraints altogether.

Exercise 2 (Review!) *Polyhedral set, extreme points (EP) and directions, representation theorem, optimality conditions (if $\exists \mathbf{x}_{min}$ then in EP), simplex method, duality in LP. Design examples of LPs.*

Remark 8 (LP as parametric MP) We can write LP (3) as MP (1) and even as PMP (2):

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}, \mathbf{c}) \mid \mathbf{x} \in C(\mathbf{A}, \mathbf{b}, \mathbf{l}, \mathbf{u})\}, \tag{4}$$

where

$f(\mathbf{x}, \mathbf{c}) = \mathbf{c}^\top \mathbf{x}$

$C(\mathbf{A}, \mathbf{b}, \mathbf{l}, \mathbf{u}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \diamondsuit \mathbf{b}, \mathbf{l} \le \mathbf{x} \le \mathbf{u}\}$

Exercise 3 (Review!) *Basics about parametric linear programming from previous lectures on dynamic programming (DP).*

Definition 4 (Nonlinear program) We define a *nonlinear program* (NLP) as:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}) \mid \mathbf{g}(\mathbf{x}) \diamondsuit \mathbf{0}, \mathbf{x} \in X\}, \tag{5}$$

where:

$\mathbf{g} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is nonlinear.

$X \subseteq \mathbb{R}^n$.

Exercise 4 (Review!) *Calculus-like NLP examples, convexity (epigraph), optimality conditions ($\nabla f(\mathbf{x}) = \mathbf{0}$, positive definite $\mathbf{H}(\mathbf{x})$, KKT), ideas of algorithms. Design own examples of NLPs.*

Remark 9 (NLP as PMP) We can write NLP (5) as PMP (2):

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x},\mathbf{a}) \mid \mathbf{x} \in C(\mathbf{a})\}, \tag{6}$$

where

$C(\mathbf{a}) = \{\mathbf{x} \mid \mathbf{g}(\mathbf{x},\mathbf{a})\Diamond\mathbf{0}, \mathbf{x} \in X\}$

Exercise 5 (Review!) *Basics about parametric nonlinear programming from DP lectures.*

Definition 5 (Integer program) LP (3) with additional integrality conditions is called *integer linear program* (ILP):

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{\mathbf{c}^\top\mathbf{x} \mid \mathbf{A}\mathbf{x}\Diamond\mathbf{b}, \mathbf{x} \in D\}, \tag{7}$$

where: $D \subset \mathbb{R}^n$ is a discrete set. $D \subset \mathcal{Z}^n$ ($\mathcal{Z}$ is a set of all integers) for ILP, $D \subset \mathcal{Z}^l \times \mathbb{R}^{n-l}$ for mixed ILP (MILP), and $D \subseteq \{0,1\}^n$ for 0-1 programs.

Remark 10 (More on IP) In the case of NLP when we replace set X with set D, see (5), we get integer nonlinear program (INLP). ILP can be written as PMP similarly as LP and NLP before. Some combinatorial decision problems (with a finite feasible set) are modelled by IP (integer programs/integer programming). In general, they are studied in discrete (combinatorial) optimization.

Exercise 6 (Review!) *Basic ideas on bounding in Branch-and-Bound algorithm, logical conditions modelling by 0-1 variables (see H.P. Williams), knapsack is again one of the KEY applications for this topic.*

Remark 11 (Network flows) LP (3) with a special matrix $\mathbf{A}$ (vertex-arc incidence matrix derived from the directed graph related to the problem modelled by network flows) is called a *network flow model* (NF).

Exercise 7 (Review!) *How to model NF by LP, integrality of NF solution.*

Remark 12 (Decomposable programs) MP can be written as a two-step *decomposed program* (2DP):

$$? = \min_{\mathbf{x}_1}\{\min_{\mathbf{x}_2}\{f(\mathbf{x}_1,\mathbf{x}_2) \mid \ x_2 \in C_2(\mathbf{x}_1)\} \mid \mathbf{x}_1 \in C_1\}, \tag{8}$$

where

$C_1 \subseteq \mathbb{R}^{n_1}$, $\forall \mathbf{x}_1 \in C_1 : C_2(\mathbf{x}_1) \subseteq \mathbb{R}^{n_2}$, and $\mathbf{x} = (\mathbf{x}_1^\top, \mathbf{x}_2^\top)^\top$,

$C = \{\mathbf{x} \mid \mathbf{x} = (\mathbf{x}_1^\top, \mathbf{x}_2^\top)^\top,\ \mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2(\mathbf{x}_1)\}$, so $n = n_1 + n_2$.

<u>In the special case:</u>

$$? = \min_{\mathbf{x}_1}\{f_1(\mathbf{x}_1, \min_{\mathbf{x}_2}\{f_2(\mathbf{x}_2) \mid \mathbf{x}_2 \in C_2(\mathbf{x}_1)\}) \mid \mathbf{x}_1 \in C_1\}, \tag{9}$$

where $f_1 : C_1 \times \mathbb{R} \longrightarrow \mathbb{R}$, $f_2 : \mathbb{R}^{n_2} \longrightarrow \mathbb{R}$, $f(\mathbf{x}) = f_1(\mathbf{x}_1, f(\mathbf{x}_2))$ and decomposability assumption is satisfied (f_1 is strictly increasing in the second argument).
<u>Under further assumptions</u> about *states* (special NLP, LP, IP, NF case), we write:

$$\begin{aligned} ? = \min_{\mathbf{x}_1,\mathbf{y}_1}\{f_1(\mathbf{x}_1, \min_{\mathbf{x}_2}\{f_2(\mathbf{x}_2) \mid \mathbf{x}_2 \in C_{2y}(\mathbf{y}_1)\}) \mid \\ \mathbf{x}_1 \in C_1, \mathbf{y}_1 = \mathbf{h}_1(\mathbf{x}_1)\}, \end{aligned} \tag{10}$$

where

$\mathbf{y}_1$ is a state variable, $\mathbf{h}_1 : C_1 \longrightarrow \mathbb{R}^{m_1}$, $m_1 \ll n_1$ is assumed.
$\forall \mathbf{y}_1 \in \mathbb{R}^{m_1} \mid (\exists \mathbf{x}_1 \in C_1 : \mathbf{y}_1 = \mathbf{h}_1(\mathbf{x}_1)) : C_{2y}(\mathbf{y}_1) \subseteq \mathbb{R}^{n_2}$,
$C = \{\mathbf{x} \mid \mathbf{x} = (\mathbf{x}_1^\top, \mathbf{x}_2^\top)^\top, \mathbf{x}_1 \in C_1, \mathbf{y}_1 = \mathbf{h}_1(\mathbf{x}_1), \mathbf{x}_2 \in C_2(\mathbf{y}_1)\}$.

Such decomposition is widely applied to MPs (especially IPs, LPs). The inner program in (9) is already PMP (2).

Exercise 8 (Review!) *Decomposition assumption, states. Design examples (knapsack, LP, NLP)!*

Remark 13 (2DPs as PMPs) 2DPs (8), (9), and (10) can be written as PMPs (2):

$$\begin{aligned} ? = \min_{\mathbf{x}}\{f(\mathbf{x}, \mathbf{a}) \mid \mathbf{x} \in C(\mathbf{a})\} = \\ \min_{\mathbf{x}_1,\mathbf{x}_2}\{f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{a}_1, \mathbf{a}_2) \mid (\mathbf{x}_1^\top, \mathbf{x}_2^\top)^\top \in C(\mathbf{a}_1, \mathbf{a}_2)\} \end{aligned} \tag{11}$$

$$\begin{aligned} ? = \min_{\mathbf{x}_1}\{\min_{\mathbf{x}_2}\{f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{a}_1, \mathbf{a}_2) \mid \mathbf{x}_2 \in C_2(\mathbf{x}_1, \mathbf{a}_2)\} \mid \\ \mathbf{x}_1 \in C_1(\mathbf{a}_1)\}, \end{aligned} \tag{12}$$

$$\begin{aligned} ? = \min_{\mathbf{x}_1}\{f_1(\mathbf{x}_1, \min_{\mathbf{x}_2}\{f_2(\mathbf{x}_2, \mathbf{a}_2) \mid \mathbf{x}_2 \in C_2(\mathbf{x}_1, \mathbf{a}_2)\}, \mathbf{a}_1) \mid \\ \mathbf{x}_1 \in C_1(\mathbf{a}_1)\}, \end{aligned} \tag{13}$$

$$\begin{aligned} ? = \min_{\mathbf{x}_1,\mathbf{y}_1}\{f_1(\mathbf{x}_1, \min_{\mathbf{x}_2}\{f_2(\mathbf{x}_2, \mathbf{a}_2) \mid \mathbf{x}_2 \in C_{2y}(\mathbf{y}_1, \mathbf{a}_2)\}, \mathbf{a}_1) \mid \\ \mathbf{x}_1 \in C_1(\mathbf{a}_1),\ \mathbf{y}_1 = \mathbf{h}_1(\mathbf{x}_1, \mathbf{a}_1)\}, \end{aligned} \tag{14}$$

where

$\mathbf{a}_1 \in \mathbb{R}^{K_1}$, $\mathbf{a}_2 \in \mathbb{R}^{K_2}$, are constant parameters. For PMP (2) relation we define $\mathbf{a} = (\mathbf{a}_1^\top, \mathbf{a}_2^\top)^\top$, $K = K_1 + K_2$.

Exercise 9 (Review!) *Design examples based on LP.*

Remark 14 (Split description of 2DP) 2DP as in (9) can be written as two connected programs:

$$? = \min_{\mathbf{x}_1}\{f_1(\mathbf{x}_1, z_{2,\min}(\mathbf{x}_1)) \mid \mathbf{x}_1 \in C_1(\mathbf{a}_1)\}, \qquad \text{where}$$

$$z_{2,\min}(\mathbf{x}_1) = \min_{\mathbf{x}_2}\{f_2(\mathbf{x}_2, \mathbf{a}_2) \mid \mathbf{x}_2 \in C_2(\mathbf{x}_1, \mathbf{a}_2)\}. \tag{15}$$

For (10) we get:

$$\begin{aligned} ? = \min_{\mathbf{x}_1,\mathbf{y}_1}\{&f_1(\mathbf{x}_1, z_{2y,\min}(\mathbf{y}_1), \mathbf{a}_1) \mid \\ &\mathbf{x}_1 \in C_1(\mathbf{a}_1),\ \mathbf{y}_1 = \mathbf{h}_1(\mathbf{x}_1, \mathbf{a}_1)\}, \qquad \text{where} \end{aligned}$$

$$z_{2y,\min}(\mathbf{y}_1) = \min_{\mathbf{x}_2}\{f_2(\mathbf{x}_2, \mathbf{a}_2) \mid \mathbf{x}_2 \in C_2(\mathbf{y}_1, \mathbf{a}_2)\}. \tag{16}$$

Remark 15 (Nested programs) If decomposition (see (9)) is applied recursively then we talk about a *dynamic program* or *nested program* or *multi-step program* or *multiperiod program* (MDP):

$$\begin{aligned} ? &= \min_{\mathbf{x}_1}\{f_1(\mathbf{x}_1, z_{2,\min}(\mathbf{x}_1)) \mid \mathbf{x}_1 \in C_1(\mathbf{a}_1)\} \\ z_{2,\min}(\mathbf{x}_1) &= \min_{\mathbf{x}_2}\{f_2(\mathbf{x}_2, z_{3,\min}(\mathbf{x}_2), \mathbf{a}_2) \mid \mathbf{x}_2 \in C_2(\mathbf{x}_1, \mathbf{a}_2)\} \\ &\ldots \\ z_{t,\min}(\mathbf{x}_{t-1}) &= \min_{\mathbf{x}_t}\{f_t(\mathbf{x}_t, z_{t+1,\min}(\mathbf{x}_t), \mathbf{a}_t) \mid \mathbf{x}_t \in C_t(\mathbf{x}_{t-1}, \mathbf{a}_t)\} \\ &\ldots \\ z_{T,\min}(\mathbf{x}_{T-1}) &= \min_{\mathbf{x}_T}\{f_T(\mathbf{x}_T, \mathbf{a}_T) \mid \mathbf{x}_T \in C_T(\mathbf{x}_{T-1}, \mathbf{a}_T)\}, \end{aligned} \tag{17}$$

where symbols are used in similar way as for previous two-step cases.

Exercise 10 (Review!) *Similar description as (17) can be applied to the case with states (10) — write it yourself. Use the knapsack problem as an example.*

Example 1 (Cases used for further explanations)

- *Calculus-like* problems without/with bounds, e.g.:

$$\min(x-a)^2$$

- *Unconstrained minimization* problems:

$$\min_{\mathbf{x}\in\mathbb{R}^n}\{(\mathbf{D}\mathbf{x}+\mathbf{c})^\top(\mathbf{D}\mathbf{x}+\mathbf{c})\}$$

- Non-smooth (piece-wise defined) convex objective function.
- *Continuous knapsack* problems:

$$\max_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} \mid \mathbf{a}^\top\mathbf{x}\le b, \mathbf{x}\in[0,1]^n\}$$

- *Linear programming* problems:

$$\max_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} \mid \mathbf{A}\mathbf{x}=\mathbf{b}, \mathbf{x}\ge\mathbf{0}\}$$

- Melt control (charge optimization), production, inventory, and transportation problems by LP.
- *Discrete knapsack* problems:

$$\max_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} \mid \mathbf{a}^\top\mathbf{x}\le b, \mathbf{x}\in\{0,1\}^n\}$$

- Assignment and other combinatorial problems.
- *Quadratic programming* problems:

$$\min_{\mathbf{x}}\{\frac{1}{2}\mathbf{x}^\top\mathbf{D}\mathbf{x}+\mathbf{c}^\top\mathbf{x} \mid \mathbf{A}\mathbf{x}=\mathbf{b}, \mathbf{x}\ge\mathbf{0}\}.$$

- Investment problems, multicriteria-based decisions, and utility function, *further NLPs*.

1.3 The key ideas of stochastic programming

Remark 16 (Underlying programs) From the parametric mathematical program, we obtain an underlying program by replacing some constant parameters by random variables. We ask what is the meaning of the underlying program. It will be quite clear after observation $\boldsymbol{\xi}^s$ substitutes for $\boldsymbol{\xi}$, but what happens to the program when the realization of the randomness is not observed? Although the description is unclear, it is often used, because it may help from the modelling point of view. Therefore, it is understood as a syntactically correct description, for which semantics is given later.

Remark 17 (Deterministic reformulations/equivalents) We work mainly with *deterministic reformulations (equivalents)* that correctly interpret random elements. All these programs that involve random parameters in syntactically correct ways, are called *stochastic programs.*

Remark 18 (Explicit form) One of the important questions is whether the deterministic equivalent may be expressed in *the explicit form* of a traditional mathematical program that is 'easily solvable'.

Remark 19 (Wait-and-see approach) The main question that should be answered is when the decision will be made — before the random parameters $\boldsymbol{\xi}$ are observed or after the observations $\boldsymbol{\xi}^s$ are known. According to Madansky, when the decision $\mathbf{x}$ is made after observing the randomness $\boldsymbol{\xi}$, this case is called the *wait-and-see* (WS) approach. Wallace characterized WS approach as: "Analysis under uncertainty of a decision under certainty." This approach is valuable when we know the realization of $\boldsymbol{\xi}$ before making our decision, and it assumes the perfect information about the future. In this case, we may modify our decision by observation, and hence, the decision $\mathbf{x}$ is a function $\mathbf{x}(\boldsymbol{\xi})$ of the random vector $\boldsymbol{\xi}$. Also, the outcome $f(\mathbf{x}(\boldsymbol{\xi}); \boldsymbol{\xi})$ is a random variable. This approach has its importance specifically for long-term planning. The analysis of wait-and-see decisions is also applied in mathematical statistics (cf. regression analysis).

Remark 20 (Here-and-now approach) Decision makers must often make decisions before the observations of $\boldsymbol{\xi}$ are known. In this case, they are using the so-called *here-and-now* (HN) approach by Madansky. The decision $\mathbf{x}$ must be the same for any future realization of $\boldsymbol{\xi}$. Stochastic programming deals primarily with here-and-now decisions, because the typical decision situation is described by the lack of observations.

Remark 21 (Risk and uncertainty) In stochastic programming we focus on probability distributions, instead of constant parameters. How much information is available regarding a probability distribution? Stochastic programming deals particularly with programs in which the probability distributions of the random parameters $\boldsymbol{\xi}$ are fully known. This case was called *programming under risk* by Luce and Raiffa. This basic assumption is not always satisfactory in practice. In some cases, sufficient statistical data are available and the distribution can be determined with high reliability by statistical methods. But frequently, not enough information is available, as $\boldsymbol{\xi}$ is a multi-dimensional random vector, and estimating procedures need a large amount of data. In these situations, when no observations exist, the experts may create their own probability distributions, which are usually different from the true distributions, and post-optimality procedures are then required. This lack of knowledge leads to models, which will not render suggestions as efficient for the decision process as those with programming under risk. Therefore, defensive modelling approaches such as a minimax approach or a 'fat solution' approach are used as frequent tools in such situations.

Remark 22 (Measurability and existence) The first step in using SProg requires that all used symbols are well defined. It is assumed that $f(\mathbf{x}; \boldsymbol{\xi})$ is measurable, and that the expectation $E_{\xi} f(\mathbf{x}; \boldsymbol{\xi})$ exists for all $\mathbf{x}$.

Remark 23 (Continuity) This property of the objective function is usually studied in stochastic programming, because the continuity is necessary for the existence theorems of the optimal solution and is also an important assumption for the application of direct search methods.

Remark 24 (Convexity) For the convex objective and the convex feasible set, every local minimum is also a global one, and the set of optimal decisions is convex in this case. The assumption of convexity also plays a key role in convergence of many numerical algorithms.

Remark 25 (Differentiability) This property allows a theoretical use of the well-known concepts from nonlinear programming. The validity of conditions under which we may compute gradient $\nabla_x E_\xi f(\mathbf{x};\boldsymbol{\xi})$ together with the subdifferentiability property for stochastic programming were discussed by Wets. Then we use the formula

$$\nabla_x E_\xi f(\mathbf{x};\boldsymbol{\xi}) = \textstyle\int_\Xi \nabla_x f(\mathbf{x};\boldsymbol{\xi}) P(\mathrm{d}\boldsymbol{\xi}).$$

Remark 26 (Optimality conditions) Using the subdifferential properties of expectation functionals, optimality conditions may be obtained from known conditions for nonlinear programming.

Remark 27 (Algorithms) Useful algorithms for stochastic programs may be classified from the viewpoint of whether the deterministic equivalent is completely specified in an explicit way. In this case, modified traditional techniques of mathematical programming can be used. But, there are various classes of deterministic equivalents that must be solved by special algorithms, because they cannot be expressed as explicit mathematical programs.

Remark 28 (Linearity) Stochastic programs having linear objective and constraints in the underlying program are called *stochastic linear programs* (SLP). It is known that the linearity assumption in deterministic mathematical programming guarantees many important properties. There are also general algorithms useful for all deterministic linear programs. Linearity also implies important properties of stochastic programs, e. g. from the linearity of $f(\mathbf{x};\boldsymbol{\xi})$, with respect to $\mathbf{x}$, the linearity of $E_\xi f(\mathbf{x};\boldsymbol{\xi})$ follows, but the construction of the deterministic equivalents cannot be realized in this direct way, because the deterministic equivalents of stochastic linear programs are not linear mathematical programs in general. Therefore, the influence of linearity must be studied for different deterministic equivalents separately, and specifically, there are no general algorithms in stochastic linear programming.

Remark 29 (Computational problems) The formulation may often be analysed as a normal deterministic nonlinear program. But there is a problem when the objective and the constraints are in the form of expectations having integral representations. It seems that after computing those integrals and identifying the program's properties (convexity, differentiability, linearity), a standard mathematical programming algorithm may be chosen to solve the problem. When the expectations are calculated, there are still several serious computational problems related to solving stochastic programs, because expectations often lead to explicit deterministic programs with exponentially increasing size. It is still an open question how to compute values of multidimensional integrals repeatedly. We need to calculate

$$E_\xi f(\mathbf{x};\boldsymbol{\xi}) = \textstyle\int_\Xi f(\mathbf{x};\boldsymbol{\xi}) P(\mathrm{d}\boldsymbol{\xi}).$$

It is hard to compute the expectation above, and also its gradient, repeatedly. These problems are usually solved by various stochastic and deterministic approximation schemes using pseudorandom sequences for multidimensional integration.

Manageable cases occur when the distribution of $\boldsymbol{\xi}$ is discrete and concentrated in not too many atoms, because it leads to a summation instead of an integration or when function f is separable in $\boldsymbol{\xi}$, and so

$$E_\xi f(\mathbf{x};\boldsymbol{\xi}) = \sum_i \int_{\Xi_i} f_i(\mathbf{x};\xi_i) P(\mathrm{d}\xi_i),$$

and we may use one-dimensional numerical integration routines.

Remark 30 (Approximations) The basic assumption in stochastic programs is that all probability distributions of the random parameters are well defined. This assumption is not often satisfied in practice. Therefore, the distributions are chosen by the modeller, or they are verified by historical data. This increases the importance of the analysis of the influence of distribution changes on the results. Such changes can be studied from different viewpoints, so the analysis of consequences of using one distribution instead of another one in stochastic programs leads to a wide theoretical research in several directions.

Remark 31 (Approximation schemes) They are based on the replacement of the original expectations by simpler approximations. This can be done by replacing $f(\mathbf{x}; \boldsymbol{\xi})$ by a suitable separable function with respect to the components ξ_i of $\boldsymbol{\xi}$. The other possibility is based on a discretization of $\boldsymbol{\xi}$, because the calculation of expectations is reduced to a multiple summation.

Remark 32 (Comparisons) We have defined different deterministic equivalents based on different approaches to the random parameters contained in stochastic programs. It seems reasonable to compare the optimal values for the different equivalents. The following important quantities are later defined for this purpose EEV, EVPI, and VSS.

1.4 Underlying (stochastic) programs

Definition 6 (Underlying program – UP) We define *underlying (stochastic) program* as:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}, \boldsymbol{\xi}) \mid \mathbf{x} \in C(\boldsymbol{\xi})\}, \tag{18}$$

where:
$\boldsymbol{\xi} : \Omega \longrightarrow \mathrm{I\!R}^K$ is a *random vector*, for
$(\Omega, \mathcal{F}, P)$ given *probability space.*

Remark 33 (UP and PMP) UP (18) is derived from (2) by replacing $\mathbf{a}$ with $\boldsymbol{\xi}$.

Remark 34 (Notation)

- As $\boldsymbol{\xi}$ is an $\mathcal{F}$-measurable mapping, it induces a probability distribution on $\mathrm{I\!R}^K$. We denote a probability space as $(\mathrm{I\!R}^K, \mathcal{B}, \mathcal{P})$ or $(\Xi, \mathcal{B}, \mathcal{P})$, where Ξ is a support of $\mathcal{P}$ (the smallest set by $\subseteq$ such that $\mathcal{P}(\Xi) = 1$). $\mathcal{B}$ is a σ-field of Borel's sets on $\mathrm{I\!R}^K$ (or projections of those sets on Ξ).
- Derived probabilities are computed by the rule $\forall B \in \mathcal{B} : \mathcal{P}(B) := P(\{\omega \mid \boldsymbol{\xi}(\omega) \in B\})$. Because of $\mathcal{F}$-measurability $\{\omega \mid \boldsymbol{\xi}(\omega) \in B\} \in \mathcal{F}$, and hence, $\mathcal{P}$ domain is specified fully and consistently.
- To emphasize the use of random vector $\boldsymbol{\xi}$, we often write $P(\boldsymbol{\xi} \in B)$ instead of $\mathcal{P}(B)$. $P(\boldsymbol{\xi} \in B)$ is a short version of $P(\{\omega \mid \boldsymbol{\xi}(\omega) \in B\})$.
- $\forall \omega^s \in \Omega : \boldsymbol{\xi}(\omega^s) \in \mathrm{I\!R}^K$ is a *realization* (observation) of $\boldsymbol{\xi}$. In short, we write $\boldsymbol{\xi}^s$.

Exercise 11 (Review!) *Probability space, random variable, random vector, probability distribution, Borel's sets, σ-field, $\mathcal{F}$-measurable, realization.*

Example 2 (Underlying programs)

a) Calculus:
$$\min_{x \in \mathbb{R}} (x - \xi)^2, \quad \xi \sim Be(p) \;\text{ Bernoulli distribution.}$$

b) LSQ:
$$\min_{\mathbf{x} \in \mathbb{R}^n} \sum_{k=1}^{K} (\mathbf{A}\mathbf{x} - \boldsymbol{\xi})^\top (\mathbf{A}\mathbf{x} - \boldsymbol{\xi}), \quad \boldsymbol{\xi} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

c) LP:
$$\min_{\mathbf{x}} \{\mathbf{c}^\top \mathbf{x} \mid \mathbf{A}\mathbf{x} \Diamond \mathbf{b}\}, \quad \boldsymbol{\xi} = (\mathbf{c}, \mathbf{A}, \mathbf{b}) \;\text{ random.}$$

d) LP written alternatively in more compact way:
$$\min_{\mathbf{x}} \{\mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x} \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x} \Diamond \mathbf{b}(\boldsymbol{\xi})\}, \quad \boldsymbol{\xi} \;\text{ random.}$$

Remark 35 (Notation) We denote presence of random parameters in the model in different ways:

- We write ξ or $\boldsymbol{\xi}$ directly in MP as in Example 2 a) and b). It is simple as it is clear what is considered to be random, as we denote random elements by Greek letters.
- We write MP in the common style, and afterwards, we list components of $\boldsymbol{\xi}$ as in Example 2 c).
- We write $(\boldsymbol{\xi})$ after letters denoting random parameters as in Example 2 d). It may look too cumbersome at the first view but for general cases it is immediately seen what is random.

  In addition, there are two cases for which this notation may be used: $\mathbf{c}(\boldsymbol{\xi})$ reminds that $\boldsymbol{\xi} = (\mathbf{c}, \ldots)$ as in c) or $\mathbf{c}(\boldsymbol{\xi})$ may mean that it depends on some random vector $\boldsymbol{\xi}$ (usually less dimensional).

Remark 36 (UP characteristics) We see that the underlying (stochastic) program (18) is mainly characterized by:

- Optimization goal: min or max.
- The form of an objective f (the possibility to utilize Calculus in an easy way), whether and how depends on $\boldsymbol{\xi}$.
- The presence of constraints, so whether $C(\boldsymbol{\xi}) = \mathbb{R}^n$.
- Type of constraints and dependence of involved terms on $\boldsymbol{\xi}$.
- Dimension k of random vector $\boldsymbol{\xi}$.
- Dimension n of decision variable $\mathbf{x}$.

- Type of probability distribution involved (discrete (with a finite or infinite support) or continuous).
- Number S of realizations (scenarios) of $\boldsymbol{\xi}$ considered for the finite discrete case.

Example 3 (The first engine choice) We have introduced underlying (stochastic) program (18). It is a mathematical program involving random parameters. We want to solve it. The attempt to do it correctly will immediately lead to several important questions. They will remain of the same importance even for the simplest possible case. Let us choose the simplest example as the "first engine" to test our further ideas:

$$? \in \underset{x \in \mathbb{R}}{\operatorname{argmin}} (x - \xi)^2, \quad \xi \sim Be(0.6). \tag{19}$$

It is chosen as it is an *unconstrained minimization* problem with *one random* ($k = 1$) and *one decision* variable ($n = 1$) involved in the objective. We consider the smallest possible number of scenarios $S = 2$. In addition, it is easy to utilize Calculus (even common sense) for the objective.

Question 1 (What is the meaning of the UP?) The first important question appears:

> What is the meaning of
>
> $$"? \in \underset{\mathbf{x}}{\operatorname{argmin}} \{ f(\mathbf{x}, \boldsymbol{\xi}) \mid \mathbf{x} \in C(\boldsymbol{\xi}) \}"?$$

LECTURE 2: BASIC WAIT-AND-SEE

Example 4 We apply question to our "simple engine example":

$$\text{What is the meaning of } ? \in \underset{x \in \mathbb{R}}{\operatorname{argmin}} (x - \xi)^2, \quad \xi \sim Be(p) \ ?$$

Why is it difficult to decide? One may say that by common sense (or Calculus) $x_{\min} = \xi$. However, till now, an optimal solution was described by constant values (LP, NLP, IP, DP)! How to set up and model this new one? Toss the coin? Can we react on the observation (realization) of $\boldsymbol{\xi}$? Do we need to decide before the realization $\boldsymbol{\xi}^s$ is known? It leads to the conclusion:

> Underlying program (18) is written in the syntactically correct way (correct mathematical symbols, specified parameters and their dimensions), however, it is meaningless[2] (without additional information).

It leads to the next question that searches for additional information:

Question 2 (What is coming first: $\boldsymbol{\xi}^s$ or x?) Why is it important to ask and answer this question? Because by the answer we can model optimal decision in different ways. Precisely in two obvious ways:

- $\boldsymbol{\xi}^s$ precedes $\mathbf{x}$ i.e. it is known when the decision is taken.
- $\boldsymbol{\xi}^s$ follows $\mathbf{x}$ i.e. it is not known when the decision is taken.

[2] We may also say that the program description is incomplete.

Knowing realization $\boldsymbol{\xi}^s$, we can react on it and adopt our decision to it!

Question 3 (Who will give the answer?) For the answer whether $\boldsymbol{\xi}^s$ follows or precedes $\mathbf{x}$ are definitely responsible the modeller (e.g., mathematician) and the user (e.g., engineer). Certainly, they have to follow real conditions if the model is applied in practice. For educational examples as Example 3 we are "lucky" that we may play with both cases.

2 Objective function reformulations - finite case

2.1 Wait-and-see programs

Example 5 ($\boldsymbol{\xi}^s$ known) We continue with Example 3. Remember that it is only *unconstrained univariate (n=1) optimization problem with the quadratic objective function and random variable ξ (k=1) with known finite discrete probability distribution ($\xi \sim Be(0.6)$) involved in it.*

$$? \in \operatorname*{argmin}_{x \in \mathbb{R}} (x-\xi)^2, \quad \xi \sim Be(0.6). \tag{20}$$

At first, we solve the program as PMP i.e. forgetting about random nature of ξ. We can use calculus (or common sense reasoning) and get the optimal solution in the closed form:

$$\min_x (x-\xi)^2 \Rightarrow \tfrac{\partial}{\partial x}(x-\xi)^2 = 0 \Rightarrow x_{\min} = \xi. \tag{21}$$

In addition $z_{\min} = (x_{\min} - \xi)^2 = (\xi - \xi)^2 = 0$. How to interpret the fact that $\mathbf{x}_{\min}$ depends on ξ, and hence changes with the change of ξ? We utilize the parametric programming approach and write:

$$\xi = 1 \Rightarrow x_{\min} = 1, z_{\min} = 0$$
$$\xi = 0 \Rightarrow x_{\min} = 0, z_{\min} = 0$$

We see that $x_{\min}$ and $z_{\min}$ are functions of ξ. So, we will write:

$$x_{\min}(\xi) = \xi, \quad z_{\min}(\xi) = 0.$$

Now we remember that we deal with *probability distribution of* ξ. So, we may conclude that optimal solution $x_{\min}(\xi)$ is a random variable and $z_{\min}(\xi)$ is also a random variable (although degenerate in this case). It is quite natural to ask question about probability distributions of $x_{\min}(\xi), z_{\min}(\xi)$ now. As we know that $P(\xi = 1) = 0.6$ and $P(\xi = 0) = 0.4$ from $\xi \sim Be(0.6)$, we may write:

$$P(x_{\min}(\xi) = 1) = 0.6, \;\; P(x_{\min}(\xi) = 0) = 0.4,$$
$$x_{\min}(\xi) \sim \mathrm{Be}(0.6), \quad P(z_{\min}(\xi) = 0) = 1.$$

As we have described random variables $x_{\min}(\xi), \; z_{\min}(\xi)$ completely, we can deal with them as with other random variables in probability, e.g., compute characteristics based on moments.

$$E_\xi[x_{\min}(\xi)] = 0.6(1) + 0.4(0) = 0.6,$$

$$\mathrm{var}_\xi[x_{\min}(\xi)] = 0.6(1^2) + 0.4(0^2) - 0.6^2 = 0.24,$$

$$E_\xi[z_{\min}(\xi)] = 0, \;\; \mathrm{var}_\xi[z_{\min}(\xi)] = 0.$$

Remark 37 (UP characteristics — continuation) By Example 5, we see that the underlying (stochastic) program (18) is further characterized by:

- Whether the solution of UP is in the *closed form* (i.e. an explicit formula in the closed form could be derived for it.) In our Example 5) the answer is YES. Otherwise, a suitable numerical algorithm could be needed.
- Whether the *probability distribution of $\boldsymbol{\xi}$ is specified completely*. In Example 5 $\xi \sim Be(0.6)$, so the answer is YES.
- Whether *we may wait* for realization $\boldsymbol{\xi}^s$ of $\boldsymbol{\xi}$. In Example 5, we adopted $\mathbf{x}_{\min}$ by values $\xi^0 = 0$ and $\xi^1 = 1$ and we wrote $\mathbf{x}_{\min}(\xi)$ and derived a probability distribution of $\mathbf{x}_{\min}(\xi)$, so definitely YES!!

Remark 38 (Wait-and-See (WS) answer) So, the first question that should be answered is when the decision will be made — before the random parameters $\boldsymbol{\xi}$ are observed or after the observations $\boldsymbol{\xi}^s$ are known. According to Madansky, when the decision $\mathbf{x}$ is made after observing the randomness $\boldsymbol{\xi}$, this case is called the *wait-and-see* (WS) approach.

Remark 39 (More on WS) Wallace characterized WS approach as:

> "Analysis under uncertainty of a decision under certainty."

The WS approach is valuable when we know the realization of $\boldsymbol{\xi}$ before making our decision, and it assumes the perfect information about the future. In this case, we may modify our decision by observation, and hence, the decision $\mathbf{x}$ will be understood as a function $\mathbf{x}(\boldsymbol{\xi})$ of the random vector $\boldsymbol{\xi}$. Also, the outcome $f(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})$ will be understood as a random variable (see Definition 7 and Remark 40). This approach has its importance specifically for long-term planning.

Exercise 12 *The wait-and-see approach and analysis of WS decisions is also applied in mathematical statistics. Where? Find examples! Hint: Think about regression analysis.*

Definition 7 (WS deterministic reformulation) Let the UP (see Definition 6 for details) be given:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}, \boldsymbol{\xi}) \mid \mathbf{x} \in C(\boldsymbol{\xi})\}.$$

We define its *wait-and-see (WS) deterministic reformulation* (WS program):

$$? \in \underset{\mathbf{x}(\xi)}{\operatorname{argmin}}\{f(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}) \mid \mathbf{x}(\boldsymbol{\xi}) \in C(\boldsymbol{\xi})\}, \tag{22}$$

where $\mathbf{x}(\boldsymbol{\xi})$ denotes a (measurable) mapping $\mathbf{x} : \mathbb{R}^K \longrightarrow \mathbb{R}^n$.

Remark 40 (WS notation and interpretation) In Definition 7 we emphasize WS approach (the fact that the decision $\mathbf{x}$ depends on $\boldsymbol{\xi}$) by writing $\mathbf{x}(\boldsymbol{\xi})$. So $\mathbf{x}(\boldsymbol{\xi})$ is a function of $\boldsymbol{\xi}$ now, i.e. $\mathbf{x} : \mathbb{R}^K \longrightarrow \mathbb{R}^n$. Even more, there is an induced probability distribution related to $\mathbf{x}(\boldsymbol{\xi})$. Therefore, WS deterministic reformulation (22) could be interpreted as a special parametric mathematical program (cf. (2)[3]) with "probability-based weights". To emphasize the fact that not only one $\mathbf{x}_{\min}$ is computed, we may also write:

$$\forall \boldsymbol{\xi} \in \Xi : \ ? \in \underset{\mathbf{x}(\xi)}{\operatorname{argmin}}\{f(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}) \mid \mathbf{x}(\boldsymbol{\xi}) \in C(\boldsymbol{\xi})\} \tag{23}$$

We may emphasize the relation of the solution to WS reformulation (22) by using superscripts WS i.e. $\mathbf{x}_{\min}^{\mathrm{WS}}(\boldsymbol{\xi})$ and $z_{\min}^{\mathrm{WS}}(\boldsymbol{\xi})$ (instead of $\mathbf{x}_{\min}(\boldsymbol{\xi})$ and $z_{\min}(\boldsymbol{\xi})$) in the future. We denote the set of all optimal solutions by $X_{\min}(\boldsymbol{\xi})$ (or $X_{\min}^{\mathrm{WS}}(\boldsymbol{\xi})$) i.e.

$$X_{\min}(\boldsymbol{\xi}) = \underset{\mathbf{x}(\xi)}{\operatorname{argmin}}\{f(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}) \mid \mathbf{x}(\boldsymbol{\xi}) \in C(\boldsymbol{\xi})\}. \tag{24}$$

For typesetting reasons, we do not use boldface $\boldsymbol{\xi}$ in subscripts, superscript, and below symbols as $\sum$, min, argmin, etc.

Remark 41 (General comments on WS) We consider general deterministic WS reformulation within this paragraph, however, we begin with the following assumptions valid till now: $C = \mathbb{R}^n$, $\mathbf{x} = x \in \mathbb{R}$, $\boldsymbol{\xi} = \xi : \Omega \longrightarrow \mathbb{R}$, $\mathbf{x}_{\min}(\boldsymbol{\xi})$ computed by an explicit closed formula, so $\exists \boldsymbol{\psi} : \mathbb{R}^K \longrightarrow \mathbb{R}^n : \mathbf{x}_{\min}(\boldsymbol{\xi}) = \boldsymbol{\psi}(\boldsymbol{\xi})$, and $\boldsymbol{\xi}$ probability distribution is finite discrete and known (and we wait-and-see for $\boldsymbol{\xi}^s$).

- In the case of another finite discrete distribution (either typical Bi, H or general $p(\boldsymbol{\xi}^s) = P(\boldsymbol{\xi} = \boldsymbol{\xi}^s)$), we want to utilize the algorithm:

  $\boxed{0.}$ Identify Ξ (notice that $|\Xi| < \aleph_0$) fully (listing all elements $\boldsymbol{\xi}^s \in \Xi$ or giving a generating formula), identify $\forall \boldsymbol{\xi}^s \in \Xi : p(\boldsymbol{\xi}^s) = P(\boldsymbol{\xi} = \boldsymbol{\xi}^s)$ (or $F(\boldsymbol{\xi}^s)$ cumulative probability distribution function values).

  $\boxed{1.}$ $\forall \boldsymbol{\xi}^s \in \Xi$ solve programs

  $$? \in \underset{\mathbf{x}(\xi^s)}{\operatorname{argmin}}\{f(\mathbf{x}(\boldsymbol{\xi}^s), \boldsymbol{\xi}^s) \mid \mathbf{x}(\boldsymbol{\xi}^s) \in C(\boldsymbol{\xi}^s)\}$$

  to get all their optimal solutions $\mathbf{x}_{\min}(\boldsymbol{\xi}^s)$, $\boldsymbol{\xi}^s \in \Xi$ and probabilities $P(\mathbf{x}_{\min}(\boldsymbol{\xi}) = \mathbf{x}_{\min}(\boldsymbol{\xi}^s)) = p(\boldsymbol{\xi}^s)$, and similarly, for all optimal objective function values $z_{\min}(\boldsymbol{\xi}^s) = f(x_{\min}(\boldsymbol{\xi}^s), \boldsymbol{\xi}^s)$, we obtain $P(z_{\min}(\boldsymbol{\xi}) = z_{\min}(\boldsymbol{\xi}^s)) = p(\boldsymbol{\xi}^s)$.

  Any problems? Yes:

- What if values of $\mathbf{x}_{\min}(\boldsymbol{\xi}^s)$ or $z_{\min}(\boldsymbol{\xi}^s)$ are the same for different values of $\boldsymbol{\xi}^s$? See our Example 5 for $z_{\min}(\boldsymbol{\xi})$. No problem:

  $$P(\mathbf{x}_{\min}(\boldsymbol{\xi}) = \mathbf{x}) = \sum_{\mathbf{x}_{\min}(\xi^s) = \mathbf{x}} p(\boldsymbol{\xi}^s),$$

[3] In (2), we did not emphasize dependence of $\mathbf{x}$ on $\mathbf{a}$ because we use (2) in more general syntactical sense.

similarly for $z_{\min}(\boldsymbol{\xi})$, we obtain

$$P(z_{\min}(\boldsymbol{\xi}) = z) = \sum_{z_{\min}(\xi^s)=z} p(\boldsymbol{\xi}^s).$$

- What about a non-unique solution i.e.

$$\mathbf{x}_{\min}(\boldsymbol{\xi}) \in X_{\min}(\boldsymbol{\xi}),\ \exists \boldsymbol{\xi}^s \in \Xi : |X_{\min}(\boldsymbol{\xi}^s)| > 1.$$

  This looks as a serious problem. "Wait (patiently) and you will see!" Just to touch the problem complexity: "Is it about generalization of the concept of probability distribution on Ω that is composed of subsets of $\mathbb{R}^n$?" Again WS!

- More in this direction: What about guaranteed existence of the probability distributions of $\mathbf{x}(\boldsymbol{\xi})$ and $f(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})$ and even more $\mathbf{x}_{\min}(\boldsymbol{\xi})$ and $z_{\min}(\boldsymbol{\xi})$? So, any problem with measurable $\boldsymbol{\psi}(\boldsymbol{\xi})$?

- What about unboundedness of $f(\mathbf{x}, \boldsymbol{\xi})$ (from below)? Could it occur? Certainly (Think about $\min \xi x^2$.).

- In Example 5, we computed characteristics as
  $E_\xi[x_{\min}(\xi)]$, $\mathrm{var}_\xi[x_{\min}(\xi)]$, $E_\xi[z_{\min}(\xi)]$, and $\mathrm{var}_\xi[z_{\min}(\xi)]$. Any problems with their existence? Definitely.

- If the previous characteristics exist, how to compute them in general?

- If the previous questions are positively answered, what about larger support Ξ of a finite discrete probability distribution? We will need more time for $\boxed{1.}$ loop in the algorithm.

- What about a large dimension of $\boldsymbol{\xi}$ but still finite discrete probability distribution? We will need more loops in $\boxed{1.}$.

- Any problems with a larger dimension of $\mathbf{x}$? It depends on our ability to solve

$$\forall \boldsymbol{\xi}^s \in \Xi :\ ? \in \underset{\mathbf{x}(\xi^s)}{\mathrm{argmin}}\{f(\mathbf{x}(\boldsymbol{\xi}^s), \boldsymbol{\xi}^s) \mid \mathbf{x}(\boldsymbol{\xi}^s) \in C(\boldsymbol{\xi}^s)\}.$$

- What about the even "larger" Ξ, so about continuous probability distribution of $\boldsymbol{\xi}$ case? Any problems? How to deal with this case even when $\boldsymbol{\psi}(\boldsymbol{\xi})$ is known in the closed form? For some cases, you may utilize formulas for computing transformations of random vectors.

- However, what if the solution can be obtained only by a numerical (optimization) algorithm? It means that the solution closed form is lost. For the discrete case you still may lose the previous algorithm (after collecting positive answers to all previously stated questions).

- But what if the solution can be obtained only by a numerical (optimization) algorithm for the continuous probability distribution case? Some simulation-based (sampling-based) techniques could be utilized.

- What about presence of constraints? We have already mentioned them above. So, what about the situation where $C \subseteq \mathbb{R}^n, C \neq \mathbb{R}^n$ or even worse $C(\boldsymbol{\xi}) \subseteq \mathbb{R}^n, C(\boldsymbol{\xi}) \neq \mathbb{R}^n$ (i.e. $\exists \boldsymbol{\xi}^s \in \Xi : \ C(\boldsymbol{\xi}^s) \neq \mathbb{R}^n$ or $C(\boldsymbol{\xi}) \neq \mathbb{R}^n$ $a.s.$ One new fact could be mentioned already now: The infeasibility cases may appear with constraints.

- And last but not least: What if we cannot WS?

Remark 42 (Looking backwards) Before we continue, we emphasize the most important ideas we have discussed till now:

- We reviewed deterministic optimization models (mathematical programs) of various types: linear and nonlinear, with continuous and discrete variables, and also decomposed in different ways. We tried to use unified notation.

- All the time, we have formally emphasized the importance of the explicit description of parameters involved (PMP).

- *It is important that the reader is familiar with basic descriptions of all types of programs listed and their principal features.*

- Then we introduced the underlying program as the parametric description where instead of constant parameters, we take random parameters. Notation was significantly extended.

- *For a reader, it is important to understand the used notation combining optimization and probability concepts and apply it for descriptive examples, e.g., "build the UP".*

- The initial discussion about the sequence of $\mathbf{x}$ and $\boldsymbol{\xi}^s$ resulted in the WS approach. The idea was to let it naturally grow from a similar "parametric" problem, adding "weights".

- Because the simple example was repeatedly used, there was a large reminder what is more complicated (and hence, it will be further discussed) than it looks on the first view.

- *Still the key to the success at this stage is the ability to deal with similar simple examples!*

LECTURE 3: FROM IS AND EV TO EO – BASICS

2.2 Here-and-now programs

Remark 43 (Here-and-now (HN) answer) Many decision makers must often make decisions before the observations of $\boldsymbol{\xi}$ are known. In this case, they are using the so-called *here-and-now* (HN) approach by Madansky. The decision $\mathbf{x}$ must be the same[4] for any future realization of $\boldsymbol{\xi}$. Stochastic programming deals primarily with here-and-now decisions, because the typical decision situation is described by the lack of observations.

Remark 44 (Stochastic programs) We further utilize the concept *stochastic program* in the broad sense i.e. for all programs that in some way involve random parameters. It means that we call stochastic programs both underlying programs and deterministic reformulations.

Remark 45 (Section outline) Before we go to details, we give a short outline of the section. There are three main lines that are developed through the section:

- There is a main story based on the simple example already introduced for the WS case. The idea is to go step by step from one possible HN deterministic reformulation to another one in a motivated way. This line is composed of questions and computational examples and we learn about abbreviations as IS, EV, EO, VO, and MM.

- The second line is motivated by the idea to specify deterministic reformulations precisely. There are definitions immediately following entering examples and remarks on notation that may look too compact for the first reading.

- The third and last line tries to answer questions like: "When we developed this deterministic reformulation, how good it is and how it could be compared with other reformulations?" It mixes definitions, remarks about notation and computational examples as well. The reader will learn about another set of abbreviations as EEV, VSS, and EVPI.

Example 6 (Individual scenario approach) We further continue with Example 3, see UP:

$$\min(x-\xi)^2, \quad \xi \sim Be(0.6).$$

We consider the decision situation when we cannot adopt our decision x by observation ξ^s. We have to take here-and-now decision, the same decision x for all future realizations ξ^s. In comparison with WS approach, there will be several different ways how to do it. The first general idea for the HN case is to utilize knowledge we already have i.e. to apply one of known WS solutions for certain realization ξ^s i.e. to choose $\mathbf{x}_{\min}(\boldsymbol{\xi}^s)$. We will further denote $\mathbf{x}_{\min}(\boldsymbol{\xi}^s)$ shortly as $\mathbf{x}^s_{\min}$. Those realizations $\boldsymbol{\xi}^s$ are often called *scenarios* in application problems. Therefore, we may try to apply the WS solution for a selected *individual scenario* that we consider by some reasons important. This way of thinking is quite popular for practitioners solving complex problems especially in economics. So, we solve the problem for one selected scenario. In our case, we will do it for both scenarios and we will compare solutions.

$$\begin{array}{llll} \text{For } \xi^0 = 0: & \min x^2 & \Rightarrow & x^0_{\min} = 0,\ z^0_{\min} = 0. \\ \text{For } \xi^1 = 1: & \min(x-1)^2 & \Rightarrow & x^1_{\min} = 1,\ z^1_{\min} = 0. \end{array}$$

[4] Could it be possible that the WS solution will be constant with respect to $\boldsymbol{\xi}$, i.e. $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$?

Definition 8 (IS deterministic reformulation) Let the UP (see Definition 6 for details) be given:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x},\boldsymbol{\xi}) \mid \mathbf{x} \in C(\boldsymbol{\xi})\},$$

We define its here-and-now *individual scenario (IS) deterministic reformulation* (IS program):

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x},\boldsymbol{\xi}^s) \mid \mathbf{x} \in C(\boldsymbol{\xi}^s)\}, \tag{25}$$

where $\boldsymbol{\xi}^s \in \Xi$ is a specified individual scenario. We denote the minimal objective function value as $z^{\mathrm{IS}}_{\mathrm{min}}$ and minimum as $\mathbf{x}^{\mathrm{IS}}_{\mathrm{min}}$.

Example 7 (Comparison of IS solutions) To learn how good is the obtained solution, it is reasonable to evaluate $f(\mathbf{x},\boldsymbol{\xi})$ for $\mathbf{x}^s_{\mathrm{min}}$ in general i.e. to compute $f(\mathbf{x}^s_{\mathrm{min}},\boldsymbol{\xi})$. As a result, we will get a random variable, in our case for $x^0_{\mathrm{min}} = 0$, from $(x^0_{\mathrm{min}} - \xi)^2$ we get $(-\xi)^2$ that may achieve either value 0 with probability 0.4 (as $P(\xi = 0) = 0.4$) or value 1 with probability 0.6 (as $P(\xi = 1) = 0.6$).

Similarly, for $x^1_{\mathrm{min}} = 1$, from $(x^1_{\mathrm{min}} - \xi)^2$ we get $(1 - \xi)^2$ that may achieve either value 0 with probability 0.6 (as $P(\xi = 1) = 0.6$) or value 1 with probability 0.4 (as $P(\xi = 0) = 0.4$).

Therefore, we have two random variables $\zeta^{\mathrm{IS}}_0 = f(x^0_{\mathrm{min}},\xi)$ and $\zeta^{\mathrm{IS}}_1 = f(x^1_{\mathrm{min}},\xi)$, which probability distributions describe the future costs in the case of implementation of the related optimal solution:

$$\begin{array}{l}
\text{For } x^0_{\mathrm{min}}: \; P(\zeta^{\mathrm{IS}}_0 = 0) = 0.4, \; P(\zeta^{\mathrm{IS}}_0 = 1) = 0.6. \\
\text{For } x^1_{\mathrm{min}}: \; P(\zeta^{\mathrm{IS}}_1 = 0) = 0.6, \; P(\zeta^{\mathrm{IS}}_1 = 1) = 0.4.
\end{array}$$

It is immediately seen that the choice of x^1_{min} looks more suitable, as 0 cost is more frequent. It also looks that "in average" we will spend less:

$$\begin{array}{l}
\text{For } x^0_{\mathrm{min}}: \; E[\zeta^{\mathrm{IS}}_0] = 0(0.4) + 1(0.6) = 0.6. \\
\text{For } x^1_{\mathrm{min}}: \; E[\zeta^{\mathrm{IS}}_1] = 0(0.6) + 1(0.4) = 0.4.
\end{array}$$

Another evaluation may be motivated by our risk awareness, so we may decide to compute a variance for both cases:

$$\begin{array}{l}
\mathrm{var}[\zeta^{\mathrm{IS}}_0] = E[(\zeta^{\mathrm{IS}}_0)^2] - (E[\zeta^{\mathrm{IS}}_0])^2 = \\
0^2(0.4) + 1^2(0.6) - (0.6)^2 = 0.24, \\
\mathrm{var}[\zeta^{\mathrm{IS}}_1] = E[(\zeta^{\mathrm{IS}}_1)^2] - (E[\zeta^{\mathrm{IS}}_1])^2 = \\
0^2(0.6) + 1^2(0.4) - (0.4)^2 = 0.24.
\end{array}$$

So, both solutions are at the same level of risk (measured by var).

Question 4 (How to improve the IS solution?) The previous example motivates the question:

How to improve the IS solution?

So, as one reasonable possibility looks the idea to replace $\boldsymbol{\xi}$ with some compromise real value between ξ^0 and ξ^1. It could be a convex combination or weighted average, and hence, precisely $E[\xi]$, shortly $E\xi$.

Example 8 (Expected value used) For Example 3, we compute $E\xi = 0.4(0) + 0.6(1) = 0.6$ and although this "scenario" could not appear as a realization of ξ, we utilize it and solve the program:

$$\min(x - E\xi) = \min(x - 0.6)^2 \quad \Rightarrow \quad x_{\min} = 0.6,\ z_{\min} = 0.$$

Definition 9 (EV deterministic reformulation) Let the UP (see Definition 6 for details) be given:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}, \boldsymbol{\xi}) \mid \mathbf{x} \in C(\boldsymbol{\xi})\},$$

We define its here-and-now *expected value (EV) deterministic reformulation* (EV program):

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}, E\boldsymbol{\xi}) \mid \mathbf{x} \in C(E\boldsymbol{\xi})\}, \tag{26}$$

where $E\boldsymbol{\xi}$ is an expected value of $\boldsymbol{\xi}$. We denote the minimal objective function value as $z^{\mathrm{EV}}_{\min}$ and minimum as $\mathbf{x}^{\mathrm{EV}}_{\min}$.

Question 5 (About quality of the EV solution) The natural question appears:

How good is the solution $\mathbf{x}^{\mathrm{EV}}_{\min}$ for the underlying objective function?

Example 9 (Solution of the EV program) We answer Question 5 for our example i.e. "How good is the solution $\mathbf{x}^{\mathrm{EV}}_{\min}$ for the underlying objective function?" So, we compute $f(\mathbf{x}^{\mathrm{EV}}_{\min}, \boldsymbol{\xi})$ and in our case $\zeta^{\mathrm{EV}} = (0.6 - \xi)^2$:

$$\text{For } x^{\mathrm{EV}}_{\min}:\ P(\zeta^{\mathrm{EV}} = 0.36) = 0.4,\ P(\zeta^{\mathrm{EV}} = 0.16) = 0.6.$$

As before, the reasonable criterion is to compute $E[\zeta^{\mathrm{EV}}]$ because it can be interpreted as an "average cost".

$$E[\zeta^{\mathrm{EV}}] = 0.16(0.6) + 0.36(0.4) = 0.096 + 0.144 = 0.24.$$

The next step could be to compute variance $\mathrm{var}[\zeta^{\mathrm{EV}}]$. By using obvious formula, we get:

$$\begin{gathered}\mathrm{var}[\zeta^{\mathrm{EV}}] = E[(\zeta^{\mathrm{EV}})^2] - (E[\zeta^{\mathrm{EV}}])^2 = \\ 0.16^2(0.6) + 0.36^2(0.4) - (0.24)^2 = \\ 0.01536 + 0.05184 - 0.0576 = 0.0096.\end{gathered}$$

Definition 10 (EEV) For the EV deterministic reformulation, we define EEV as follows:

$$\mathrm{EEV} = E_{\xi}[f(\mathbf{x}^{\mathrm{EV}}_{\min}, \boldsymbol{\xi})] = E[\zeta^{\mathrm{EV}}]. \tag{27}$$

We use the name EEV as $E[\zeta^{\mathrm{EV}}]$ is in fact an Expected objective function value for the optimal solution of the Expected Value deterministic reformulation[5].

[5]The EEV abbreviation is traditionally used for this concept in stochastic programming literature.

Remark 46 (Using EEV) The EEV characteristic can be used to measure whether $z_{\min}^{\mathrm{EV}}$ looks realistic by computing the difference between the optimistic forecasted objective function value $z_{\min}^{\mathrm{EV}}$ and true average cost computed by EEV:

$$\mathrm{EEV} - z_{\min}^{\mathrm{EV}}, \quad \text{for the maximum case:} \quad z_{\max}^{\mathrm{EV}} - \mathrm{EEV}.$$

Example 10 (Evaluation of EVsolution) In Example 3, we get EEV $= 0.24$ and $z_{\min}^{\mathrm{EV}} = 0$, $\mathrm{EEV} - z_{\min}^{\mathrm{EV}} = 0.24$.

Example 11 (Comparison of IS and EV solutions) It is clear that instead of the discussion whether $z_{\min}^{\mathrm{EV}}$ is too optimistic in comparison with $E[\zeta^{\mathrm{EV}}]$, it is more important to compare different optimal solutions we have obtained.

We see that the value of $E[\zeta^{\mathrm{EV}}]$ is lower, and hence, by expected value better than $E[\zeta_0^{\mathrm{IS}}] = 0.6$ and $E[\zeta_1^{\mathrm{IS}}] = 0.4$. Again, as for $E[\zeta^{\mathrm{EV}}]$, variance for the EV case is preferable because $\mathrm{var}[\zeta^{\mathrm{EV}}] = 0.096 < \mathrm{var}[\zeta_0^{\mathrm{IS}}] = \mathrm{var}[\zeta_1^{\mathrm{IS}}] = 0.24$.

So, we have found the way, how the optimal solutions of the different deterministic reformulations can be compared. We derived some conclusions about relations between solutions of IS and EV programs related to Example 3.

It must be also emphasized that it would be too optimistic to make any conclusions about suitability of programs (deterministic reformulations) from conclusions made only about the optimal solutions of some instances of programs.

We have to strongly warn a reader now that all these conclusions and also conclusions related to further examples cannot be generalized without presentation of needed theory. Therefore, the basic examples are usually introduced to motivate and explain further theoretical development.

Remark 47 (Other characteristics) In dependence on the application needs, further characteristics of $\zeta^{\odot}$ random variables[6] could be utilized, e.g., modes, medians, quantiles, and sums of the expected value and multiple of standard deviation.

Remark 48 (The expectation for an objective) Using the previous paragraphs, we derive the idea: "Take the characteristic used for comparisons of $\zeta^{\odot}$ as an objective function." By this idea, we choose $E[f(\mathbf{x}, \boldsymbol{\xi})]$.

Definition 11 (EO deterministic reformulation) Let the UP (see Definition 6 for details) be given:

$$? \in \underset{\mathbf{x}}{\mathrm{argmin}}\{f(\mathbf{x}, \boldsymbol{\xi}) \mid \mathbf{x} \in C(\boldsymbol{\xi})\},$$

We define its here-and-now *expected objective (EO) deterministic reformulation* (EO program):

$$? \in \underset{\mathbf{x}}{\mathrm{argmin}}\{E[f(\mathbf{x}, \boldsymbol{\xi})] \mid \mathbf{x} \in \mathbb{R}^n\}. \tag{28}$$

We denote the minimal objective function value as $z_{\min}^{\mathrm{EO}}$ and minimum as $\mathbf{x}_{\min}^{\mathrm{EO}}$. As before, for comparisons, we introduce ζ^{EO} i.e. $\zeta^{\mathrm{EO}} = f(\mathbf{x}_{\min}^{\mathrm{EO}}, \boldsymbol{\xi})$. Notice please that EO program (28) is unconstrained one.

[6] $\zeta^{\odot}$ is used to denote the UP objective function value for any deterministic reformulation, even for those that will appear in the future.

Example 12 (EO applied) We specify the EO program for Example 3:

$$\min\{E[f(\mathbf{x},\boldsymbol{\xi})] \mid \mathbf{x} \in \mathbb{R}^n\} = \min_{x\in\mathbb{R}}\{0.4x^2 + 0.6(x-1)^2\}.$$

Then we get:

$$(0.4x^2 + 0.6(x-1)^2)' = (x^2 - 1.2x + 0.6)' = 0 \;\Rightarrow$$

$$2x - 1.2 = 0 \;\Rightarrow\; x_{\min}^{\mathrm{EO}} = 0.6,\; z_{\min}^{\mathrm{EO}} = 0.24.$$

$$P(\zeta^{\mathrm{EO}} = 0.16) = 0.6, \quad P(\zeta^{\mathrm{EO}} = 0.36) = 0.4$$

By Definition 11, we know that $z_{\min}^{\mathrm{EO}} = E[\zeta^{\mathrm{EO}}]$. For Example 3, $x_{\min}^{\mathrm{EO}} = x_{\min}^{\mathrm{EV}} = 0.6$ then $z_{\min}^{\mathrm{EO}} = E[\zeta^{\mathrm{EO}}] = \mathrm{EEV} = E[\zeta^{\mathrm{EV}}] = 0.24$. Similarly, $\mathrm{var}[\zeta^{\mathrm{EO}}] = \mathrm{var}[\zeta^{\mathrm{EV}}]$ and even more probability distributions of ζ^{EO} and ζ^{EV} are identical. Therefore, for the example studied, the EV and EO approaches are equivalent.

LECTURE 4: EVPI AND VSS – COMPARISONS

Question 6 (About the EV and EO solutions) By the previous example, we ask:

> Are EV and EO programs equivalent[7] in general?

The immediate answer is no, as EO is an unconstrained program. Therefore, we can ask again:

> Are EV and EO programs equivalent for unconstrained programs?

Remark 49 (EV and EO solutions compared) For the same objective function $(x-\xi)^2$ as in Example 3 and any probability distribution of ξ, we may compute for the objective functions of EO and EV programs respectively:

$$E[f(\mathbf{x},\boldsymbol{\xi})] = E[(x-\xi)^2] =$$

$$E[x^2 - 2x\xi + \xi^2] = x^2 - 2E\xi x + E[\xi^2] \tag{29}$$

$$f(\mathbf{x}, E\boldsymbol{\xi}) = (x - E\xi)^2 = x^2 - 2E\xi x + (E\xi)^2. \tag{30}$$

Utilizing Calculus, we compute minima for both cases:

$$\tfrac{\partial}{\partial x}E[(x-\xi)^2] = 2x - 2E\xi = 0,$$

$$\tfrac{\partial}{\partial x}(x-E\xi)^2 = 2x - 2E\xi = 0.$$

So, $x_{\min}^{\mathrm{EO}} = x_{\min}^{\mathrm{EV}}$ and although mostly $z_{\min}^{EO} \neq z_{\min}^{EV}$, the values for comparison $E[\zeta^{\mathrm{EO}}]$ and $E[\zeta^{EV}] = \mathrm{EEV}$ are obviously equal by their definitions for generalized Example 3.

[7] We take the concept of equivalence of programs only intuitively by the previous paragraphs i.e. there is no difference between ζ^{EV} and ζ^{EO} and their probability distributions.

Remark 50 (General comparison of solutions) In general, the question is about the difference between $E[f(\mathbf{x},\boldsymbol{\xi})]$ and $f(\mathbf{x},E\boldsymbol{\xi})$. In our example, we subtract (30) from (29) and we obtain:

$$E[f(\mathbf{x},\boldsymbol{\xi})] - f(\mathbf{x},E\boldsymbol{\xi}) = E[\xi^2] - (E\xi)^2 = \mathrm{var}[\xi] \geq 0. \tag{31}$$

So, in our case, the difference is constant and derivatives are equal. In addition, we may conclude that with the larger variance of ξ, $z^{\mathrm{EV}}_{\mathrm{min}}$ becomes more optimistic (less realistic) than $z^{\mathrm{EO}}_{\mathrm{min}}$.

Remark 51 (EV and EO objective function values) After the discussion about the optimal solutions, we may learn more about the relation of values of objective functions, and hence, about the difference $E[f(\mathbf{x},\boldsymbol{\xi})] - f(\mathbf{x},E\boldsymbol{\xi})$. We see from (31) that $E[f(\mathbf{x},\boldsymbol{\xi})] > f(\mathbf{x},E\boldsymbol{\xi})$. The conclusion may be obtained by using known Jensen's inequality (1906) saying that for $f(\mathbf{x},\boldsymbol{\xi})$ convex at $\boldsymbol{\xi}$, $E[f(\mathbf{x},\boldsymbol{\xi})] \geq f(\mathbf{x},E\boldsymbol{\xi})$.

Remark 52 (The EO objective value and EEV) At the end, we have to say something about the relation between EEV and the optimal EO objective function value. We know that $z^{\mathrm{EO}}_{\mathrm{min}} = E[f(\mathbf{x}_{\mathrm{min}},\boldsymbol{\xi})] = \mathrm{globmin}_{\mathbf{x}}\{E[f(\mathbf{x},\boldsymbol{\xi})] \mid \mathbf{x} \in \mathbb{R}^n\}$ in general. As it is a global minimum, then $\forall \mathbf{x} \in \mathbb{R}^n : E[f(\mathbf{x}^{\mathrm{EO}}_{\mathrm{min}},\boldsymbol{\xi})] \leq E[f(\mathbf{x},\boldsymbol{\xi})]$ in general. Then, specifically:

$$E[\zeta^{\mathrm{EO}}] = E[f(\mathbf{x}^{\mathrm{EO}}_{\mathrm{min}},\boldsymbol{\xi})] \leq E[f(\mathbf{x}^{\mathrm{EV}}_{\mathrm{min}},\boldsymbol{\xi})] = E[\zeta^{\mathrm{EV}}] = EEV.$$

We have to note that those and previous inequalities require the existence of $E\boldsymbol{\xi}$ and $E[f(\mathbf{x},\boldsymbol{\xi})]$. For the finite discrete cases, we study now, it is satisfied. For more general cases (e.g. continuous $\boldsymbol{\xi}$), it remains the question for the future when the needed expectations exist and how to compute them.

Definition 12 (VSS) We define the Value of Stochastic Solution[8] (VSS) as:

$$\mathrm{VSS} = \mathrm{EEV} - z^{\mathrm{EO}}_{\mathrm{min}}. \tag{32}$$

The concept is introduced to compare the here-and-now EO optimal solution and EV optimal solution by expectations of objective function values. For maximization programs, we write $\mathrm{VSS} = z^{\mathrm{EO}}_{\mathrm{max}} - \mathrm{EEV}$.

Remark 53 (About VSS) The VSS represents the Value of the Stochastic Solution. It is an important characteristic that measures how much can be saved when the true HN approach is used instead of the EV approach. A small value of the VSS means that the approximation of the stochastic program by the EV program is a good one.

Example 13 (VSS computed) For Example 3, we get from Examples 9 and 10 that $\mathrm{EEV} = 0.24$ and as from Example 12 $z^{\mathrm{EO}}_{\mathrm{min}} = 0.24$, we compute $\mathrm{VSS} = \mathrm{EEV} - z^{\mathrm{EO}}_{\mathrm{min}} = 0.24 - 0.24 = 0$. Therefore, for this specific application it is enough to solve the EV program (from the viewpoint considering the EO objective function).

[8] The VSS abbreviation is traditionally used in stochastic programming literature for the similar case.

Exercise 13 *Design an example for which $x_{min}^{EO} \neq x_{min}^{EV}$. Develop your own simple example (analogue to Example 3) and formulate IS, EV, and EO programs. Find their optimal solutions $\mathbf{x}_{min}^{\odot}$ and optimal objective function values $z_{min}^{\odot}$. Derive related $\zeta^{\odot}$ random variables and compute their expectations and variances (including EEV). Compare the solutions of different deterministic reformulations by that means.*

Remark 54 (Comparison of WS and HN) We have introduced the way how to compare optimal solutions of two here-and-now deterministic reformulations EV and EO by means of the VSS. In the similar way, we try to find how to compare optimal solutions of WS and EO (HN) programs. By previous discussions, we consider the EO program as a suitable representative of the class of HN deterministic reformulations.

Remark 55 (Notation) We have used $\zeta^{\odot}$ only for HN programs. However, there is no restriction that does not allow us to use it also for the WS programs. We define a random variable ζ^{WS}:

$$\zeta^{\mathrm{WS}} = z_{\mathrm{min}}^{\mathrm{WS}}(\boldsymbol{\xi}) = f(\mathbf{x}_{\mathrm{min}}^{\mathrm{WS}}(\boldsymbol{\xi}), \boldsymbol{\xi}). \tag{33}$$

Although $\mathbf{x}_{\mathrm{min}}(\boldsymbol{\xi})$ value changes randomly, it still can be applied to computation of the values of true objective function $f(\mathbf{x}, \boldsymbol{\xi})$ as before.

Definition 13 (EVPI) We define the Expected Value of Perfect Information[9] (EVPI) as:

$$\mathrm{EVPI} = z_{\mathrm{min}}^{\mathrm{EO}} - E[z_{\mathrm{min}}^{\mathrm{WS}}(\boldsymbol{\xi})]. \tag{34}$$

The concept is introduced to compare the here-and-now EO optimal solution and WS optimal solution by expectations of objective function values. For maximization programs, we write:

$$\mathrm{EVPI} = E[z_{\mathrm{max}}^{\mathrm{WS}}(\boldsymbol{\xi})] - z_{\mathrm{max}}^{\mathrm{EO}}.$$

Remark 56 (About EVPI) The EVPI denotes the Expected Value of Perfect Information. It measures how much it is reasonable to pay to obtain perfect information about the future. A small value of EVPI informs about little savings when we reach perfect information; the large EVPI says that the information about the future is valuable. If we obtain only sample information, the improvement on the optimum value is called the expected value of sample information (EVSI).

Example 14 (EVPI computed) For Example 3, we obtain from Example 5 that $E[z_{\mathrm{min}}^{\mathrm{WS}}(\xi)] = 0$ and as from Example 12 $z_{\mathrm{min}}^{\mathrm{EO}} = 0.24$, we compute $\mathrm{EVPI} = z_{\mathrm{min}}^{\mathrm{EO}} - E[z_{\mathrm{min}}^{\mathrm{WS}}(\xi)] = 0.24 - 0 = 0.24$. Therefore, for this specific application it is valuable to learn more about possible realizations of ξ (from the viewpoint considering the EO objective function).

Exercise 14 *Develop your own simple examples (analogue to Example 3) and compute EEV, VSS, and EVPI values. Decide which deterministic reformulations you have to build. Change the program data to see various values of characteristics computed.*

[9]The EVPI abbreviation is traditionally used in stochastic programming literature and in stochastic models of operations research as well.

Theorem 1 (Inequalities) Let $z^{\text{WS}}_{\min}(\boldsymbol{\xi}) = \min\{f(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})\}$,
$z^{\text{EO}}_{\min} = \min\{E[f(\mathbf{x}, \boldsymbol{\xi})]\}$, $z^{\text{EV}}_{\min} = \min\{f(\mathbf{x}, E\boldsymbol{\xi})\}$, and $EEV = E[f(\mathbf{x}^{\text{EV}}_{\min}, \boldsymbol{\xi})]$.
Then: $E[z^{\text{WS}}_{\min}(\boldsymbol{\xi})] \leq z^{\text{EO}}_{\min} \leq EEV$. In addition, if f is convex in $\boldsymbol{\xi}$ then also $z^{\text{EV}}_{\min} \leq E[z^{\text{WS}}_{\min}(\boldsymbol{\xi})]$.

Proof: $\forall \boldsymbol{\xi}^s \in \Xi : z^{\text{WS}}_{\min}(\boldsymbol{\xi}^s) \leq f(\mathbf{x}^{\text{EO}}_{\min}, \boldsymbol{\xi})$ as we deal with the optimal solution of the WS problem. Taking expectation of both sides yields in the first inequality. As $\mathbf{x}^{\text{EO}}_{\min}$ is the optimal solution of the EO problem then $z^{\text{EO}}_{\min} = E[f(\mathbf{x}^{\text{EO}}_{\min}, \boldsymbol{\xi})] \leq E[f(\mathbf{x}^{\text{EV}}_{\min}, \boldsymbol{\xi})]$. If in addition f is convex in $\boldsymbol{\xi}$ the last inequality is a direct consequence of Jensen's inequality. $\square$

Remark 57 (Extended readings) Read Chapter from Birge-Louveaux on EVPI and VSS.

LECTURE 5: RISK AVERSE BASICS

Remark 58 (Another objective function) Till now, we have used the expected value of $f(\mathbf{x}, \boldsymbol{\xi})$ as a good criterion to compare and find optimal solutions. The basic idea was to minimize "average costs". The idea is realistic when we have the chance to apply such a policy many times in the future. However, the average costs do not guarantee that there are no outlying costs. Therefore, we may think about some other criterion that is more "risk averse". As we have computed variance before, we discuss it now.

Exercise 15 *Try to design your own deterministic reformulations, give their interpretations and discuss their properties (e.g., derived from probability theory) suitable for optimization.*

Definition 14 (VO deterministic reformulation) Let the UP (see Definition 6 for details) be given:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}, \boldsymbol{\xi}) \mid \mathbf{x} \in \mathbb{R}^n\},$$

We define its here-and-now *variance objective (VO) deterministic reformulation* (VO program):

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{\operatorname{var}[f(\mathbf{x}, \boldsymbol{\xi})] \mid \mathbf{x} \in \mathbb{R}^n\}. \tag{35}$$

We denote the minimal objective function value as $z^{\text{VO}}_{\min}$ and minimum as $\mathbf{x}^{\text{VO}}_{\min}$. As before, for comparisons, we introduce ζ^{VO} i.e. $\zeta^{\text{VO}} = f(\mathbf{x}^{\text{VO}}_{\min}, \boldsymbol{\xi})$. Notice please that VO program (35) is also an unconstrained one.

Remark 59 (Standard deviation) Another choice could be to use a standard deviation $\sqrt{\operatorname{var}[f(\mathbf{x}, \boldsymbol{\xi})]}$ as an objective function value. Although it is used often for evaluation purposes, it is seldom chosen for the objective function, usually for two reasons: (1) Because of strict monotonicity of $\sqrt{\cdot}$, the set of the optimal solutions is the same[10] as for VO program, (2) $\sqrt{\cdot}$ can easily create numerical problems in some basic optimization algorithms because of its domain.

[10] See, e.g., Sundaram, R.K.: A First Course in Optimization Theory, Cambridge University Press, 1996, page 76, Theorem 2.5

Example 15 (VO computed) We formulate and solve the VO program for Example 3:

$$\min\{\mathrm{var}[(x-\xi)^2] \mid x \in \mathbb{R}\}, \quad \xi \sim Be(0.6).$$

We solve this program using probability and Calculus formulas:

$$\begin{aligned}\mathrm{var}[(x-\xi)^2] &= 0.6((x-1)^2)^2 + 0.4((x-0)^2)^2 - \\ &\quad (0.6(x-1)^2 + 0.4(x-0)^2)^2.\end{aligned}$$

We simplify the term expressing the VO objective and compute the optimal solution:

$$\begin{aligned}\mathrm{var}[(x-\xi)^2] &= 0.6(x^4 - 4x^3 + 6x^2 - 4x + 1) + 0.4x^4 - \\ &\quad (0.6(x^2 - 2x + 1) + 0.4x^2)^2 \\ &= x^4 - 2.4x^3 + 3.6x^2 - 2.4x + 0.6 - \\ &\quad x^4 + 2.4x^3 - 2.64x^2 + 1.44x - 0.36 \\ &= 0.96x^2 - 0.96x + 0.24. \\ (\mathrm{var}[(x-\xi)^2])' &= 2(0.96)x - 0.96 \quad \text{and as} \\ (\mathrm{var}[(x-\xi)^2])'' &= 1.92 > 0 \quad \Rightarrow \\ x_{\min}^{\mathrm{VO}} &= 0.5, \quad \text{and} \quad z_{\min}^{\mathrm{VO}} = 0.0625.\end{aligned}$$

We may add that $E[\zeta_{\min}^{\mathrm{VO}}] = 0.25$.

Exercise 16 *Using MATLAB, draw a figure for the objective function of Example 15. Develop and solve similar simple examples.*

Example 16 (VO computed — continuation) We consider the UP as in Example 3 and the VO deterministic reformulation as in Example 15. However, we do not discuss details about the probability distribution of ξ, we only assume that all further needed moments exist and they are finite. Therefore:

$$\min\{\mathrm{var}[(x-\xi)^2] \mid x \in \mathbb{R}\},$$

We solve this program again using probability and Calculus:

$$\begin{aligned}\mathrm{var}[(x-\xi)^2] &= E[(x-\xi)^4] - (E[(x-\xi)^2])^2 = \\ &\quad x^4 - 4x^3 E\xi + 6x^2 E[\xi^2] - 4E[\xi^3]x + E[\xi^4] - \\ &\quad (x^2 - 2xE\xi + E[\xi^2])^2 \\ &= 4(E[\xi^2] - (E\xi)^2)x^2 + 4(E[\xi^2]E\xi - E[\xi^3])x + \\ &\quad E[\xi^4] - (E[\xi^2])^2.\end{aligned}$$

We compute derivatives and we obtain the VO optimal solution:

$$\begin{aligned}(\mathrm{var}[(x-\xi)^2])' &= 8(E[\xi^2] - (E\xi)^2)x + 4(E[\xi^2]E\xi - E[\xi^3]), \\ (\mathrm{var}[(x-\xi)^2])'' &= 8(E[\xi^2] - (E\xi)^2) = 8\mathrm{var}[\xi] > 0 \quad \Rightarrow \\ x_{\min}^{\mathrm{VO}} &= \frac{E[\xi^3] - E[\xi^2]E\xi}{2(E[\xi^2] - (E\xi)^2)}.\end{aligned}$$

Exercise 17 *Check that the result* $x^{VO}_{min} = 0.5$ *of Example 15 could be obtained from the result of Example 16 by substitution.*

Exercise 18 *Compute*

$$\zeta^{VO} = \left(\frac{E[\xi^3]-E[\xi^2]E\xi}{2(E[\xi^2]-(E\xi)^2)} - \xi\right)^2$$

for Example 16 and also compute its characteristics.

Exercise 19 *Try to use the MATLAB Symbolic Toolbox for computations similar to those in Example 16.*

Remark 60 (Minmax idea) In fact, we have introduced the VO program to avoid large fluctuations of $f(\mathbf{x}, \boldsymbol{\xi})$. We can be even more strict and we can decide to minimize the maximum of fluctuations, therefore, minimize $\max_{\xi}\{f(\mathbf{x}, \boldsymbol{\xi})\}$. We introduce:

Definition 15 (MM deterministic reformulation) Let the UP (see Definition 6 for details) be given:

$$? \in \operatorname*{argmin}_{\mathbf{x}}\{f(\mathbf{x}, \boldsymbol{\xi}) \mid \mathbf{x} \in \mathbb{R}^n\},$$

We define its here-and-now *min-max (MM) deterministic reformulation* (MM program):

$$? \in \operatorname*{argmin}_{\mathbf{x}}\{\max_{\xi^s}\{f(\mathbf{x}, \boldsymbol{\xi}^s) \mid \boldsymbol{\xi}^s \in \Xi\} \mid \mathbf{x} \in \mathbb{R}^n\}. \tag{36}$$

We denote the minimal objective function value as $z^{\mathrm{MM}}_{\min}$ and minimum as $\mathbf{x}^{\mathrm{MM}}_{\min}$. As before, for comparisons, we introduce ζ^{MM} i.e. $\zeta^{\mathrm{MM}} = f(\mathbf{x}^{\mathrm{MM}}_{\min}, \boldsymbol{\xi})$. Notice please that MM program (36) is also an unconstrained one.

Example 17 (MM solution) We formulate and solve the MM program for Example 3:

$$\begin{aligned}\min_{x}\{\max_{\xi^s}\{(x-\xi^s)^2 \mid \xi^s \in \{0,1\}\} \mid x \in \mathbb{R}\} = \\ \min_{x}\{\max\{x^2, (x-1)^2\} \mid x \in \mathbb{R}\}.\end{aligned} \tag{37}$$

We solve this program by using a trick based on the concept of epigraph of the objective function (see NLP). So, we may rewrite the program (37) as:

$$? \in \operatorname*{argmin}_{x,z}\{z \mid z \geq x^2, z \geq (x-1)^2, x, z \in \mathbb{R}\}. \tag{38}$$

The program (38) could be easily solved (e.g., graphically). We obtain $x^{\mathrm{MM}}_{\min} = 0.5$ and $z^{\mathrm{MM}}_{\min} = 0.25$.

Exercise 20 *Solve (38) graphically. Discuss possible solution techniques for (36) when* Ξ *is a large set and when* $\mathbf{x} \in \mathbb{R}^n, n > 1$.

Remark 61 (More criteria) We have seen various optimization criteria for HN deterministic reformulations. The question is what to do if we would like to follow several criteria at once. In fact, it is the same question as we have solved till now, however, on the higher level of abstraction. Let us explain the idea.

Previously, we discussed the problem how to decide in front of future realizations $\boldsymbol{\xi}^s$. With discrete and finite probability distribution of $\boldsymbol{\xi}$, we could interpret it as a multicriteria optimization problem with many objective functions $f(\mathbf{x}, \boldsymbol{\xi}^s)$, $\boldsymbol{\xi}^s \in \Xi$ and their importance is weighted by $p_s = P(\boldsymbol{\xi} = \boldsymbol{\xi}^s)$. In such a situation, our EO deterministic reformulation is equivalent to the choice of *scalar reformulation* based on the weighted average in multicriteria optimization. For some of other deterministic reformulations, we may find also its counterpartners among scalar reformulations in multicriteria optimization. Underlying multicriteria optimization program[11] is usually specified as follows:

$$? \in \operatorname{argmin}\{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in C\}, \tag{39}$$

where $\mathbf{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is a vector objective function. Again the description is syntactically correct, however, without semantics. Therefore, to solve the problem, at first, we have to specify a scalar reformulation. It is often based on some aggregation of $\mathbf{f}$ components as, e.g., weighted average[12].

Remark 62 (Aggregating EO and VO) Applying the ideas of the previous paragraph on the higher level, we may try to think how to aggregate EO and VO objective functions in the same program[13]:

$$? \in \operatorname{argmin}\{(E[f(\mathbf{x}, \boldsymbol{\xi})], \operatorname{var}[f(\mathbf{x}, \boldsymbol{\xi})]) \mid \mathbf{x} \in \mathbb{R}^n\}. \tag{40}$$

Remark 63 (Convex combination) For program (40), we may find the following formulation often used:

$$\min\{\lambda E[f(\mathbf{x}, \boldsymbol{\xi})] + (1-\lambda)\operatorname{var}[f(\mathbf{x}, \boldsymbol{\xi})] \mid \mathbf{x} \in \mathbb{R}^n\}, \tag{41}$$

where $\lambda \in [0, 1]$ is a weighting parameter chosen by the wish of the modeller to prefer either low costs or smaller fluctuations.

Exercise 21 *Solve program (41) for data of Example 3 and* $\lambda = 0.5$*.*

Exercise 22 *Consider program (41) with data from Example 3 and analyze it as a parametric program i.e. mainly discuss the dependence of the optimal solution and optimal objective function value of (41) on* λ*.*

Exercise 23 *Think about the MATLAB implementation to visualize changes of optimal solution and optimal objective with respect to the change of* λ*.*

Remark 64 (Utility function) In addition to the choice of convex combination (41), we have to mention that aggregation may be formulated in more general way. When

[11] See,e.g., Steuer: Multicriteria Optimization, for more information.

[12] See also the goal programming chapters in OR books.

[13] The EO and VO objective functions already processed ("aggregated") randomness $\boldsymbol{\xi}$.

some financial costs are evaluated, very often the concept of a *utility function* is introduced. This function usually changes the scale of the cost function applying some "outer" nonlinear mapping to its values. Hence, for the cost function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, the utility function $u : \mathbb{R} \longrightarrow \mathbb{R}$ use will lead to the new composed objective function $u(f(\mathbf{x}))$. This could be applied to the multicriteria optimization case as for $\mathbf{f} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, we may have $u : \mathbb{R}^m \longrightarrow \mathbb{R}$ that plays the role of aggregating and utility function at once[14]. The use of utility functions will be discussed in detail later.

Exercise 24 *Search literature, find and list different utility functions. Notice their properties. Why is it important that for minimization programs, the utility functions are usually increasing convex functions? Try to design your own original utility functions, explain reasons for definition of their form.*

Remark 65 (Notation overview)

WS	$f^{\mathrm{WS}}(\mathbf{x}(\boldsymbol{\xi}))$	$=$	$f(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})$	$\mathbf{x}^{\mathrm{WS}}_{\min}(\boldsymbol{\xi})$	$z^{\mathrm{WS}}_{\min}(\boldsymbol{\xi})$	ζ^{WS}
IS	$f^{\mathrm{IS}}(\mathbf{x})$	$=$	$f(\mathbf{x}, \boldsymbol{\xi}^s)$	$\mathbf{x}^{\mathrm{IS}}_{\min}$	$z^{\mathrm{IS}}_{\min}$	ζ^{IS}
EV	$f^{\mathrm{EV}}(\mathbf{x})$	$=$	$f(\mathbf{x}, E\boldsymbol{\xi})$	$\mathbf{x}^{\mathrm{EV}}_{\min}$	$z^{\mathrm{EV}}_{\min}$	ζ^{EV}
EO	$f^{\mathrm{EO}}(\mathbf{x})$	$=$	$E[f(\mathbf{x}, \boldsymbol{\xi})]$	$\mathbf{x}^{\mathrm{EO}}_{\min}$	$z^{\mathrm{EO}}_{\min}$	ζ^{EO}
VO	$f^{\mathrm{VO}}(\mathbf{x})$	$=$	$\mathrm{var}[f(\mathbf{x}, \boldsymbol{\xi})]$	$\mathbf{x}^{\mathrm{VO}}_{\min}$	$z^{\mathrm{VO}}_{\min}$	ζ^{VO}
MM	$f^{\mathrm{MM}}(\mathbf{x})$	$=$	$\max_{\boldsymbol{\xi}^s}[f(\mathbf{x}, \boldsymbol{\xi}^s)]$	$\mathbf{x}^{\mathrm{MM}}_{\min}$	$z^{\mathrm{MM}}_{\min}$	ζ^{MM}

We may further denote, e.g., $E[f(\mathbf{x}^{\mathrm{EV}}_{\min}, \boldsymbol{\xi})]$ as $\mathbf{x}^{\mathrm{EV}}_{\min}$ at f^{EO} (see the table in Remark 66) or even shortly: EV@EO or $x@f$.

Remark 66 (Tables for comparisons)

	WS	IS	EV	EO	VO	MM
WS	ζ^{WS}	$f^{\mathrm{IS}}(\mathbf{x}^{\mathrm{WS}}_{\min}(\boldsymbol{\xi}^s))$	$f^{\mathrm{EV}}(\mathbf{x}^{\mathrm{WS}}_{\min}(E\boldsymbol{\xi}))$	$E[\zeta^{\mathrm{WS}}]$	$\mathrm{var}[\zeta^{\mathrm{WS}}]$	$f^{\mathrm{MM}}(\mathbf{x}^{\mathrm{WS}}_{\min}(\boldsymbol{\xi}))$
IS	ζ^{IS}	$z^{\mathrm{IS}}_{\min}$	$f^{\mathrm{EV}}(\mathbf{x}^{\mathrm{IS}}_{\min})$	$f^{\mathrm{EO}}(\mathbf{x}^{\mathrm{IS}}_{\min})$	$f^{\mathrm{VO}}(\mathbf{x}^{\mathrm{IS}}_{\min})$	$f^{\mathrm{MM}}(\mathbf{x}^{\mathrm{IS}}_{\min})$
EV	ζ^{EV}	$f^{\mathrm{IS}}(\mathbf{x}^{\mathrm{EV}}_{\min})$	$z^{\mathrm{EV}}_{\min}$	$f^{\mathrm{EO}}(\mathbf{x}^{\mathrm{EV}}_{\min})$	$f^{\mathrm{VO}}(\mathbf{x}^{\mathrm{EV}}_{\min})$	$f^{\mathrm{MM}}(\mathbf{x}^{\mathrm{EV}}_{\min})$
EO	ζ^{EO}	$f^{\mathrm{IS}}(\mathbf{x}^{\mathrm{EO}}_{\min})$	$f^{\mathrm{EV}}(\mathbf{x}^{\mathrm{EO}}_{\min})$	$z^{\mathrm{EO}}_{\min}$	$f^{\mathrm{VO}}(\mathbf{x}^{\mathrm{EO}}_{\min})$	$f^{\mathrm{MM}}(\mathbf{x}^{\mathrm{EO}}_{\min})$
VO	ζ^{VO}	$f^{\mathrm{IS}}(\mathbf{x}^{\mathrm{VO}}_{\min})$	$f^{\mathrm{EV}}(\mathbf{x}^{\mathrm{VO}}_{\min})$	$f^{\mathrm{EO}}(\mathbf{x}^{\mathrm{VO}}_{\min})$	$z^{\mathrm{VO}}_{\min}$	$f^{\mathrm{MM}}(\mathbf{x}^{\mathrm{VO}}_{\min})$
MM	ζ^{MM}	$f^{\mathrm{IS}}(\mathbf{x}^{\mathrm{MM}}_{\min})$	$f^{\mathrm{EV}}(\mathbf{x}^{\mathrm{MM}}_{\min})$	$f^{\mathrm{EO}}(\mathbf{x}^{\mathrm{MM}}_{\min})$	$f^{\mathrm{VO}}(\mathbf{x}^{\mathrm{MM}}_{\min})$	$z^{\mathrm{MM}}_{\min}$

Search in the table: EEV, VSS, $E[z^{\mathrm{WS}}_{\min}(\boldsymbol{\xi})]$, and EVPI.

Exercise 25 *Fill the table in Remark 66 with values for Example 3.*

[14]Some authors also consider the use of deterministic reformulations for UPs as the application of some utility function in wide sense.

LECTURE 6: LARGE DISCRETE CASES

2.3 The use of modeling language

Example 18 (Large) We generalize our educational Example 3 and we introduce an underlying program as follows:

$$? \operatorname*{argmin}_{\mathbf{x} \in \mathrm{I\!R}^n} \{ (\mathbf{D}(\boldsymbol{\xi})\mathbf{x} + \mathbf{c}(\boldsymbol{\xi}))^\top (\mathbf{D}(\boldsymbol{\xi})\mathbf{x} + \mathbf{c}(\boldsymbol{\xi})) \mid \mathbf{x} \geq \mathbf{0} \}, \tag{42}$$

where $\forall s \in \mathcal{S} : \mathbf{D}(\boldsymbol{\xi}^s) \in \mathrm{I\!R}^{n \times n}, \mathbf{c}(\boldsymbol{\xi}^s) \in \mathrm{I\!R}^n$. We denote elements of Ξ by indices from $\mathcal{S}$ as $|\Xi| < \aleph_0$.

Exercise 26 *Discuss whether Example 3 can be considered as a special case of Example 18.*

Example 19 (Large – continuation) We write the GAMS program `SProg.GMS` to solve (42):

```
$OFFLISTING
$EOLCOM //
$INLINECOM { }
$TITLE Various deterministic reformulations for underlying programs
* Options:
OPTION LIMROW   = 0;   OPTION LIMCOL   = 0;
OPTION SOLPRINT = OFF; OPTION SYSOUT   = OFF;
```

At first, the compiler directive `$` switches off the output of the listing in the standard `SProg.LST` output file. The next directives extend the used comment symbols[15]. `$TITLE` names the problem solved. `OPTION`s further reduce the amount of the output placed in `SProg.LST`. `*` as the first row symbol denotes a comment.

```
SETS // All indices:
  J indices of variables        /   1*2   /,
  I indices of constraints      /   1*2   /,
  K dimension of random vector  /   1*2   /,
  S indices of realizations     / S01*S05 /,
  SChosen(S) subset for IS      / S01     /,
  TYPE reformulation type       / "WS", "IS", "EV", "EO", "VO", "MM" /;
```

`SETS` section denotes all indices used later. In our case $|\mathcal{J}| = n = 2$ for dimension of $\mathbf{x}$, $|\mathcal{I}| = m = 2$ (for the future use of constraints), $|\mathcal{K}| = K = 2$ for dimension of $\boldsymbol{\xi}$, $|\Xi| = |\mathcal{S}| = S = 5$ for a number of realizations $\boldsymbol{\xi}^s$. One scenario used for IS approach is identified by `SChosen`. Abbreviations already utilized for different types of deterministic reformulations are listed in `TYPE` set.

Exercise 27 *Before you proceed, derive various deterministic reformulations for the underlying program (42).*

[15] Anytime you need details about the GAMS, check its help files or `www.gams.com`.

Example 20 (Large – continuation 2) Random parameters have to be specified in the underlying program in detail:

```
SCALARS    // Randomness:
  LSUPPORT lower bound for distribution support Xi /  -6    /,
  USUPPORT upper bound for distribution support Xi /   6    /,
  DOMINANT initial constant for the diagonal of Q  /   1    /,
  RANGE    range for some random initialization    /   1    /,
  INITA / 1 /,INITB / 1 /, EEV, VSS, EVPI scalars for known characteristics;
```

`SCALARS` section introduces needed information for random generators (mainly support ranges) and also further identifiers for future evaluations of results.

```
PARAMETERS XI(K,S) "realizations (scenarios) of xi",
  P(S) probabilities of scenarios;
OPTION SEED  = 123456;             // seed initializes random generator
* For all xi components and all scenarios s,random values are generated
XI(K,S)      = UNIFORM(LSUPPORT, USUPPORT);
ALIAS(S,SS);                       // S and SS are synonyma
P(S)         = UNIFORM(0,1);       // probabilities are generated
P(S)         = P(S)/SUM(SS, P(SS)); //  and computed
```

Realizations $\xi_k^s, k \in \mathcal{K}, s \in \mathcal{S}$ and related probabilities $p_s, s \in \mathcal{S}$ are generated to get educational data.

Exercise 28 *Develop your own alternative procedures generating primary data in various ways.*

Example 21 (Large – continuation 3) In practical applications, large-dimensional random inputs for stochastic programs very often depend on the primary random input of much smaller dimension.

```
PARAMETERS
  ZETA(S, TYPE)   values of zeta random variables,
  EZETA( TYPE)    expected values of zeta,
  VARZETA(TYPE)   variances of zeta,
  XSOL(J,S, TYPE) optimal solutions for reformulations,
  AT(TYPE,TYPE)   AT table contains comparison of reformulations,
  D(J,J),   C(J),   A(I,J),   B(I),   // model related parameters
  DS(J,J,S),CS(J,S),AS(I,J,S),BS(I,S),// parameters for scenarios
  DD(J,J,K),CC(J,K),AA(I,J,K),BB(I,K);// primary data parameters
ALIAS(J,JJ); // JJ is used as another index J for square matrix D
* Initialization of primary data:
DD(J,JJ,K) = UNIFORM(-RANGE,RANGE); CC(J,K) = UNIFORM(-RANGE,RANGE);
AA(I,J,K)  = UNIFORM(-RANGE,RANGE); BB(I,K) = UNIFORM(-RANGE,RANGE);
* Emptying scenario-related parameters:
DS(J,JJ,S)  = 0; CS(J,S) = 0;
* Filling some scenario parameters with initial values:
AS(I,J,S)   = INITA; BS(I,S) = INITB;
* To achieve positive definite DS and so convexity of the objective:
DS(J,J,S)   = DOMINANT;
```

```
* Scenario coefficients are derived by linear transformation
* from xi values and primary data:
DS(J,JJ,S)  = DS(J,JJ,S) + SUM(K, DD(J,JJ,K) * XI(K,S));
CS(J,S)     = CS(J,S)    + SUM(K, CC(J,K)    * XI(K,S));
AS(I,J,S)   = AS(I,J,S)  + SUM(K, AA(I,J,K)  * XI(K,S));
BS(I,S)     = BS(I,S)    + SUM(K, BB(I,K)    * XI(K,S));
```

Various parameters for the further programs are introduced, randomly initialized and then scenario-related parameters are obtained from primary data by linear transformations $\forall s \in \mathcal{S}$:

$$
\begin{aligned}
\mathbf{D}(\boldsymbol{\xi}^s) &= \mathbf{D}_0(\boldsymbol{\xi}^s) + \sum_{k\in\mathcal{K}} \mathbf{D}_k \xi_k^s, \quad \mathbf{c}(\boldsymbol{\xi}^s) = \mathbf{c}_0(\boldsymbol{\xi}^s) + \sum_{k\in\mathcal{K}} \mathbf{c}_k \xi_k^s, \\
\mathbf{A}(\boldsymbol{\xi}^s) &= \mathbf{A}_0(\boldsymbol{\xi}^s) + \sum_{k\in\mathcal{K}} \mathbf{A}_k \xi_k^s, \quad \mathbf{b}(\boldsymbol{\xi}^s) = \mathbf{b}_0(\boldsymbol{\xi}^s) + \sum_{k\in\mathcal{K}} \mathbf{b}_k \xi_k^s.
\end{aligned}
$$

Exercise 29 *Give particular cases of the approach used in Example 21.*

Example 22 (Large – continuation 4) For further use, weighted averages of scenario-related values are computed and all input data is displayed:

```
* At first averages computed and assigned for constraints:
A(I,J)      = SUM(S,P(S)*AS(I,J,S)); B(I)  = SUM(S,P(S)*BS(I,S));
D(J,JJ)     = SUM(S,P(S)*DS(J,JJ,S));C(J)  = SUM(S,P(S)*CS(J,S));
* All inputs are displayed:
DISPLAY S,XI,P, K,J,DD,CC, I,AA,BB, DS,CS,AS,BS, D,C,A,B;
```

Then, the important list of specifications of variables and constraints follows:

```
* Variables and constraints:
VARIABLES
  Z    general value for an objective function,
  X(J) general variable for optimization problem;
POSITIVE VARIABLES X(J); // decision variable is nonnegative
* General constraints:
EQUATIONS
  OBJ        general objective of the quadratic form,
  CONSTR(I)  general demand-like linear constraints;
OBJ..           Z =E= SUM(J,(SUM(JJ, D(J,JJ)*X(JJ)) + C(J))*
                            (SUM(JJ, D(J,JJ)*X(JJ)) + C(J)));
CONSTR(I)..  B(I) =L= SUM(J,A(I,J)*X(J));
```

Notice that the objective function value is named explicitly as `Z`. In this way, the objective function becomes a constraint. All constraints are named and they could be indexed. After declaration of names (and the related allocation of memory), constraints are defined, so specified by formulas. The presence of linear constraints is used to allow a reader to do some later experiments with constrained programs where constraints do not involve and involve random parameters.

Exercise 30 *Becoming an expert in the use of this program, try to modify the objective function in the non-trivial way that you like.*

Example 23 (Large – continuation 5) We continue now with the specification and solution of reformulations. We begin with WS. The other cases follow the similar scheme. At first, the model is declared by the list of constraints involved. Then, output file OUT.TXT is open and heading is written.

```
* WAIT-AND-SEE APPROACH:
* ======================
MODEL WS / OBJ
// CONSTR
/;
FILE OUT / "OUT.TXT" /; // defines the internal end external output files
PUT OUT;                // opens the file for the output
PUT / @1 "OUTPUT FOR WS:" /; PUT @1 "--------------" / /;// output heading
PUT @1 "Scen.:",@10"Prob.:",@20"Obj.f:",@30"Type",@35"Sol?",@40"Zeta:",@50;
LOOP(J,PUT"  X(",J.TL:3:0,'): ';);LOOP(K,PUT"  Xi(",K.TL:3:0,'):';);PUT /;
LOOP(S, // forall scenarios:
  D(J,JJ) = DS(J,JJ,S); // Updates model values with scenario values
  C(J)    = CS(J,S);    // A(I,J)  = AS(I,J,S); B(I)    = BS(I,S);
  SOLVE WS MINIMIZING Z USING NLP; // Solves the program
  ZETA(S,"WS") = Z.L; XSOL(J,S,"WS") = X.L(J);
  PUT @1 S.TL:8,@10 P(S):8:6,@20 Z.L:9:4,@30 WS.MODELSTAT:3:0,
    @35 WS.SOLVESTAT:3:0, @40 ZETA(S,"WS"):9:4, @50;
  LOOP(J, PUT X.L(J):9:4, ' ';);LOOP(K, PUT XI(K,S):9:4, ' ';); PUT /;);
EZETA("WS")   = SUM(S,P(S)*ZETA(S,"WS"));
VARZETA("WS") = SUM(S,P(S)*(ZETA(S,"WS")-SUM(SS,P(SS)*ZETA(SS,"WS")))*
                           (ZETA(S,"WS")-SUM(SS,P(SS)*ZETA(SS,"WS"))));
PUT / "E(zetaWS)   = ",@40   EZETA("WS"):9:4;
PUT / "var(zetaWS) = ",@40 VARZETA("WS"):9:4 / /;
```

Then, within the LOOP statement program data is modified repeatedly as the current scenario changes. The model is SOLVEd. The results are copied into tables further used for additional computations and outputs that are generated in the output file. For syntax rules and details look at the GAMS Help. For the input data given, the results are obtained:

```
OUTPUT FOR WS:
--------------
Scen.:   Prob.:    Obj.f:    Type Sol? Zeta:      X(1  ):   X(2  ):   Xi(1  ):  Xi(2  ):
S01      0.149413    0.0000   2    1     0.0000    0.7383    0.4503   -5.3951    0.1672
S02      0.060308    0.0000   2    1     0.0000    3.2992    3.0747    2.2398    2.0303
S03      0.410657    0.0090   2    1     0.0090    0.0000    0.0038   -0.2018    0.1942
S04      0.069502    2.8552   2    1     2.8552    0.6323    0.0000    2.8292   -3.3289
S05      0.310120    0.0000   2    1     0.0000    2.9565    1.1339   -0.5379   -4.6167
E(zetaWS)   =                            0.2021
var(zetaWS) =                            0.5258
```

Exercise 31 *Analyze the previous source code and give the formulas by using mathematical notation.*

Exercise 32 *Learn details about the used GAMS report writing (output) control.*

Example 24 (Large – continuation 6) For IS reformulation we write:

```
* INDIVIDUAL SCENARIO APPROACH:
* =============================
MODEL IS / OBJ
*  CONSTR
/; // FILE OUTIS / "OUTIS.TXT" /; PUT OUTIS;
PUT / @1 "OUTPUT FOR IS:" /; PUT @1 "--------------" / /;
PUT @1"Scen.:",@10 "Prob.:",@20"Obj.f:",@30 "Type",@35"Sol?",@40"Zeta:",@50;
LOOP(J, PUT "  X(",J.TL:3:0,'): ';);
LOOP(K, PUT "  Xi(",K.TL:3:0,'):';);PUT /;
LOOP(SChosen, // For chosen scenario:
  D(J,JJ) = DS(J,JJ,SChosen);
  C(J)    = CS(J,SChosen); // A(I,J)=AS(I,J,SChosen);B(I)=BS(I,SChosen);
  SOLVE IS MINIMIZING Z USING NLP;);
LOOP(S,ZETA(S,"IS") = SUM(J,(SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S))*
                            (SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S)));
  XSOL(J,S,"IS") = X.L(J);
  PUT @1 S.TL:8,@10 P(S):8:6,@20 Z.L:9:4,@30 IS.MODELSTAT:3:0,
    @35 IS.SOLVESTAT:3:0, @40 ZETA(S,"IS"):9:4, @50;
  LOOP(J, PUT X.L(J):9:4, ' ';); LOOP(K, PUT XI(K,S):9:4, ' ';);PUT /;);
EZETA("IS")   = SUM(S, P(S)*ZETA(S,"IS"));
VARZETA("IS") = SUM(S, P(S)*(ZETA(S,"IS") - SUM(SS, P(SS)*ZETA(SS,"IS")))*
  (ZETA(S,"IS") - SUM(SS, P(SS)*ZETA(SS,"IS"))));
PUT / "E(zetaIS)   = ", @40 EZETA("IS"):9:4 ;
PUT / "var(zetaIS) = ", @40 VARZETA("IS"):9:4 / /;
```

We may learn more about generation of output tables. There are formatting directives as `,` `#`, `/` and control on number of characters and decimal places displayed. Results are in `OUT.TXT`:

```
OUTPUT FOR IS:
--------------
Scen.:   Prob.:    Obj.f:    Type Sol? Zeta:      X(1  ):   X(2  ):   Xi(1  ):  Xi(2  ):
S01      0.149413    0.0000   2    1      0.0000    0.7383    0.4503   -5.3951    0.1672
S02      0.060308    0.0000   2    1      7.8726    0.7383    0.4503    2.2398    2.0303
S03      0.410657    0.0000   2    1      0.9051    0.7383    0.4503   -0.2018    0.1942
S04      0.069502    0.0000   2    1      4.5996    0.7383    0.4503    2.8292   -3.3289
S05      0.310120    0.0000   2    1     12.5314    0.7383    0.4503   -0.5379   -4.6167
E(zetaIS)   =                             5.0524
var(zetaIS) =                            28.7178
```

Example 25 (Large – continuation 7) EV approach updates program data with computed expectations:

```
* EXPECTED VALUE APPROACH:
* ========================
MODEL EV /  OBJ
*  CONSTR
/; // FILE OUTEV / "OUTEV.TXT" /; PUT OUTEV;
PUT / @1 "OUTPUT FOR EV:" /;
PUT @1 "--------------" / /;
PUT @1"Scen.:",@10"Prob.:",@20"Obj.f:",@30"Type",@35"Sol?",@40 "Zeta:",@50;
LOOP(J,PUT "  X(",J.TL:3:0,'): ';);LOOP(K,PUT "  Xi(",K.TL:3:0,'):';);PUT /;
D(J,JJ) = SUM(S,P(S)* DS(J,JJ,S)); // A(I,J)  = SUM(S,P(S)*AS(I,J,S));
C(J)    = SUM(S,P(S)* CS(J,S));    // B(I)    = SUM(S,P(S)*BS(I,S));
```

```
SOLVE EV MINIMIZING Z USING NLP;
LOOP(S, ZETA(S,"EV") = SUM(J,(SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S))*
                             (SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S)));
  XSOL(J,S,"EV") = X.L(J);
  PUT @1 S.TL:8, @10 P(S):8:6, @20 Z.L:9:4, @30 EV.MODELSTAT:3:0,
    @35 EV.SOLVESTAT:3:0, @40 ZETA(S,"EV"):9:4, @50;
  LOOP(J, PUT X.L(J):9:4, ' ';); LOOP(K, PUT XI(K,S):9:4, ' ';);PUT /;);
EZETA("EV")   = SUM(S, P(S)*ZETA(S,"EV"));
VARZETA("EV") = SUM(S, P(S)*(ZETA(S,"EV")-SUM(SS,P(SS)*ZETA(SS,"EV")))*
  (ZETA(S,"EV") - SUM(SS, P(SS)*ZETA(SS,"EV"))));
PUT / "E(zetaEV)   = ", @40 EZETA("EV"):9:4;
PUT / "EEV         = ", @40 EZETA("EV"):9:4;
PUT / "var(zetaEV) = ", @40 VARZETA("EV"):9:4 / /;
```

The tables with results have the similar structure. First column identifies scenarios by indices from $\mathcal{S}$. The second column lists probabilities p_s. The third presents $z_{\min}^{\mathrm{EV}}$. It is the same here as for each scenario the same $\mathbf{x}_{\min}^{\mathrm{EV}}$ was applied. *However, compare with WS results!* Then the `Type` column contains 2 that means by the GAMS documentation identification of the locally optimal solution (minimum here). The `Sol?` column contains 1, so no error has appeared during the GAMS/MINOS solver run.

```
OUTPUT FOR EV:
--------------
Scen.:    Prob.:     Obj.f:    Type Sol? Zeta:       X(1  ):   X(2  ):   Xi(1  ):  Xi(2  ):
S01       0.149413     0.0000   2    1     8.4140    1.3441    0.4406   -5.3951    0.1672
S02       0.060308     0.0000   2    1    15.0496    1.3441    0.4406    2.2398    2.0303
S03       0.410657     0.0000   2    1     2.2881    1.3441    0.4406   -0.2018    0.1942
S04       0.069502     0.0000   2    1     5.7585    1.3441    0.4406    2.8292   -3.3289
S05       0.310120     0.0000   2    1     7.2779    1.3441    0.4406   -0.5379   -4.6167
E(zetaEV)   =   EEV =                      5.7616
var(zetaEV) =                             11.9215
```

Example 26 (Large – continuation 8) With EO approach, we have to change the objective function below. Previously used $f(\mathbf{x},\boldsymbol{\xi})$ in different forms as $f(\mathbf{x}(\boldsymbol{\xi}),\boldsymbol{\xi})$, $f(\mathbf{x},\boldsymbol{\xi}^s)$, and $f(\mathbf{x},E\boldsymbol{\xi})$ has to be changed now to the form $E[f(\mathbf{x},\boldsymbol{\xi})] = \sum_{s\in\mathcal{S}} p_s f(\mathbf{x},\boldsymbol{\xi}^s)$. See below, how it is implemented:

```
* EXPECTED OBJECTIVE APPROACH:
* ============================
EQUATIONS  OBJEO  EO objective of the quadratic form;
OBJEO..    Z =E= SUM(S, P(S) *
                     SUM(J,(SUM(JJ, DS(J,JJ,S)*X(JJ)) + CS(J,S))*
                           (SUM(JJ, DS(J,JJ,S)*X(JJ)) + CS(J,S))));
MODEL EO / OBJEO
*  CONSTR
/; // FILE OUT / "OUTEO.TXT" /; PUT OUTEO;
PUT / @1 "OUTPUT FOR EO:" /; PUT @1 "--------------" / /;
PUT @1"Scen.:",@10"Prob.:",@20"Obj.f:",@30 "Type",@35"Sol?",@40 "Zeta:",@50;
LOOP(J,PUT "  X(",J.TL:3:0,'): ';);LOOP(K,PUT "  Xi(",K.TL:3:0,'):';);PUT /;
SOLVE EO MINIMIZING Z USING NLP;
LOOP(S, ZETA(S,"EO") = SUM(J,(SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S))*
                             (SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S)));
  XSOL(J,S,"EO") = X.L(J);
  PUT @1 S.TL:8, @10 P(S):8:6, @20 Z.L:9:4, @30 EO.MODELSTAT:3:0,
```

```
    @35 EO.SOLVESTAT:3:0, @40 ZETA(S,"EO"):9:4, @50;
  LOOP(J,PUT X.L(J):9:4, ' ';);LOOP(K,PUT XI(K,S):9:4, ' ';);PUT /;);
EZETA("EO")    = SUM(S, P(S)*ZETA(S,"EO"));
VARZETA("EO") = SUM(S, P(S)*(ZETA(S,"EO") - SUM(SS, P(SS)*ZETA(SS,"EO")))*
  (ZETA(S,"EO") - SUM(SS, P(SS)*ZETA(SS,"EO"))));
PUT / "E(zetaEO)   = ", @40 EZETA("EO"):9:4;
PUT / "var(zetaEO) = ",@40 VARZETA("EO"):9:4 / /;
EVPI = EZETA("EO") - EZETA("WS"); PUT   "EVPI        = ", @40 EVPI:9:4 /;
VSS  = EZETA("EV") - EZETA("EO"); PUT   "VSS         = ", @40 VSS:9:4 / /;
```

The table of results also involves columns containing ζ^{EO} (for other cases, see $\zeta^{\odot}$), $x^{\mathrm{EO}}_{j,\min}, j \in \mathcal{J}$ and $\xi^s_k, k \in \mathcal{K}$ columns. With EO, we also compute EVPI and VSS.

```
OUTPUT FOR EO:
--------------
Scen.:   Prob.:    Obj.f:    Type Sol? Zeta:      X(1  ):   X(2  ):   Xi(1  ):  Xi(2  ):
S01      0.149413     4.6251   2    1     1.4522    1.0993    0.5749   -5.3951    0.1672
S02      0.060308     4.6251   2    1     9.0926    1.0993    0.5749    2.2398    2.0303
S03      0.410657     4.6251   2    1     1.7726    1.0993    0.5749   -0.2018    0.1942
S04      0.069502     4.6251   2    1     5.7003    1.0993    0.5749    2.8292   -3.3289
S05      0.310120     4.6251   2    1     8.8212    1.0993    0.5749   -0.5379   -4.6167
E(zetaEO)   =                             4.6251
var(zetaEO) =                            11.5901

EVPI        =                             4.4229
VSS         =                             1.1366
```

Example 27 (Large – continuation 9) The VO reformulation uses a different objective function of the form $\mathrm{var}[f(\mathbf{x}, \boldsymbol{\xi})]$.

```
* VARIANCE OBJECTIVE APPROACH:
* ============================
EQUATIONS OBJVO VO objective of the quadratic form;
OBJVO..   Z =E=   SUM(S, P(S) *
  (SUM(J,(SUM(JJ, DS(J,JJ,S)*X(JJ)) + CS(J,S))*
         (SUM(JJ, DS(J,JJ,S)*X(JJ)) + CS(J,S))) -
     SUM(SS, P(SS) * SUM(J,(SUM(JJ, DS(J,JJ,SS)*X(JJ)) + CS(J,SS))*
                           (SUM(JJ, DS(J,JJ,SS)*X(JJ)) + CS(J,SS))))) *
  (SUM(J,(SUM(JJ, DS(J,JJ,S)*X(JJ)) + CS(J,S))*
         (SUM(JJ, DS(J,JJ,S)*X(JJ)) + CS(J,S))) -
     SUM(SS, P(SS) * SUM(J,(SUM(JJ, DS(J,JJ,SS)*X(JJ)) + CS(J,SS))*
                           (SUM(JJ, DS(J,JJ,SS)*X(JJ)) + CS(J,SS))))));
MODEL VO / OBJVO
*  CONSTR
/; // FILE OUTVO / "OUTVO.TXT" /; PUT OUTVO;
PUT / @1 "OUTPUT FOR VO:" /; PUT @1 "--------------" / /;
PUT @1"Scen.:",@10"Prob.:",@20"Obj.f:",@30 "Type",@35"Sol?",@40"Zeta:",@50;
LOOP(J,PUT "  X(",J.TL:3:0,'): ';);LOOP(K,PUT "  Xi(",K.TL:3:0,'):';);PUT /;
SOLVE VO MINIMIZING Z USING NLP;
LOOP(S, ZETA(S,"VO") = SUM(J,(SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S))*
                             (SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S)));
  XSOL(J,S,"VO") = X.L(J);
  PUT @1 S.TL:8, @10 P(S):8:6, @20 Z.L:9:4, @30 VO.MODELSTAT:3:0,
    @35 VO.SOLVESTAT:3:0, @40 ZETA(S,"VO"):9:4, @50;
  LOOP(J,PUT X.L(J):9:4, ' ';);LOOP(K,PUT XI(K,S):9:4, ' ';);PUT /;);
```

```
EZETA("VO")   = SUM(S, P(S)*ZETA(S,"VO"));
VARZETA("VO") = SUM(S, P(S)*(ZETA(S,"VO") - SUM(SS, P(SS)*ZETA(SS,"VO")))*
  (ZETA(S,"VO") - SUM(SS, P(SS)*ZETA(SS,"EO"))));
PUT / "E(zetaVO)   = ", @40 EZETA("VO"):9:4;
PUT / "var(zetaVO) = ", @40 VARZETA("VO"):9:4 / /;
```

In the tables with results below, a reader finds information about $E[\zeta^{\mathrm{VO}}]$ and $\mathrm{var}[\zeta^{\mathrm{VO}}]$.

```
OUTPUT FOR VO:
--------------
Scen.:   Prob.:     Obj.f:    Type Sol? Zeta:      X(1  ):   X(2  ):   Xi(1  ):  Xi(2  ):
S01      0.149413     3.4070   2    1     6.6177    1.5514    0.8899   -5.3951    0.1672
S02      0.060308     3.4070   2    1     8.7424    1.5514    0.8899    2.2398    2.0303
S03      0.410657     3.4070   2    1     3.5384    1.5514    0.8899   -0.2018    0.1942
S04      0.069502     3.4070   2    1     8.8372    1.5514    0.8899    2.8292   -3.3289
S05      0.310120     3.4070   2    1     6.3715    1.5514    0.8899   -0.5379   -4.6167
E(zetaVO)   =                             5.5592
var(zetaVO) =                             3.4070
```

Exercise 33 *Write the VO objective function by using summation-index mathematical formulas.*

Example 28 (Large – continuation 10) The min-max deterministic reformulation $\min_{\mathbf{x}} \max_{\boldsymbol{\xi}^s}\{f(\mathbf{x},\boldsymbol{\xi}^s)\}$ is for the computational purposes already changed to the constrained form $\min_{z,\mathbf{x}}\{z \mid z \geq f(\mathbf{x},\boldsymbol{\xi}^s), \boldsymbol{\xi}^s \in \Xi\}$. The solution is in a tabular form as before.

```
* MIN-MAX APPROACH:
* =================
EQUATIONS OBJMM(S) MM modified constraints from the objective;
OBJMM(S).. Z =G= SUM(J,(SUM(JJ, DS(J,JJ,S)*X(JJ)) + CS(J,S))*
                    (SUM(JJ, DS(J,JJ,S)*X(JJ)) + CS(J,S)));
MODEL MM / OBJMM
*  CONSTR
/; // FILE OUT / "OUT.TXT" /; PUT OUT;
PUT / @1 "OUTPUT FOR MM:" /; PUT @1 "--------------" / /;
PUT @1"Scen.:",@10"Prob.:",@20"Obj.f:",@30"Type",@35"Sol?",@40"Zeta:",@50;
LOOP(J,PUT"  X(",J.TL:3:0,'): ';);LOOP(K,PUT"  Xi(",K.TL:3:0,'):';);PUT /;
SOLVE MM MINIMIZING Z USING NLP;
LOOP(S, ZETA(S,"MM") = SUM(J,(SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S))*
                         (SUM(JJ, DS(J,JJ,S)*X.L(JJ)) + CS(J,S)));
  XSOL(J,S,"MM") = X.L(J);
  PUT @1 S.TL:8, @10 P(S):8:6, @20 Z.L:9:4, @30 MM.MODELSTAT:3:0,
    @35 MM.SOLVESTAT:3:0, @40 ZETA(S,"MM"):9:4, @50;
  LOOP(J,PUT X.L(J):9:4, ' ';);LOOP(K, PUT XI(K,S):9:4, ' ';);PUT /;);
EZETA("MM")   = SUM(S, P(S)*ZETA(S,"MM"));
VARZETA("MM") = SUM(S, P(S)*(ZETA(S,"MM") - SUM(SS, P(SS)*ZETA(SS,"MM")))*
  (ZETA(S,"MM") - SUM(SS, P(SS)*ZETA(SS,"MM"))));
PUT / "E(zetaMM)   = ", @40 EZETA("MM"):9:4;
PUT / "var(zetaMM) = ", @40 VARZETA("MM"):9:4 / /;
```

```
OUTPUT FOR MM:
--------------
Scen.:   Prob.:     Obj.f:    Type Sol? Zeta:      X(1  ):   X(2  ):   Xi(1  ):  Xi(2  ):
S01      0.149413     7.7642   2    1     4.1142    1.3670    0.8283   -5.3951    0.1672
```

```
S02      0.060308     7.7642   2    1     7.7642    1.3670    0.8283    2.2398    2.0303
S03      0.410657     7.7642   2    1     2.8533    1.3670    0.8283   -0.2018    0.1942
S04      0.069502     7.7642   2    1     7.7642    1.3670    0.8283    2.8292   -3.3289
S05      0.310120     7.7642   2    1     7.7642    1.3670    0.8283   -0.5379   -4.6167
E(zetaMM)    =                          5.2022
var(zetaMM) =                           5.3304
```

Exercise 34 *Compare presented results for various deterministic reformulations. Interpret them.*

Exercise 35 *Use MATLAB and visualize results. Hint: Remember that $z \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^2$ and $\boldsymbol{\xi} \in \mathbb{R}^2$.*

Example 29 (Large – continuation 11) At the end of the "infinitely" continuing example, we present the part of the code that computes table with elements $x@f$, see Remark 66. The output writing code is contained only in last three rows.

```
* AT  TABLE:
* ==========
ALIAS(TYPE,TTYPE); AT(TYPE,TTYPE)=0;
LOOP(TYPE, AT(TYPE,"WS") = SUM(S, P(S) *
  SUM(J,(SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))*
        (SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))));););
LOOP(TYPE, AT(TYPE,"IS") = SUM(SChosen,
  SUM(J,(SUM(JJ, DS(J,JJ,SChosen)*XSOL(JJ,SChosen,TYPE))+CS(J,SChosen))*
        (SUM(JJ, DS(J,JJ,SChosen)*XSOL(JJ,SChosen,TYPE))+CS(J,SChosen))));););
LOOP(TYPE, AT(TYPE,"EV") =
  SUM(J,SUM(S, P(S)*(SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S)))*
        SUM(S, P(S)*(SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))));););
AT("WS","EV")=AT("EV","EV");
LOOP(TYPE, AT(TYPE,"EO") = SUM(S, P(S) *
    SUM(J,(SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))*
          (SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))));););
LOOP(TYPE, AT(TYPE,"VO") = SUM(S, P(S) *
    (SUM(J,(SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))*
           (SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))) -
  SUM(SS, P(SS)*SUM(J,(SUM(JJ,DS(J,JJ,SS)*XSOL(JJ,SS,TYPE))+CS(J,SS))*
                   (SUM(JJ, DS(J,JJ,SS)*XSOL(JJ,SS,TYPE)) + CS(J,SS)))))*
    (SUM(J,(SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))*
           (SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))) -
  SUM(SS, P(SS)*SUM(J,(SUM(JJ,DS(J,JJ,SS)*XSOL(JJ,SS,TYPE))+CS(J,SS))*
                   (SUM(JJ, DS(J,JJ,SS)*XSOL(JJ,SS,TYPE)) + CS(J,SS))))));););
LOOP(TYPE, AT(TYPE,"MM") =
  SMAX(S, SUM(J,(SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))*
               (SUM(JJ, DS(J,JJ,S)*XSOL(JJ,S,TYPE)) + CS(J,S))));););
PUT / /; PUT @8; LOOP(TTYPE, PUT TTYPE.TL:8:0); PUT /;
LOOP(TYPE,PUT TYPE.TL:4,@5;
  LOOP(TTYPE,PUT" ",AT(TYPE,TTYPE):7:3;);PUT/;);PUT/ /;
```

Explain why `AT("WS","EV")=AT("EV","EV")` *assignment is used above.* The reader sees that to minimize the number of various formulas above of type $f^{\odot}(\ )$ and related computations for different solutions $\mathbf{x}^{\odot}_{\min}$, we have stored the used values in the inefficient way. *It is a challenge for a reader to rewrite the code into a more efficient one.*

All assignment statements utilize the previous formulas that were written for specific models.
Table of type $x@f$ allows us to make various comparisons for different optimal solutions. We have to mention that because of the table shape, we present $E[\zeta^{\odot}]$ instead of $\zeta^{\odot}$ in the first column. It has to be emphasized now that it would not be a good idea to state also any general conclusions about comparisons of reformulations based on one set of instances!

```
        WS     IS     EV     EO     VO      MM
WS      0.202  0.000  0.000  0.202   0.526   2.855
IS      5.052  0.000  0.871  5.052  28.718  12.531
EV      5.762  8.414  0.000  5.762  11.921  15.050
EO      4.625  1.452  0.443  4.625  11.590   9.093
VO      5.559  6.618  1.383  5.559   3.407   8.837
MM      5.202  4.114  1.238  5.202   5.330   7.764
```

Exercise 36 *Interpret the diagonal elements of the table.*

Exercise 37 *Discuss table components separately by rows and columns.*

Exercise 38 *Interpret different table elements. Especially: Why WS@MM = 2.855? Where to find EEV? How to compute EVPI and VSS? Why VO@VO > WS@VO?*

Exercise 39 *Try to run the IS program for other scenarios. Compare and discuss various results.*

Exercise 40 *In the case that you are restricted only to the IS case, which scenario you suggest to use?*

Exercise 41 *Use the source code* `SProg.GMS` *and modify the problem size by enlarging the number of variables, primary random parameters, and scenarios.*

Exercise 42 *Modify the GAMS code to model the utility objective function based on the convex combination (weighted average) of EO and VO objective functions. Study the influence of the weights.*

Exercise 43 *Make a copy of the GAMS program and introduce another objective function. Test the code also for the fixed constraints.*

Remark 67 (Conclusions) We have shown that immediately after introduction of main ideas on the elementary example, we may implement discussed reformulations in the optimization modelling language GAMS[16]. We have seen that we may easily enlarge the number of decision variables n and the number of primary random components K. It also does not matter whether the optimal solution was obtained in the closed form or by a numerical algorithm (as with GAMS solvers).

However, the implementation shows that till now we are limited mainly to the case of *known finite discrete probability distribution*. In addition, we have only discussed the *unconstrained optimization case*. We definitely utilized the advantages of the existence of *closed algebraic formulas* for the objective function with respect to $\mathbf{x}$, $\mathbf{D}(\boldsymbol{\xi})$, and

[16]The choice of modelling language is not significantly important. Check on www sites that you may use, e.g. AMPL, AIMMS, or XPRESS instead of GAMS.

$\mathbf{c}(\boldsymbol{\xi})$. The fact that there are closed algebraic formulas describing the dependence of derived random parameters $D(\boldsymbol{\xi})$ and $\mathbf{c}(\boldsymbol{\xi})$ on $\boldsymbol{\xi}$ is also important.

Before we continue the discussion on three mentioned cases (continuous probability distributions, constraints, and general functions describing dependencies), in the next section, we still have to ask questions about possible complications, as about $\boldsymbol{\xi}(\mathbf{x})$, $P \in \Pi$, and probability and quantile minimization.

LECTURE 7: PO AND QO RISK AVERSE

2.4 Advanced cases

Remark 68 (Minimizing probability) We have discussed different deterministic reformulations. In applications, you may find various requirements, e.g., to increase reliability of some equipment. Therefore, we discuss how to optimize probability that is how to minimize probability of high costs and maximize probability of low costs.

Definition 16 (PO deterministic reformulation) Let the UP (see Definition 6 for details) be given:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x}, \boldsymbol{\xi}) \mid \mathbf{x} \in \mathbb{R}^n\},$$

We define its here-and-now *probabilistic objective (PO) deterministic reformulation* (PO program):

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{P(f(\mathbf{x}, \boldsymbol{\xi}) > b) \mid \mathbf{x} \in \mathbb{R}^n\}, \tag{43}$$

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

We denote the minimal objective function value as $z_{\min}^{\mathrm{PO}}$ and the minimum as $\mathbf{x}_{\min}^{\mathrm{PO}}$. As before, for comparisons, we introduce ζ^{PO} i.e. $\zeta^{\mathrm{PO}} = f(\mathbf{x}_{\min}^{\mathrm{PO}}, \boldsymbol{\xi})$. Notice please that the PO program (43) is the unconstrained one.

Exercise 44 *Find different practical applications that may require minimization of probability of the high costs.*

Exercise 45 *Draw figures to interpret the PO objective function and its values graphically.*

Exercise 46 *For the chosen PO objective function, given bound b, and fixed value $\mathbf{x}$ interpret (even graphically) the objective function value.*

Theorem 2 (PO discrete case) We have the PO program with a discrete finite probability distribution of $\boldsymbol{\xi}$:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{P(f(\mathbf{x}, \boldsymbol{\xi}) > b) \mid \mathbf{x} \in \mathbb{R}^n\}, \tag{44}$$

and $\forall \mathbf{x} \in \mathbb{R}^n$ we denote $P(f(\mathbf{x}, \boldsymbol{\xi}) = f(\mathbf{x}, \boldsymbol{\xi}^s)) = p(\boldsymbol{\xi}^s) = p_s$. We further assume that $\forall \mathbf{x} \in \mathbb{R}^n,\ \forall \boldsymbol{\xi}^s \in \Xi : f(\mathbf{x}, \boldsymbol{\xi}^s) - b$ is bounded from above by M.

Then, the following 0-1 mathematical program solves the PO program:

$$\begin{aligned} ? \in \operatorname*{argmin}_{z,\mathbf{x}} \{ z \mid \forall s \in \mathcal{S} : \delta_s = 1 \Rightarrow\ f(\mathbf{x}, \boldsymbol{\xi}^s) \le b, \\ \sum_{s \in \mathcal{S}} p_s(1-\delta_s) = z,\ \ \delta_s \in \{0,1\}, s \in \mathcal{S}\}, \end{aligned} \tag{45}$$

where $\mathcal{S}$ is a set of indices of realizations from Ξ. In addition, the constraint:

$$\begin{aligned} \delta_s = 1 \Rightarrow f(\mathbf{x}, \boldsymbol{\xi}^s) \le b \qquad & \text{can be replaced by} \\ f(\mathbf{x}, \boldsymbol{\xi}^s) \le b + M(1-\delta_s). & \end{aligned} \tag{46}$$

Proof: Utilize your knowledge from integer programming about indicator variables to prove Theorem 2. □

Exercise 47 *Review indicator variables from IP&DP.*

Exercise 48 *Why we do not prefer* $\iff$ *instead of* $\Rightarrow$ *in the previous Theorem 2?*

Exercise 49 *Write a simple PO program and derive the related 0-1 mathematical program.*

Exercise 50 *Write the PO program for Example 3 and derive related programs (with implications and with bound M).*

Exercise 51 *Implement the 0-1 mathematical program in the GAMS.*

Corollary 3 (Maximizing probability) For maximization:

$$? \in \operatorname*{argmax}_{\mathbf{x}} \{ P(f(\mathbf{x}, \boldsymbol{\xi}) \le b) \mid \mathbf{x} \in \mathbb{R}^n \}, \tag{47}$$

We write:

$$\begin{aligned} ? \in \operatorname*{argmax}_{z,\mathbf{x}} \{ z \mid \forall s \in \mathcal{S} : \delta_s = 1 \Rightarrow\ f(\mathbf{x}, \boldsymbol{\xi}^s) \le b, \\ \sum_{s \in \mathcal{S}} p_s \delta_s = z,\ \ \delta_s \in \{0,1\}, s \in \mathcal{S}\}, \end{aligned} \tag{48}$$

and we replace the implication constraints:

$$\begin{aligned} \delta_s = 1 \Rightarrow f(\mathbf{x}, \boldsymbol{\xi}^s) \le b \\ f(\mathbf{x}, \boldsymbol{\xi}^s) \le b + M(1-\delta_s) \end{aligned} \tag{49}$$

Proof: Utilize your knowledge from integer programming and utilize previous Theorem 2 to prove the corollary. □

Exercise 52 *Compare various formulations of PO programs that search maxima with those that search minima.*

Remark 69 (Minimizing quantile) The quantile in statistics is usually considered as a more robust characteristic in comparison with the mean. Therefore, it could be interesting to learn what we can expect when we minimize the quantile as a representative bound for the objective function level.

Exercise 53 *Find different practical applications that may require minimization of the quantile value.*

Exercise 54 *Draw figures to interpret the QO objective function and its values graphically.*

Exercise 55 *For the chosen QO objective function, given level α, and fixed value $\mathbf{x}$ interpret (even graphically) the objective function value.*

Definition 17 (QO deterministic reformulation) Let the UP (see Definition 6 for details) be given:

$$? \in \underset{\mathbf{x}}{\operatorname{argmin}}\{f(\mathbf{x},\boldsymbol{\xi}) \mid \mathbf{x} \in \mathbb{R}^n\},$$

We define its here-and-now *quantile objective (QO) deterministic reformulation* (QO program):

$$? \in \underset{\mathbf{x},z}{\operatorname{argmin}}\{z \mid P(f(\mathbf{x},\boldsymbol{\xi}) \le z) \ge \alpha\}, \tag{50}$$

where $\alpha \in [0,1]$ is a certain significance level for the objective function $f(\mathbf{x},\boldsymbol{\xi})$ value that we want to have guaranteed.

We denote the minimal objective function value as $z_{\min}^{\mathrm{QO}}$ and the minimum as $\mathbf{x}_{\min}^{\mathrm{QO}}$. As before, for comparisons, we introduce ζ^{QO} i.e. $\zeta^{\mathrm{QO}} = f(\mathbf{x}_{\min}^{\mathrm{QO}},\boldsymbol{\xi})$. Notice please that the QO program (50) is an unconstrained one.

Theorem 4 (QO discrete case) We have the QO program with a discrete finite probability distribution of $\boldsymbol{\xi}$:

$$? \in \underset{\mathbf{x},z}{\operatorname{argmin}}\{z \mid P(f(\mathbf{x},\boldsymbol{\xi}) \le z) \ge \alpha\}, \tag{51}$$

and $\forall \mathbf{x} \in \mathbb{R}^n$ we denote $P(f(\mathbf{x},\boldsymbol{\xi}) = f(\mathbf{x},\boldsymbol{\xi}^s)) = p(\boldsymbol{\xi}^s) = p_s$. We further assume that $\forall \mathbf{x} \in \mathbb{R}^n,\ \forall \boldsymbol{\xi}^s \in \Xi : f(\mathbf{x},\boldsymbol{\xi}^s) - b$ is bounded from above by M.
Then, the following 0-1 mathematical program solves the QO program:

$$\begin{aligned} ? \in \underset{z,\mathbf{x}}{\operatorname{argmin}}\{z \mid \forall s \in \mathcal{S} : \delta_s = 1 \Rightarrow\ & f(\mathbf{x},\boldsymbol{\xi}^s) \le z, \\ \sum_{s \in \mathcal{S}} p_s \delta_s \ge \alpha,\ & \delta_s \in \{0,1\}, s \in \mathcal{S}\}, \end{aligned} \tag{52}$$

where $\mathcal{S}$ is a set of indices of realizations from Ξ. In addition, the constraint:

$$\begin{aligned} \delta_s = 1 \Rightarrow f(\mathbf{x},\boldsymbol{\xi}^s) \le z \qquad & \text{can be replaced by} \\ f(\mathbf{x},\boldsymbol{\xi}^s) \le z + M(1-\delta_s). & \end{aligned} \tag{53}$$

Proof: Utilize your knowledge from integer programming about indicator variables to prove the theorem. Add the experience with the proof of Theorem 2. □

Exercise 56 *Why we do not prefer $\iff$ instead of $\Rightarrow$ in Theorem 4?*

Exercise 57 *Write a simple QO program and derive the related 0-1 mathematical program.*

Exercise 58 *Write the QO program for Example 3 and derive related programs (with implications and with bound M).*

Exercise 59 *Implement the 0-1 mathematical program in GAMS.*

Exercise 60 *Formulate Theorem 4 for the quantile maximization.*

LECTURE 8: DISTRIBUTION CHANGES

Example 30 (Decision dependent randomness) The interesting decision situation appears when the probability distribution of $\boldsymbol{\xi}$ changes with the change of $\mathbf{x}$. We write $\boldsymbol{\xi}(\mathbf{x})$.

Example 31 (Basic) So, we assume that we have the following underlying program, however, we reduce the feasible set to two values of x and we assume that with the choice of the value of the decision variable x also probability distribution changes:

$$\begin{gathered}\min_x\{(x-\xi(x))^2 \mid x \in \{0,1\}\}, \\ \xi(0) \sim Be(0.5),\ \xi(1) \sim Be(0.6).\end{gathered}$$

We have to choose deterministic reformulation. We cannot choose WS as in the original case, so we assume that the realization of random variable follows the choice of x. Even more, we assume that after the choice of x, the probability distribution $\xi(x)$ is assigned and then realization $\xi^s(x)$ is received. So, we choose the EO reformulation:

$$\begin{gathered}\min_x\{E[(x-\xi(x))^2] \mid x \in \{0,1\}\}, \\ \xi(0) \sim Be(0.4),\ \xi(1) \sim Be(0.6).\end{gathered}$$

As we see, the expected value changes with x twice as it also depends on the change of the probability distribution $\xi(x)$. Because we work with the finite number of the feasible solutions x, we may compute the value of the objective function for each x, and then compare them, and hence, solve the program by a full enumeration:

$$\begin{aligned} x=0: \quad & E[(x-\xi(x))^2] = E[(0-\xi(0))^2] = \\ & \qquad 0.5(0^2)+0.5(1^2) = 0.5, \\ x=1: \quad & E[(x-\xi(x))^2] = E[(1-\xi(1))^2] = \\ & \qquad 0.6(0^2)+0.4(1^2) = 0.4, \end{aligned}$$

and so, we choose $x_{\min} = 1$ because we are minimizing and the related objective function value is smaller.

Exercise 61 *Visualize the steps of the decision making in the previous example by using a so called decision tree*[17]*. Hint: In the case that you are not familiar with this concept look at any OR book for decision theory chapter (Taha, Lieberman-Hillier, Winston, etc.).*

Exercise 62 *Compare concepts introduced in decision theory (EVPI, EVSI, etc.) with the concepts that we have introduced for comparison of stochastic programs.*

Exercise 63 *Write your own GAMS program that solves similar problems of larger size where $n > 1$, $K > 1$ and $|\mathcal{S}| > 2$. Assume that the number of feasible solutions $\mathbf{x}$ is finite and all probability distributions $\boldsymbol{\xi}(\mathbf{x})$ are discrete and finite. Hint: Remember that you do not need to use* `SOLVE` *statement as you only compute maximum of the finite set of values, so* `SMAX` *statement is more suitable.*

[17]Decision trees usually represent simple multistage decision making and their use is a suitable introduction into multistage stochastic programs.

Remark 70 (Continuous x) The principal problems begin when the feasible set contains infinite number $\aleph_1$ of elements, as e.g., $\mathbb{R}$. The full enumeration does not work and discretization of the feasible set to utilize the previous finite discrete case could be a matter in dispute before we know more about the $\boldsymbol{\xi}(\mathbf{x})$ dependence.

Example 32 (Piece-wise constant) The case when the dependence $\boldsymbol{\xi}(\mathbf{x})$ is piece-wise constant, for example:

$$E[f(\mathbf{x},\boldsymbol{\xi}(\mathbf{x}))] = \int_{\Omega} f(\mathbf{x},\boldsymbol{\xi}(\omega,\mathbf{x}))dP(\omega,\mathbf{x}),$$

where $\forall \mathbf{x} \in C_r,\ \forall A \in \mathcal{F}:\ \ P(A,\mathbf{x}) = P_r(A),\ \ \mathcal{R} = \{1,\ldots,R\} \subset \mathbb{N}$ and $\bigcup_{r\in\mathcal{R}} C_r = C$, and $\forall r_1, r_2 \in \mathcal{R}, r_1 \neq r_2 : C_{r_1} \cap C_{r_2} = \emptyset$, can be rewritten as follows:

$$\forall r \in \mathcal{R}, \forall \mathbf{x} \in C_r : E[f(\mathbf{x},\boldsymbol{\xi}(\mathbf{x}))] = \int_{\Omega} f(\mathbf{x},\boldsymbol{\xi}(\omega,\mathbf{x}))dP_r(\omega).$$

Therefore, the minimization problem:

$$\min_{\mathbf{x}}\{E[f(\mathbf{x},\boldsymbol{\xi}(\mathbf{x}))] \mid \mathbf{x} \in C\},$$

can be rewritten as:

$$\min_{r\in\mathcal{R}} \min_{\mathbf{x}}\{E[f(\mathbf{x},\boldsymbol{\xi}_r)] \mid \mathbf{x} \in C_r\},$$

where $\boldsymbol{\xi}_r$ identifies P_r.

The rest of the lectures 8 and 9 will be completed later by hand written notes and pictures from camera.

Remark 71 (General formulas) So for the case of decision dependent randomness $\boldsymbol{\xi}(\mathbf{x})$, we may write different formulas for expectations $E[f(\mathbf{x},\boldsymbol{\xi}(\mathbf{x}))]$. We may review the basic formula that looks like:

Example 33 (Simple, explanatory)

Remark 72 (What can depend on x?) $\Omega_{\mathbf{x}}$, $\mathcal{F}_{\mathbf{x}}$, $\mathcal{P}_{\mathbf{x}}$, and $\boldsymbol{\xi}(\cdot,\bar{\mathbf{x}})$.

Remark 73 (Expressing $\boldsymbol{\xi}(\mathbf{x})$ by using $\mathbf{g}(\mathbf{x},\boldsymbol{\eta})$)

Remark 74 (The case of dependence of distribution parameters on x)

Question 7 (What is known about $\boldsymbol{\xi}$ distribution?)

Example 34 (The unknown distribution case) Two considered probability distributions and the worst case analysis.

Remark 75 (Matrix game theory relation) Pure strategies, saddle point.

Remark 76 (Probabilistic extension)

Remark 77 (Primal and dual LP in use)

LECTURE 9: WAIT-AND-SEE IN STATISTICS

Možné téma únorových přednášek

3 Objective function reformulations - continuous case

3.1 Wait-and-see and statistics

Example 35 (Distribution of the estimate of mean) There are some cases that allow us to identify the probability distribution of the optimal solution and objective function value. Let $\xi \sim \mathcal{N}(\mu, \varsigma^2)$, i.i.d. sample $\boldsymbol{\xi} = (\xi_1, \dots, \xi_N)$, and realizations $\xi_1^s, \dots, \xi_N^s$ be given. We utilize μ parameter estimate by using $\bar{\xi}$ that minimizes a sample variance $\frac{1}{N}\sum_{i=1}^N(\xi_i^s - x)$. Hence, for aforementioned realizations, the optimal solution $x_{\min} = \bar{\boldsymbol{\xi}}^s$ is computed by Calculus. The obtained explicit formula is generalized for $\min_{x(\xi)} \frac{1}{N}\sum_{i=1}^N(\xi_i - x(\boldsymbol{\xi}))$ to obtain probability distribution $x_{\min}(\boldsymbol{\xi}) \sim \mathcal{N}(\mu, \varsigma^2/N)$. Zdrruraznit ovšem, že zde i dále využíváme jednoznačnosti řešení viz dále u WS x hat! Odkaz též na rozdělení účelové funkce po úpravě - chi kvadrát.

Example 36 (Distribution of linear regression parameters) Viz má skripta – odvození formulí a Anděl pro určení rozdělení - znají z regrese také - stačí rychle!

Remark 78 (Extended readings) An excellent discussion is at the beginning of Chapter by Pflug in SProg Handbook.

LECTURE 10: HERE-AND-NOW NEWSBOY

3.2 Here-and-now and utility

Remark 79 (Motivation) This section is about unconstrained HN problems and basic computations for the case of continuous probability distribution. It also represents the bridge to constrained problems.

Example 37 (The newsboy problem with EO and uniform demand) The solution involves, at first, the general derivation and then the use of the probability distribution.

For a newsboy (newsvendor in the USA) problem, we denote: $x \in \mathbb{R}$ a number of newspapers to buy, d is the cost of buying one piece of newspaper. c ($c > d$) is a selling price, ξ is a random demand with a known probability distribution and existing $E\xi$. We maximize the expected profit $E\{z\}$. So:

$$z = f(x, \xi) = \begin{cases} c\xi - dx & \text{for } x \geq \xi \\ cx - dx & \text{for } x < \xi \end{cases}$$

Then:

$$E\{f(x,\xi)\} = \int_{-\infty}^{\infty} f(x,\xi)dF(\xi) =$$

$$\int_{\xi \leq x} (c\xi - dx)dF(\xi) + \int_{\xi > x} (cx - dx)dF(\xi) = (c-d)x - c\int_{\xi \leq x}(x - \xi)dF(\xi).$$

We assume $\xi \in \Xi = [a,b]$ and we get:

$$E\{f(x,\xi)\} = \begin{cases} (c-d)x & \text{for } \ x < a \\ (c-d)x - c\int\limits_a^x (x-\xi)dF(\xi) & \text{for } \ x \in [a,b] \\ -dx + cE\xi & \text{for } \ x > b \end{cases}$$

cf. $(c-d)x - c\int_a^b (x-\xi)dF(\xi) = (c-d)x - cx + cE\xi$. So, $E\{f(x,\xi)\} = (c-d)x - cE_\xi(x-\xi)_+$ where $+$ index means a positive part of the value in parenthesis.
For $\xi \sim U(a,b) : E_\xi\{f(x,\xi)\} = (c-d)x - c\int_a^x (x-\xi)\frac{1}{b-a}d\xi = (c-d)x - \frac{c(x-a)^2}{2(b-a)}$. Then $\max\{E\{f(x,\xi)\} \mid x \geq 0\} = \max\{(c-d)x - cE_\xi(x-\xi)_+ \mid x \geq 0\} = \max\{(c-d)x - \frac{c(x-a)^2}{2(b-a)} \mid x \geq 0\}$. Therefore $x_{\max} = a + \frac{(b-a)(c-d)}{c}$.

Example 38 (The newsboy problem, EO and normal demand) For the newsboy problem, we again denote: $x \in \mathbb{R}$ a number of newspapers to buy, d is the cost of buying one piece of newspaper, c ($c > d$) is a selling price, ξ is now a random demand with a normal probability distribution $\xi \sim N(\mu, \sigma^2)$. We maximize the expected profit $E\{z\}$. So:

$$z = f(x,\xi) = \begin{cases} c\xi - dx & \text{for } \ x \geq \xi \\ cx - dx & \text{for } \ x < \xi \end{cases}$$

Then:

$$E[f(x,\xi)] = \int\limits_{-\infty}^{\infty} f(x,\xi)dF(\xi) =$$

$$\int\limits_{\xi \leq x} (c\xi - dx)dF(\xi) + \int\limits_{\xi > x} (cx - dx)dF(\xi) = (c-d)x - c\int\limits_{\xi \leq x} (x-\xi)dF(\xi).$$

We further assume $\xi \sim N(\mu, \sigma^2)$, we denote a cumulative distribution function of the standard normal probability distribution as $\Phi(\cdot)$ and we obtain:

$$E\{f(x,\xi)\} = (c-d)x - c\int\limits_{-\infty}^{x} (x-\xi)\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(\xi-\mu)^2}{2\sigma^2}}d\xi =$$

$$(c-d)x - c(x-\mu)\Phi(\tfrac{x-\mu}{\sigma}) + \tfrac{c}{\sqrt{2\pi}}\int\limits_{-\infty}^{x} \tfrac{\xi-\mu}{\sigma}e^{-\frac{(\xi-\mu)^2}{2\sigma^2}}d\xi =$$

$$(c-d)x - c(x-\mu)\Phi(\tfrac{x-\mu}{\sigma}) + \tfrac{c\sigma}{\sqrt{2\pi}}\int\limits_{-\infty}^{\frac{x-\mu}{\sigma}} ye^{-\frac{y^2}{2}}dy =$$

$$(c-d)x - c(x-\mu)\Phi(\tfrac{x-\mu}{\sigma}) - \tfrac{c\sigma}{\sqrt{2\pi}}[e^{-\frac{y^2}{2}}]_{-\infty}^{\frac{x-\mu}{\sigma}} =$$

$$(c-d-c\Phi(\tfrac{x-\mu}{\sigma}))x + c\mu\Phi(\tfrac{x-\mu}{\sigma}) - \tfrac{c\sigma}{\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

As the objective function is smooth, we can compute its derivative:

$$0 = [E[f(x,\xi)]]' =$$

$$c-d-c\Phi(\tfrac{x-\mu}{\sigma}) - cx[\Phi(\tfrac{x-\mu}{\sigma})]' + c\mu[\Phi(\tfrac{x-\mu}{\sigma})]' - \tfrac{c\sigma}{\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}(-\tfrac{2(x-\mu)}{2\sigma^2}) =$$

$$c-d-c\Phi(\tfrac{x-\mu}{\sigma}) - \tfrac{cx}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} + \tfrac{c\mu}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} + \tfrac{c}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}(x-\mu) =$$

$$c-d-c\Phi(\tfrac{x-\mu}{\sigma}) \Rightarrow$$

$x_{\max} = \mu + \sigma u_{1-\frac{d}{c}}$. Literature as the Handbook of Stochastic Programming offers less technical, more theoretical and shorter derivation of the result. The handbook copies are available on the request.

Remark 80 (Extended readings) There are more examples, discussions, and generalizations. See Dupačová textbook (in Czech) for figures and the discrete distribution case, SProg Handbook for generalizations and wide use (several chapters), Stochastic Inventory book for further links.

LECTURE 11: HERE-AND-KNOW MARKOWITZ

Example 39 (Utility function evaluation by Freund (Markowitz like)) We have $\mathbf{c}(\boldsymbol{\xi}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $a > 0, a \in \mathbb{R}$. Denote $\mu_x = \mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x}$ and $\sigma_x^2 = \mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x}$. We know that $\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x} \sim N(\mu_x, \sigma_x^2)$. Then

$$
\begin{aligned}
E_\xi\{1-\exp(-a\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x})\} &= \int_{-\infty}^{\infty} (1-\exp(-at))\tfrac{1}{\sqrt{2\pi}\sigma_x}\exp(\tfrac{-(t-\mu_x)^2}{2\sigma_x^2})dt \\
&= 1 - \tfrac{1}{\sqrt{2\pi}\sigma_x}\int_{-\infty}^{\infty} \exp(-at - \tfrac{(t-\mu_x)^2}{2\sigma_x^2})dt \\
&= 1 - \tfrac{1}{\sqrt{2\pi}\sigma_x}\int_{-\infty}^{\infty} \exp(-\tfrac{1}{2\sigma_x^2}(2a\sigma_x^2 t + t^2 - 2t\mu_x + \mu_x^2))dt \\
&= 1 - \tfrac{1}{\sqrt{2\pi}\sigma_x}\int_{-\infty}^{\infty} \exp(-\tfrac{1}{2\sigma_x^2}((t-\mu_x+a\sigma_x^2)^2 - a^2\sigma_x^4 + 2a\mu_x\sigma_x^2))dt \\
&= 1 - \exp(\tfrac{a^2\sigma_x^2}{2} - a\mu_x)\tfrac{1}{\sqrt{2\pi}\sigma_x}\int_{-\infty}^{\infty} \exp(-\tfrac{1}{2\sigma_x^2}(t-\mu_x+a\sigma_x^2)^2)dt \\
&= 1 - \exp(\tfrac{a^2\sigma_x^2}{2} - a\mu_x).
\end{aligned}
$$

So, to maximize the transformed utility function i.e. $\max\{1-\exp(\frac{a^2\sigma_x^2}{2} - a\mu_x)\}$ is equivalent to minimize $\frac{a^2\sigma_x^2}{2} - a\mu_x$ i.e. $\min\{\frac{a}{2}\mathbf{x}^\top\boldsymbol{\Sigma}\mathbf{x} - \boldsymbol{\mu}^\top\mathbf{x}\}$.

This result could be also formulated for the constrained cases, e.g.:

$$
\operatorname*{argmax}_{\mathbf{x}}\{E[1-\exp(-a\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x})] \mid \mathbf{x} \in C\} = \operatorname*{argmin}_{\mathbf{x}}\{\frac{a}{2}\mathbf{x}^\top\boldsymbol{\Sigma}\mathbf{x} - \boldsymbol{\mu}^\top\mathbf{x} \mid \mathbf{x} \in C\}
$$

where $C \subset \mathbb{R}^n$ is the same fixed feasible set for both considered cases.

Remark 81 (Extended readings) Dupačová contains discussion on Freund's example.

LECTURE 12: HN TOWARDS CONSTRAINTS

The notes will be extended here with:

Remark 82 (PO objective for normal distribution) The previous discussed case of $f(\mathbf{x}, \boldsymbol{\xi}) = \mathbf{c}(\boldsymbol{\xi})^\top\mathbf{x}$ and $\mathbf{c}(\boldsymbol{\xi}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The formula for Normal distribution $u = \frac{z-\mu_x}{\sigma_x}$ utilized. The fact that Φ distribution function is increasing, so it has also increasing inverse function is utilized as well. By Dupačová CZ. Draw figures!

Remark 83 (QO objective for normal distribution) The previous discussed case $f(\mathbf{x}, \boldsymbol{\xi}) = \mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x}$ and $\mathbf{c}(\boldsymbol{\xi}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The formula for Normal distribution $u = \frac{z-\mu_x}{\sigma_x}$ utilized. The fact that Φ distribution function is increasing, so it has also increasing inverse function is utilized as well. By Dupačová CZ. Draw figures!

Remark 84 (VaR and CVaR) VaR and CVaR may be introduced by Handbook and Hurt.

Remark 85 (Markowitz and fixed constraint) Mainly a fixed constraint $\mathbf{1}^\top \mathbf{x} = w$ added (plus nonnegativity bound $\mathbf{x} \leq \mathbf{0}$ when short sales are not allowed).

Remark 86 (Markowitz and random constraint) Multicriteria idea that moves $E[\mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x}]$ either in the objective function or among constraints as $E[\mathbf{c}(\boldsymbol{\xi})^\top \mathbf{x}] \geq r$.

Remark 87 (Markowitz efficient frontier) Remarks on Markowitz model taken from the book by Hurt et al. Part I - efficient frontier etc.

LECTURE 13: RANDOM CONSTRAINTS

4 Reformulations of constraints

4.1 From WS and IS to EV and AS

Formulation. Now, a simple example is developed to illustrate our theme. At first, we have to emphasize that only elementary educational stochastic programs may be solved without any computer support. So, we introduce very simple stochastic continuous knapsack problem to present solution principles in the form:

$$\max\{\mathbf{c}^\top \mathbf{x} \mid \mathbf{a}^\top \mathbf{x} \leq b(\xi), \mathbf{x} \in [0; 1]^n\}, \tag{54}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{a} \in \mathbb{R}^n$, and $b(\xi)$ is a random knapsack capacity.

Solution. Because the knapsack is filled with loose materials, the underlying linear program is easily solvable and the optimal policy is given by nonincreasing sequence of fractions $\frac{c_j}{a_j}$.

Program instance. The following program instance is considered:

$$\begin{gathered}\max\{10x_1 + 15x_2 + 20x_3 \mid \\ 5x_1 + 10x_2 + 20x_3 \leq \xi, x_1, x_2, x_3 \in [0; 1]\},\end{gathered}$$

where ξ is a random variable, satisfying: $p_1 = P(\xi = \xi^1) = P(\xi = 3) = 0.2$; $p_2 = P(\xi = \xi^2) = P(\xi = 12) = 0.3$; $p_3 = P(\xi = \xi^3) = P(\xi = 25) = 0.5$. Thereafter, we formulate and solve several deterministic equivalents.

WS model. At first, wait-and-see model (WS) is defined as

$$\max\{\ \mathbf{c}^\top \mathbf{x}(\xi) \mid \mathbf{a}^\top \mathbf{x}(\xi) \le b(\xi), \mathbf{x}(\xi) \in [0;1]^n \text{ a.s.}\},$$

where a.s. is the usual abbreviation for 'almost surely'. Then, the sequence of programs is solved:

$$\begin{gathered}\max\{(10;15;20)\mathbf{x}(\xi^1) \mid \\ (5;10;20)\mathbf{x}(\xi^1) \le 3, \mathbf{x}(\xi^1) \in [0;1]^3\},\end{gathered}$$

$$\begin{gathered}\max\{(10;15;20)\mathbf{x}(\xi^2) \mid \\ (5;10;20)\mathbf{x}(\xi^2) \le 12, \mathbf{x}(\xi^2) \in [0;1]^3\},\end{gathered}$$

$$\begin{gathered}\max\{(10;15;20)\mathbf{x}(\xi^3) \mid \\ (5;10;20)\mathbf{x}(\xi^3) \le 25, \mathbf{x}(\xi^3) \in [0;1]^3\}.\end{gathered}$$

The optimal solutions and the related optimal objective function values are: $\mathbf{x}^{\mathrm{WS}}(\xi^1) = (0.6;0;0)^\top$, $z^{\mathrm{WS}}(\xi^1) = 6$, $\mathbf{x}^{\mathrm{WS}}(\xi^2) = (1;0.7;0)^\top$, $z^{\mathrm{WS}}(\xi^2) = 20.5$, $\mathbf{x}^{\mathrm{WS}}(\xi^3) = (1;1;0.5)^\top$, $z^{\mathrm{WS}}(\xi^3) = 35$.

Expectation of the WS objective. Usually, the expected profit is computed as $z^{\mathrm{WS}} = E_\xi z^{\mathrm{WS}}(\xi) = \sum_{i=1}^3 p_i z^{\mathrm{WS}}(\xi^i) = 24.85$. In general, we may conclude that the repeated solution of deterministic programs and simulation techniques are utilized in the situations when a direct computation of $z^{\mathrm{WS}}(\xi)$ distribution is not possible.

MM (AS - almost surely) model. The pessimistic approach (denoted by MM because of the relation to the Min-Max worst case approach) to randomness leads to

$$\max\{\ \mathbf{c}^\top \mathbf{x} \mid \mathbf{a}^\top \mathbf{x} \le \inf_{\xi \in \Xi} b(\xi), \mathbf{x} \in [0;1]^n\}.$$

Then, the simple program is solved

$$\max\{(10;15;20)\mathbf{x} \mid (5;10;20)\mathbf{x} \le 3, \mathbf{x} \in [0;1]^3\},$$

and the defensive optimal solution and optimal value are obtained $\mathbf{x}^{\mathrm{MM}} = (0.6;0;0)^\top$ $z^{\mathrm{MM}} = 6$.

EV model. The optimistic expected-value (EV) deterministic equivalent has the form

$$\max\{\ \mathbf{c}^\top \mathbf{x} \mid \mathbf{a}^\top \mathbf{x} \le E_\xi b(\xi), \mathbf{x} \in [0;1]^n\},$$

and the linear program is solved:

$$\begin{gathered}\max\{(10;15;20)\mathbf{x} \mid \\ (5;10;20)\mathbf{x} \le 16.7, \mathbf{x} \in [0;1]^3\},\end{gathered}$$

with results $\mathbf{x}^{\mathrm{EV}} = (1;1;0.085)^\top$ and $z^{\mathrm{EV}} = 26.7$. We see that both cases (MM and EV) require the solution of programs similar to the underlying program and the additional difficulties are the identification of the worst scenario and the expected value computation.

LECTURE 14: PROBABILISTIC CONSTRAINTS

4.2 Probabilistic constraints

PC model. The probabilistic constraint (PC) (or SP = separate probabilistic constraint) is considered in

$$\max\{\ \mathbf{c}^\top\mathbf{x} \mid P(\mathbf{a}^\top\mathbf{x} \le b(\xi)) \ge \alpha, \mathbf{x} \in [0;1]^n\}.$$

We utilize a distribution function $F(t) = P(b(\xi) < t)$. Then, we write $P(\mathbf{a}^\top\mathbf{x} \le b(\xi)) = 1 - P(b(\xi) < \mathbf{a}^\top\mathbf{x}) = 1 - F(\mathbf{a}^\top\mathbf{x})$ and $P(\mathbf{a}^\top\mathbf{x} \le b(\xi)) \ge \alpha \Rightarrow 1 - F(\mathbf{a}^\top\mathbf{x}) \ge \alpha$ $\Rightarrow F(\mathbf{a}^\top\mathbf{x}) \le 1 - \alpha$. To simplify this constraint, we want like to utilize inverse of F and apply it on both sides of the inequality. However, in contrast to the case of continuous random variable, F is not a bijective mapping in a general case. So, thinking about F as about the relation, we may obtain the inverse relation F^{-1}. Therefore $F^{-1}(1-\alpha)$ may generate the whole set of satisfactory values t such that $F(t) = 1 - \alpha$ for a finite subset of values from $[0, 1]$. We need just one of them but for any $\alpha \in [0, 1]$. Notice that we need to satisfy $F(t) \le 1 - \alpha$ instead of aforementioned $F(t) = 1 - \alpha$. So, we search for d satisfying $t \le d$ where $d \in F^{-1}(1-\alpha)$. We may take the smallest achievable value of the set $F^{-1}(1-\alpha)$ i.e. its infimum. It simplifies our problem for the case when $F^{-1}(1-\alpha) \ne \emptyset$. For other cases, we again utilize the fact that we are interested in $F(t) \le 1 - \alpha$. We just need to find $t_{1-\alpha}$ such that $F(t_{1-\alpha}) \ge 1 - \alpha$ and for any smaller $t < t_{1-\alpha} : F(t) < 1 - \alpha$. Then we may write $t < t_{1-\alpha}$. As we have chosen the distribution function that satisfies $\lim_{t \longrightarrow t_{0-}} F(t) = F(t_0)$ (and not necessarily $\lim_{t \longrightarrow t_{0+}} F(t) = F(t_0)$), $t_{1-\alpha}$ may not exist as introduced above. Therefore, we may use limit or infimum and with it we obtain $t \le t_{1-\alpha}$. Such a definition precisely defines the quantile function: $F^{-1}(1-\alpha) = \inf\{t \mid F(t) \ge 1 - \alpha\}$. Then we get:

$$\max\{\ \mathbf{c}^\top\mathbf{x} \mid \mathbf{a}^\top\mathbf{x} \le F^{-1}(1-\alpha), \mathbf{x} \in [0;1]^n\}.$$

The example of an explicit deterministic equivalent is derived for $\alpha = 0.7$:

$$\max\{(10; 15; 20)\mathbf{x} \mid (5; 10; 20)\mathbf{x} \le 12, \mathbf{x} \in [0;1]^3\},$$

with the optimal solution $\mathbf{x}^{\mathrm{PC}} = (1; 0.7; 0)^\top$ and optimal objective value $z^{\mathrm{PC}} = 20.5$ (PC abbreviates probabilistic constraints in general, SP denotes special case of separate probabilistic constraints and JP denotes joint probabilistic constraints). Stochastic programs with probabilistic constraints are often difficult to solve because of implicit deterministic equivalents and missing convexity properties.

Example 40 (The newsboy problem with PC and uniform demand) The deterministic reformulation is:

$$\max\{(c-d)x \mid P(\xi \ge x) \ge \alpha, x \ge 0\}, \quad \xi \sim U[a, b].$$

So, $P(\xi \ge x) = 1$ for $x < a$, $P(\xi \ge x) = 0$ for $x > b$, and $P(\xi \ge x) = \frac{b-x}{b-a}$ for $x \in [a, b]$. So, for $x \in [0, b]$ we solve $\max\{(c-d)x \mid (\frac{b-x}{b-a} \ge \alpha) \vee (x \le a)\}$. As for the first constraint $x \le \alpha a + (1-\alpha)b$ then $x_{\max} = \alpha a + (1-\alpha)b$. If $\alpha = \frac{d}{c}$ then the solution is the same as in the previous case.

Remark 88 (Basics of probabilistic constraints – random matrix) Show that even simple linear program with probabilistic constraint nay lead to the situation that the feasible set shows that it is non-convex. Draw figures!

There are different ways how to build an example to show problems with probabilistic constraints. One possibility is to define a linear program with just one constraint $\mathbf{a}^\top(\boldsymbol{\xi})\mathbf{x} \leq b$ and two variables. Then to set up $P(\xi = \xi^1) = 0.6, P(\xi = \xi^2) = 0.4$ and assign $a_1(\xi^1) = 1, a_2(\xi^1) = 3$ and $a_1(\xi^2) = 3, a_2(\xi^2) = 1$ together with $b = 3$ and $\alpha = 0.3$. Then a simple drawing of the feasible set shows that it is non-convex. Although it can be modeled by using indicator variables and ILP, the important problem for probabilistic constrained programs was identified.

LECTURE 15: RECOURSE FORMULATION

4.3 Recourse

RF model. As the next step, we take a recourse action into account. The underlying program involving the recourse function (RF) has two stages:

$$\begin{gathered}\max\{\mathbf{c}^\top\mathbf{x} - q(\xi)y^-(\xi) \mid \\ \mathbf{a}^\top\mathbf{x} + y^+(\xi) - y^-(\xi) = b(\xi), \mathbf{x} \in [0;1]^n\}.\end{gathered}$$

Penalty coefficients are $q(\xi^1) = 2$, $q(\xi^2) = 3$, $q(\xi^3) = 6$. Aforementioned models (WS, MM and EV) enriched with recourse terms give the same results as before. Only the optimal value for the PC model with the recourse term changes to 16.9. For instance, the modified EV model has the following form:

$$\begin{gathered}\max\{\mathbf{c}^\top\mathbf{x} - [E_\xi q(\xi)]y^- \mid \\ \mathbf{a}^\top\mathbf{x} + y^+ - y^- = E_\xi b(\xi), \mathbf{x} \in [0;1]^n\}.\end{gathered}$$

The structure of our here-and-now (HN) (or recourse (RF)) model complicates:

$$\begin{gathered}\max\{\mathbf{c}^\top\mathbf{x} - \textstyle\sum_{i=1}^3 p_i q(\xi^i)y^-(\xi^i) \mid \\ \mathbf{a}^\top\mathbf{x} + y^+(\xi^i) - y^-(\xi^i) = b(\xi^i), \forall i, \mathbf{x} \in [0;1]^n\}.\end{gathered}$$

Then, the first-stage optimal solution is $\mathbf{x} = (1;1;0)^\top$, recourse actions related to scenarios are $y^-(\xi^1) = 12$, $y^-(\xi^2) = 3$, $y^+(\xi^3) = 10$, and the overall objective function value is $z^{\mathrm{RF}} = 17.5$.

Comparisons. At the end of this example, we compare introduced deterministic equivalents. First, we compute the expectation of expected value (EEV) solution

$\mathrm{EEV} = E_\xi\{\mathbf{c}^\top\mathbf{x}^{\mathrm{EV}} - q(\xi)(\mathbf{a}^\top\mathbf{x}^{\mathrm{EV}} - b(\xi))_+\} = 16.99$,

where subscript + denotes that only positive values of given term are considered. Then, we see that the following inequalities are valid in our case (we denote z^{RF} as z^{HN}):

$$z^{\mathrm{MM}} < \mathrm{EEV} < z^{\mathrm{HN}} = z^{\mathrm{RF}} < z^{\mathrm{WS}} < z^{\mathrm{EV}}.$$

In addition expected value of perfect information $EVPI = z^{WS} - z^{HN} = 7.35$, and value of stochastic solution $VSS = z^{HN} - EEV = 0.51$ are calculated by formulas introduced later.

5 Applications

Lectures may be organized by applications not as the text by reformulations.

LECTURE 16: MODELLING APPLICATIONS I

LECTURE 17: MODELLING APPLICATIONS II

LECTURE 18: MODELLING APPLICATIONS III

Remark 89 (The key recommendation) The best training is: Read the following examples and learn about the solution ideas. Then take any LP verbal example from any OR book you have. Enrich the problem specification with random elements and build required programs in the similar way as it is done for the solved examples.

A. Underlying programs

Underlying program building: This is the key introductory step to the first theme. Without this program you cannot get its deterministic equivalents later!

Remark 90 (Important) For underlying programs, I accept both cases when letters denoting variables either are or are not followed by $(\boldsymbol{\xi})$.

Example 41 (General question) Explain the difference between an underlying program in stochastic programming and a common deterministic program in mathematical programming.

Answer: In comparison with the deterministic mathematical program (MP), the underlying program (SP) also involves coefficients dependent on random elements. Equivalent answers are fully acceptable.

Example 42 (Investment (Dupačová)) Build the underlying program for the investment problem:

The investor wishes to raise enough money for her child college education N years from now by investing the amount w into some of I considered investments. Let the future tuition goal be g; exceeding g after N years provides an additional income of $q\%$ of the excess while not meeting the goal would result in borrowing at the rate $r\%$, $r > q$. The investor plans to revise her investment at certain time instances prior to N using additional information that will gradually become available in the future. The time instances (and the corresponding time periods) are indexed by $t = 1$ for the initial decision, by $t = 2, \ldots, T-1$ for the revisions, and by $t = T$ for the horizon N. The main uncertainty is the return $\rho_i(t, \xi)$ on each investment i within each period t, which depends on an underlying random element ξ and is observable at the end of each period. We also introduce the following decision variables: variables describing the amounts of investments are denoted as $x_i(t), i = 1, ..., I, t = 1, ..., T$, variable y^+ denotes exceeding of the goal g after N years, variable y^- denotes not meeting the goal g after N years. The program has to determine the optimal amounts of investments and their revisions maximizing the overall profit z.

The required underlying program has the following form:

$$
\begin{aligned}
? \in \operatorname*{argmax}_{y^+, y^-, x_i(t), i \in \mathcal{I}, t \in \mathcal{T}^-} & \{ q y^+ - r y^- \mid \sum_{i \in \mathcal{I}} x_i(1) \le w, \\
& \sum_{i \in \mathcal{I}} (1 + \rho_i(t, \xi)) x_i(t) - \sum_{i \in \mathcal{I}} x_i(t+1) = 0, \quad t \in \mathcal{T}^=, \\
& \sum_{i \in \mathcal{I}} (1 + \rho_i(T-1, \xi)) x_i(T-1) - y^+ + y^- = g, \\
& y^+, y^-, x_i(t) \ge 0, i \in \mathcal{I}, t \in \mathcal{T}^- \}
\end{aligned}
$$

where $\mathcal{I} = \{1, \ldots, I\}, \mathcal{T} = \{1, \ldots, T\}, \mathcal{T}^- = \mathcal{T} \setminus \{T\}, \mathcal{T}^= = \mathcal{T} \setminus \{T-1, T\}$.

Example 43 (Production (Kall-Wallace)) Build the underlying mathematical program for the following production problem:

From n raw materials indexed by $j = 1, ..., n$, m different goods indexed by $i = 1, ..., m$ may be simultaneously produced. The unit costs of the raw materials are specified by the vector $\mathbf{c}(\boldsymbol{\xi})$, the demands for the products that must be satisfied by the vector $\mathbf{h}(\boldsymbol{\xi})$, and the production capacity, i.e. the maximal total amount of raw materials that can be processed, is denoted by a known scalar b. The output of the product i per unit of the raw material j is denoted as $t_{ij}(\boldsymbol{\xi})$ and is an element of the m by n matrix $\mathbf{T}(\boldsymbol{\xi})$. If the demand is known and it is not satisfied by the company's production then it is possible to buy and sell finished products to fulfill it. Market prices of such finished products are defined by $\mathbf{q}(\boldsymbol{\xi})$ and the amounts bought should be denoted as $\mathbf{y}$. Determine the amounts of raw materials and products that must be bought i.e. components of the vectors $\mathbf{x}$ and $\mathbf{y}$ by minimizing the overall costs z.

The required underlying program has the following form:

$$
\min_{\mathbf{x}, \mathbf{y}} \{ \mathbf{c}^\top(\boldsymbol{\xi}) \mathbf{x} + \mathbf{q}^\top(\boldsymbol{\xi}) \mathbf{y} \mid \mathbf{1}^\top \mathbf{x} \le b, \mathbf{T}(\boldsymbol{\xi}) \mathbf{x} + \mathbf{y} \ge \mathbf{h}(\boldsymbol{\xi}), \mathbf{x} \ge \mathbf{0}, \mathbf{y} \ge \mathbf{0} \}.
$$

Example 44 (Melt Control (Popela)) Build the underlying program for the melt control problem:

The problem is specified as follows: From n raw materials indexed by $j = 1, ..., n$, an alloy composed of m different elements indexed by $i = 1, ..., m$ has to be produced. The unit costs of the raw materials are defined by the vector $\mathbf{c}$ having components $c_j, j = 1, ..., n$. The goal intervals (based on percentages) that must be satisfied by the composition of the output alloy are defined by lower bounds specified by vector $\mathbf{l}$ (with components $l_i, i = 1, ..., m$) and upper bounds specified by vector $\mathbf{u}$ (with components $u_i, i = 1, ..., m$). At least w tons of the hot metal must be produced. The amount of the element i in the unit weight of raw material j is denoted as $a_{ij}(\boldsymbol{\xi}), i = 1, ..., m, j = 1, ..., n$. It is an element of the m by n composition matrix $\mathbf{A}(\boldsymbol{\xi})$. Compositions $a_{ij}(\boldsymbol{\xi})$ are considered as random variables. Because of heating, the amounts of elements in the mixed hot melt may change their weights randomly. For modeling purposes, it is useful to introduce proportions of amounts of elements that are further utilized in the hot melt. We denote them $t_{ii}(\boldsymbol{\xi}), i = 1, ..., m$, for i-th element and call them utilizations. They may be understood as nonzero elements of diagonal matrix $\mathbf{T}(\boldsymbol{\xi})$. We introduce the following variables: variables describing the amounts input raw materials are components $x_j, j = 1, ..., n$, of vector $\mathbf{x}$, variable v denotes the weight of mixture of raw materials (in tons), and vector variable $\mathbf{h}$ (with components $h_i, i = 1, ..., m$) defines the composition of the melt after utilizations $t_{ii}(\boldsymbol{\xi})$ are applied. The program has to determine the amounts of input raw materials i.e. components of the vector x minimizing the overall costs z.

The required underlying program has the following form:

$$\min_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} \mid \mathbf{T}(\boldsymbol{\xi})\mathbf{A}(\boldsymbol{\xi})\mathbf{x} = \mathbf{h},$$
$$\mathbf{1}^\top\mathbf{h} = v, v\mathbf{l} \le \mathbf{h} \le v\mathbf{u}, v \ge w, \mathbf{x} \ge \mathbf{0}\}.$$

Example 45 (Transportation (any OR book with Network Flows)) The problem is specified as follows: From m producers indexed by $i = 1, ..., m$ the products are transported to n markets indexed by $j = 1, ..., n$. Unit random transportation costs $c_{ij}(\boldsymbol{\xi})$ are known for any couple i and j. Random capacities (its probability distributions) are specified for all producers and they are denoted by $a_i(\boldsymbol{\xi}), i = 1, \ldots, m$. Random demands (its probability distributions) are specified for all markets and they are denoted by $b_j(\boldsymbol{\xi}), j = 1, \ldots, n$. The program has to determine the transported amounts x_{ij} by minimizing the overall costs z.

The required underlying program has the following form:

$$\min\{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}(\boldsymbol{\xi})x_{ij} \mid \sum_{j=1}^{n} x_{ij} \le a_i(\boldsymbol{\xi}),\ i = 1, \ldots, m$$
$$\sum_{i=1}^{m} x_{ij} \ge b_j(\boldsymbol{\xi}),\ j = 1, \ldots, n, \quad x_{ij} \ge 0,\ i = 1, \ldots, m, j = 1, \ldots, n\}.$$

Example 46 The following examples are available under request for your further training:

- Detailed discussion about Dupačová's investment problem is available from the book of Dupačová-Hurt-Štěpán and from Birge-Louveaux.
- Kall-Wallace (K-W) production example is discussed in detail on copies that you have pages 4–7.
- Details on Melt Control formulation are available from my paper (could be sent to you if you wish).
- You may also learn more about the stochastic transportation problem in relation to Theme 5.
- The Birge-Louveaux introductory chapter on modelling contains also farmer's example, financial example, etc., on pages 3–42. Use it in a reasonable way.
- A simple production example is given in the book by Urmila Diwekar.
- Several simple examples were prepared by Maarten van der Vlerk.
- Two interesting examples are in the book by Julia Higle and Suvrajeet Sen.
- The knapsack problem – see related `knapsack.pdf` you obtained. Be careful! We use min quite often but the knapsack is formulated for max!

If you want then you may get extra pages with more training examples both solved and unsolved from: Birge-Louveaux, Kall-Wallace, van der Vlerk, Diwekar, Dupačová-Hurt-Štěpán, and other resources.

Exercise 64 *Be able to build an underlying program by its verbal description. Assume that only symbols not numbers are given.*
Hints: Begin with a mathematical (usually linear) program building! Then place the right way ξ or $\boldsymbol{\xi}$! You may use the summation-index notation (as Kall-Wallace do) or matrix notation (I like the second one but it does not matter which you use [however, use it in the correct way :-)!]).

B. Deterministic reformulations

Deterministic reformulations (or equivalents): Consider the following deterministic reformulations. Place $\boldsymbol{\xi}$ correctly! Use all notes and utilize our discussions during tutorials and lectures. There are general comments in my notes and mainly the knapsack problem can be used as an example. See also the copies from Kall-Wallace book.

I. Wait-and-see reformulation

Wait-and-see (WS): See also K-W, pages 8–9 and the figures in K-W. Hint: Add suitably $\mathbf{x}(\boldsymbol{\xi})$.

Example 47 (General question) Discuss and evaluate usefulness of the WS approach for the given application.

Answer: The WS approach is usually meaningless for problems where you cannot wait with the decision for observations. However, it is still applicable, e.g., in statistical applications.

Example 48 (Investment) Define the wait-and-see (denoted WS) deterministic reformulation for the investment problem.

The following WS deterministic reformulation is obtained by introducing dependence of decision variables on random element:

$$
\begin{aligned}
? \in \operatorname*{argmax}_{y^+(\xi), y^-(\xi), x_i(t,\xi), i\in\mathcal{I}, t\in\mathcal{T}^-} & \{qy^+(\xi) - ry^-(\xi) \mid \sum_{i\in\mathcal{I}} x_i(1,\xi) \le w, \\
& \sum_{i\in\mathcal{I}} (1+\rho_i(t,\xi))x_i(t,\xi) - \sum_{i\in\mathcal{I}} x_i(t+1,\xi) = 0, \quad t \in \mathcal{T}^=, \\
& \sum_{i\in\mathcal{I}} (1+\rho_i(T-1,\xi))x_i(T-1,\xi) - y^+(\xi) + y^-(\xi) = g, \\
& y^+(\xi), y^-(\xi), x_i(t,\xi) \ge 0, i \in \mathcal{I}, t \in \mathcal{T}^-\}.
\end{aligned}
$$

All constraints are assumed to be satisfied $\forall \xi \in \Xi$ (the support set) or almost surely (the discussion on the spaces of solution mappings is not required). You may also use notation with ξ^s instead of ξ i.e. $\forall \xi^s \in \Xi$.

Example 49 (Production) Formulate the wait-and-see (denoted WS) deterministic reformulation for the production problem.

The following deterministic equivalent is useful for cases when we may wait with the decision after obtaining the realization of $\boldsymbol{\xi}$:

$$
\begin{aligned}
\min_{\mathbf{x}(\xi), \mathbf{y}(\xi)} & \{\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) + \mathbf{q}^\top(\boldsymbol{\xi})\mathbf{y}(\boldsymbol{\xi}) \mid \mathbf{1}^\top \mathbf{x}(\boldsymbol{\xi}) \le b, \\
& \mathbf{T}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) + \mathbf{y}(\boldsymbol{\xi}) \ge \mathbf{h}(\boldsymbol{\xi}), \mathbf{x}(\boldsymbol{\xi}) \ge \mathbf{0}, \mathbf{y}(\boldsymbol{\xi}) \ge \mathbf{0}, \forall \boldsymbol{\xi} \in \Xi\}.
\end{aligned}
$$

Example 50 (Melt Control) Define the wait-and-see (denoted WS) deterministic reformulation for the melt control problem.

The following WS deterministic reformulation is obtained by introducing dependence of decision variables on random element:

$$\min_{\mathbf{x}}\{\mathbf{c}^\top \mathbf{x}(\boldsymbol{\xi}) \mid \mathbf{T}(\boldsymbol{\xi})\mathbf{A}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) = \mathbf{h}(\boldsymbol{\xi}),$$
$$\mathbf{1}^\top \mathbf{h}(\boldsymbol{\xi}) = v(\boldsymbol{\xi}), v(\boldsymbol{\xi})\mathbf{l} \le \mathbf{h}(\boldsymbol{\xi}) \le v(\boldsymbol{\xi})\mathbf{u}, v(\boldsymbol{\xi}) \ge w, \mathbf{x}(\boldsymbol{\xi}) \ge \mathbf{0}\ a.s.\}.$$

II. Expected value reformulation

Expected-value (EV): See also K-W, page 5. Hint: Replace correctly $\boldsymbol{\xi}$ by $E[\boldsymbol{\xi}]$ (or shortly by $E\boldsymbol{\xi}$).

Example 51 (General question) Discuss and evaluate usefulness of the EV approach.

Answer: The EV HN optimal solution is often too optimistic and does not hedge us against variability of ξ. Therefore, the EV HN approach is not recommended for similar applications in general. For true evaluation of the EV optimal solution, the VSS value has to be computed and discussed. So, it is suitable only for the cases with a low level of the VSS value.

Example 52 (Investment) Define the expected-value (denoted EV) here-and-now deterministic reformulation for the investment problem.

The EV HN reformulation has the following form:

$$? \in \operatorname*{argmax}_{y^+, y^-, x_i(t), i\in\mathcal{I}, t\in\mathcal{T}^-} \{qy^+ - ry^- \mid \sum_{i\in\mathcal{I}} x_i(1) \le w,$$
$$\sum_{i\in\mathcal{I}}(1+\rho_i(t, E\xi))x_i(t) - \sum_{i\in\mathcal{I}} x_i(t+1) = 0, \quad t \in \mathcal{T}^=,$$
$$\sum_{i\in\mathcal{I}}(1+\rho_i(T-1, E\xi))x_i(T-1) - y^+ + y^- = g,$$
$$y^+, y^-, x_i(t) \ge 0, i \in \mathcal{I}, t \in \mathcal{T}^-\}$$

where $\mathcal{I} = \{1,\dots,I\}, \mathcal{T} = \{1,\dots,T\}, \mathcal{T}^- = \mathcal{T} \setminus \{T\}, \mathcal{T}^= = \mathcal{T} \setminus \{T-1, T\}$. Obviously, you can use either $E[\rho_i(t,\xi)]$ or $\rho_i(t, E\xi)$ as we assume that $E[\rho_i(t,\xi)] = \rho_i(t, E\xi)$ (because of linear dependence of $\rho_i(t,\cdot)$ on ξ).

Example 53 (Production) Formulate the expected-value (denoted EV) deterministic reformulation for the production problem.

This deterministic equivalent is used often and wrongly as it does not utilize the available information about $\boldsymbol{\xi}$:

$$\min_{\mathbf{x},\mathbf{y}}\{E[\mathbf{c}^\top(\boldsymbol{\xi})]\mathbf{x} + E[\mathbf{q}^\top(\boldsymbol{\xi})]\mathbf{y} \mid \mathbf{1}^\top\mathbf{x} \le b, E[\mathbf{T}(\boldsymbol{\xi})]\mathbf{x} + \mathbf{y} \ge E[\mathbf{h}(\boldsymbol{\xi})], \mathbf{x} \ge \mathbf{0}, \mathbf{y} \ge \mathbf{0}\}$$

and again we assume, e.g., $E[T(\boldsymbol{\xi})] = T(E\boldsymbol{\xi})$, so you can use slightly different notation.

Example 54 (Melt Control) Define one-stage expected-value (denoted EV) here-and-now deterministic reformulation for the melt control problem.

The EV program has the following form:

$$\min_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} \mid \mathbf{T}(E\boldsymbol{\xi})\mathbf{A}(E\boldsymbol{\xi})\mathbf{x} = \mathbf{h}(E\boldsymbol{\xi}),$$
$$\mathbf{1}^\top\mathbf{h}(E\boldsymbol{\xi}) = v(E\boldsymbol{\xi}), v(E\boldsymbol{\xi})\mathbf{l} \le \mathbf{h}(E\boldsymbol{\xi}) \le v(E\boldsymbol{\xi})\mathbf{u}, v(E\boldsymbol{\xi}) \ge w, \mathbf{x} \ge \mathbf{0}\}.$$

III. Defensive (pessimistic) reformulation

Defensive or min-max MM and permanently feasible approach: See K-W. Hint: Use a.s. or $\forall \boldsymbol{\xi} \in \Xi$ for constraints and inf or sup for the objective.

Example 55 (General) Evaluate usefulness of the approach in applications.

This deterministic equivalent is used to hedge fully against the consequences of randomness.

Example 56 (Production) Formulate the defensive (denoted MM) deterministic reformulation for the production problem.

$$\min_{\mathbf{x},\mathbf{y}} \sup_{\xi\in\Xi} \{\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x} + \mathbf{q}^\top(\boldsymbol{\xi})\mathbf{y} \mid \mathbf{1}^\top\mathbf{x} \le b, \mathbf{T}(\boldsymbol{\xi})\mathbf{x} + \mathbf{y} \ge \mathbf{h}(\boldsymbol{\xi}), \mathbf{x} \ge \mathbf{0}, \mathbf{y} \ge \mathbf{0},\ \forall \boldsymbol{\xi} \in \Xi\}.$$

Exercise 65 *Introduce the similar reformulation for the investment problem.*

Exercise 66 *Introduce the similar reformulation for the melt control problem.*

IV. Probability-based reformulations

Probabilistic constraints (PC) or (SP): See also K-W, page 14. More on probability-based objectives – see notes.

Example 57 (General question) Compare probabilistic constraints with permanently feasible (or almost surely) case.

Answer: AS is obtained as PC for $\alpha = 1$, hence the feasible region for PC is usually larger. The feasible region may be a nonconvex set (see Theme 3).

Example 58 (Melt Control) Define one-stage here-and-now deterministic equivalent for the melt control problem with the joint probabilistic constraint. The probability level to be satisfied is denoted as α.

$$\begin{gathered}\min_{\mathbf{x}} \{\mathbf{c}^\top\mathbf{x} \mid P(\mathbf{T}(\boldsymbol{\xi})\mathbf{A}(\boldsymbol{\xi})\mathbf{x} = \mathbf{h}(\boldsymbol{\xi}), \\ \mathbf{1}^\top\mathbf{h}(\boldsymbol{\xi}) = v(\boldsymbol{\xi}), v(\boldsymbol{\xi})\mathbf{l} \le \mathbf{h}(\boldsymbol{\xi}) \le v(\boldsymbol{\xi})\mathbf{u}, v(\boldsymbol{\xi}) \ge w, \mathbf{x} \ge \mathbf{0}) \ge \alpha\}\end{gathered}$$

where $\alpha \in [0, 1]$.

V. Combined HN reformulations

Combined: See previous exam papers, see also static programs (Theme 4 and notes) for examples, e.g., convex combination of EO- and VO- criteria. It is not a bad idea to be able to combine several approaches, e.g. the convex combination of various objectives plus a.s. and probabilistic constraints.

Example 59 (Combined production) Design another deterministic equivalent of the production underlying program as follows: Define the joint probabilistic constraints and the objective function based on the convex combination involving the expectation and variance of the original objective function. Use symbol λ for a convex combination and α for the constraint-related probability level. Give your short opinion when the equivalent can be applied. Hint: Develop the deterministic equivalent in several simple steps.

This deterministic equivalent is suitable for cases when the solution is repeatedly implemented and variance of the cost must be low, e.g., because of limited financial resources. Also some percentage of infeasible decisions is acceptable:

$$\min_{\mathbf{x},\mathbf{y}}\{\lambda(E_\xi\{\mathbf{c}^\top(\boldsymbol{\xi})\}\mathbf{x}+E_\xi\{\mathbf{q}^\top(\boldsymbol{\xi})\mathbf{y}(\boldsymbol{\xi})\})+(1-\lambda)\mathrm{var}\{\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x}+\mathbf{q}^\top(\boldsymbol{\xi})\mathbf{y}(\boldsymbol{\xi})\} \mid \\ \mathbf{1}^\top\mathbf{x}\le b, \mathbf{x}\ge \mathbf{0}, P(\mathbf{T}(\boldsymbol{\xi})\mathbf{x}+\mathbf{y}(\boldsymbol{\xi})\ge \mathbf{h}(\boldsymbol{\xi}), \mathbf{y}(\boldsymbol{\xi})\ge\mathbf{0})\ge\alpha\},$$

where $\alpha, \lambda \in (0,1)$.

Exercise 67 *Develop similar reformulations for the melt control and investment problems.*

VI. Comparison of HN reformulations

Here-and-now (HN): Any of EV, MM, PC is HN. So, usually it is specified — or ask me. With two-stage HN it is often the same as the RF mentioned below.

Verbal comparison: Be able to compare various deterministic equivalents verbally! Concentrate mainly on their usefulness and assumptions when they are used (e.g., WS approach against HN approach; optimistic EV approach against pessimistic MM approach).

Exercise 68 Compare different HN reformulations for the investment problem, the production example, and the melt control problem.

VII. Recourse function-based reformulation

Recourse function approach (RF): See also K-W 10–11.

Example 60 (Investment) Set $T = 2$ in 1. a., and define the here-and-now deterministic reformulation for the investment problem that satisfies: (1) The expected value of the overall profit z is maximized and (2) all constraints are satisfied almost surely. What is the relation between the defined program and the stochastic linear program with recourse?

The EO HN reformulation has the following form:

$$? \in \operatorname*{argmax}_{x_i(1), i\in\mathcal{I}}\{E[\max_{y^+(\xi),y^-(\xi)}\{qy^+(\xi)-ry^-(\xi) \mid \\ -y^+(\xi)+y^-(\xi)=g-\sum_{i\in\mathcal{I}}(1+\rho_i(1,\xi))x_i(1),\quad y^+(\xi),y^-(\xi)\ge 0,\ a.s.\}] \mid \\ \sum_{i\in\mathcal{I}}x_i(1)\le w,\ x_i(1)\ge 0, i\in\mathcal{I}\}$$

where $\mathcal{I}=\{1,\dots,I\}$. The defined program is a stochastic linear program with recourse. $x_i(1), i\in\mathcal{I}$ are the first stage variables, $y^+(\xi), y^-(\xi)$ are recourse variables and instead of common minimization we deal with maximization.

Example 61 (Production) Create a two-stage deterministic reformulation with recourse of the production underlying program. Minimize the expectation of its objective function. Denote the vector of the second stage variables as $\mathbf{y}(\boldsymbol{\xi})$.

Two-stage with recourse:

$$\min_{\mathbf{x}}\{E_\xi\{\mathbf{c}^\top(\boldsymbol{\xi})\}\mathbf{x} + \mathcal{Q}(\mathbf{x}) \mid \mathbf{1}^\top\mathbf{x} \le b, \mathbf{x} \ge \mathbf{0}\},\ \text{where}$$
$$\mathcal{Q}(\mathbf{x}) = E_\xi\{Q(\mathbf{x},\boldsymbol{\xi})\},\ \text{and}$$
$$Q(\mathbf{x},\boldsymbol{\xi}) = \min_{\mathbf{y}(\xi)}\{\mathbf{q}^\top(\boldsymbol{\xi})\mathbf{y}(\boldsymbol{\xi}) \mid \mathbf{y}(\boldsymbol{\xi}) \ge \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}(\boldsymbol{\xi})\mathbf{x}, \mathbf{y}(\boldsymbol{\xi}) \ge \mathbf{0}, \text{a.s.}\}$$

VIII. Scenario-based reformulation

Scenario-based (SB): See K-W 11 and later. It appears when the distribution of $\boldsymbol{\xi}$ is discrete finite (mainly related to RF but think also about the knapsack problem) or sample is used. It simplifies the notation.

Example 62 (Production) Give also the scenario-based formulation for the production problem under the assumption that the support of $\boldsymbol{\xi}$ is a finite set denoted by Ξ with elements $\boldsymbol{\xi}^s, s = 1, ..., S$ and related probabilities are $P(\boldsymbol{\xi} = \boldsymbol{\xi}^s) = p_s$. Hint: Decide about the form (either decomposed or compact) and use the most suitable for each case.

Two-stage scenario-based:

$$\min_{\mathbf{x},\mathbf{y}(\xi^s)}\{(\sum_{s=1}^{S} p_s\mathbf{c}^\top(\boldsymbol{\xi}^s))\mathbf{x} + \sum_{s=1}^{S} p_s\mathbf{q}^\top(\boldsymbol{\xi}^s)\mathbf{y}(\boldsymbol{\xi}^s) \mid$$
$$\mathbf{1}^\top\mathbf{x} \le b, \mathbf{x} \ge \mathbf{0}, \mathbf{T}(\boldsymbol{\xi}^s)\mathbf{x} + \mathbf{y}(\boldsymbol{\xi}^s) \ge \mathbf{h}(\boldsymbol{\xi}^s), \mathbf{y}(\boldsymbol{\xi}^s) \ge \mathbf{0}, \forall\boldsymbol{\xi}^s \in \Xi\}.$$

Example 63 (Melt control) Define one-stage scenario-based (denoted SB) here-and-now deterministic reformulation for the melt control problem. Guarantee that all constraints are satisfied almost surely. Give the scenario-based formulation under the assumption that the support of $\boldsymbol{\xi}$ is a finite set denoted by Ξ with elements $\boldsymbol{\xi}^s, s = 1, ..., S$ and related probabilities are $P(\boldsymbol{\xi} = \boldsymbol{\xi}^s) = p_s$. Compare it and its usefulness with the EV approach.

$$\min_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} \mid \mathbf{T}(\boldsymbol{\xi}^s)\mathbf{A}(\boldsymbol{\xi}^s)\mathbf{x} = \mathbf{h}(\boldsymbol{\xi}^s),$$
$$\mathbf{1}^\top\mathbf{h}(\boldsymbol{\xi}^s) = v(\boldsymbol{\xi}^s), v(\boldsymbol{\xi}^s)\mathbf{l} \le \mathbf{h}(\boldsymbol{\xi}^s) \le v(\boldsymbol{\xi}^s)\mathbf{u}, v(\boldsymbol{\xi}^s) \ge w, \mathbf{x} \ge \mathbf{0},\ s = 1, \ldots, S\}.$$

It is a defensive approach that guarantees permanently feasible solution (if exists) in comparison with EV.

IX. Multistage and multiperiod scenario-based reformulations

Stages: Almost all the previous deterministic equivalents (mainly WS, EV, MM) can be formulated for one-stage case (just $\mathbf{x}$ variable), for two-stage case (see $\mathbf{x}$ and $\mathbf{y}(\boldsymbol{\xi})$ variables), and even for two-stage multiperiod or multistage cases.

Example 64 (Investment) Set $T = 3$ and define the scenario-based here-and-now deterministic reformulation for the investment problem that maximizes the expected value of the overall profit z. Assume that $\boldsymbol{\xi}$ has a discrete probability distribution with a finite number of realizations that are denoted as $\boldsymbol{\xi}^s, s = 1, ..., S$ and related probabilities are p_s. Compare the following concepts: years, periods, and true decision stages.

The required SB HN reformulation is defined as follows:

$$
\begin{aligned}
? \in \underset{y^+(\xi^s), y^-(\xi^s), x_i(1), x_i(2,\xi^s), i\in\mathcal{I}, s\in\mathcal{S}}{\operatorname{argmax}} \{ \sum_{s\in\mathcal{S}} p_s(qy^+(\xi^s) - ry^-(\xi^s)) \mid \\
\sum_{i\in\mathcal{I}} x_i(1) \le w, \\
\sum_{i\in\mathcal{I}} (1+\rho_i(1,\xi^s))x_i(1) - \sum_{i\in\mathcal{I}} x_i(2,\xi^s) = 0, \quad s \in \mathcal{S}, \\
\sum_{i\in\mathcal{I}} (1+\rho_i(2,\xi^s))x_i(2,\xi^s) - y^+(\xi^s) + y^-(\xi^s) = g, \quad s \in \mathcal{S}, \\
y^+(\xi^s), y^-(\xi^s), x_i(1), x_i(2,\xi^s) \ge 0, i \in \mathcal{I}, s \in \mathcal{S} \}
\end{aligned}
$$

where $\mathcal{I} = \{1, \ldots, I\}, \mathcal{S} = \{1, \ldots, S\}$. The defined SB HN program deals with N years, 3 periods, and 2 decision stages as y variables cannot be considered as decisions, they only measure difference from the goal g.

Important hint: As the first theme is based on the connection of everything to the derivation of the underlying program, be careful about it. In the case of no (or small) success with the underlying program building, add at least some general comments on it and on the selection of deterministic equivalents (using general formulas) to deserve some 'symbolic' (but still non-zero) marks.

Exercise 69 *For the given (developed earlier by the request in the text – one-stage or two-stage) underlying program and random vector $\boldsymbol{\xi}$ with the probability distribution ..., write down the selected (see above) deterministic equivalent. Try to explain when it is useful (considering practical situations)!*

6 WS programs – theory

LECTURE 19: CONVEXITY I

6.1 Convexity

Basic sets: Be able to define (specify by notation) the following concepts and compare them:
So, $\Xi^*(\mathbf{x})$ are the sets of those $\boldsymbol{\xi}$ for which given $\mathbf{x}$ is feasible i.e. $\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \ge \mathbf{0}$: $\Xi^*(\mathbf{x}) = \{\boldsymbol{\xi} \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x} \ge \mathbf{b}(\boldsymbol{\xi}), \boldsymbol{\xi} \in \Xi\}$,
$\Xi^* = \cap_x \Xi^*(\mathbf{x})$ is the set of those $\boldsymbol{\xi}$ for which every $\mathbf{x}$ is feasible,
$C(\boldsymbol{\xi})$ is the set of those $\mathbf{x}$ which are feasible for given $\boldsymbol{\xi}$ i.e. $\forall \boldsymbol{\xi} \in \Xi : \mathbf{C}(\boldsymbol{\xi}) = \{\mathbf{x} \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x} \ge \mathbf{b}(\boldsymbol{\xi}), \mathbf{x} \ge \mathbf{0}\}$, and
$C = \cap_{\xi\in\Xi} C(\boldsymbol{\xi})$ is the set of permanently feasible $\mathbf{x}$.
Define and discuss sets: $\{\boldsymbol{\xi} \mid C(\boldsymbol{\xi}) \ne \emptyset\}$ is **required** and $\{\mathbf{x} \mid \Xi^*(\mathbf{x}) \ne \emptyset\}$.

Example 65 Utilize figures (see our drawings from the lecture and transparency 4) for explanations! For similar figure as you have on page 4A you can use for the oval set $S = \{(x, \xi) \mid x^2 + \xi^2 \le r^2\}$ and the rectangular area $W = [-a, a] \times [-b, b]$. Choosing various values of radius r of circular area S and a and b you are getting different figures like with 4A.

Theorem about convexity of sets I: On pages (transparencies) 5 and 5A we have the first (state and prove!) "Theorem about convexity of sets $\Xi^*(\mathbf{x}), C(\boldsymbol{\xi})$ and their intersections".

Theorem: We have the WS SLP: $\min\{\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) \geq \mathbf{b}(\boldsymbol{\xi}), \mathbf{x}(\boldsymbol{\xi}) \geq \mathbf{0}, \boldsymbol{\xi} \in \Xi\}$ where $\boldsymbol{\xi}$ is random vector with the support Ξ. We assume that $\mathbf{A}(\boldsymbol{\xi}), \mathbf{c}(\boldsymbol{\xi}), \mathbf{b}(\xi)$ are linear in $\boldsymbol{\xi}$ (see proof for the notation but you may put it also here) and we assume that $\Xi \subset \mathbb{R}^k$ is a convex set. Then:

(1) $\forall \mathbf{x} \in \mathbb{R}^n : \Xi^*(\mathbf{x})$ is a convex set (and even polyhedral if Ξ is polyhedral).

(2) Ξ^* is a convex set.

(3) $\forall \boldsymbol{\xi} \in \Xi : C(\boldsymbol{\xi})$ is a convex (and polyhedral) set.

(4) $C = \cap_{\xi \in \Xi} C(\boldsymbol{\xi})$ is a convex set.

Proof: (1) For given x and $\Xi^*(\mathbf{x}) = \{\boldsymbol{\xi} \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x} \geq \mathbf{b}(\boldsymbol{\xi}), \boldsymbol{\xi} \in \Xi\}$ and $\mathbf{A}(\boldsymbol{\xi}) = \mathbf{A}_0 + \sum_{i=1}^k \mathbf{A}_i \xi_i$, $\mathbf{c}(\boldsymbol{\xi}) = \mathbf{c}_0 + \sum_{i=1}^k \mathbf{c}_i \xi_i$, and $\mathbf{b}(\boldsymbol{\xi}) = \mathbf{b}_0 + \sum_{i=1}^k \mathbf{b}_i \xi_i$, we have $\Xi^*(\mathbf{x})$ specified by linear constraints an Ξ convex set. Then it is an intersection of closed halfspaces and hyperplanes (so convex sets) and Ξ convex. Because the intersection of convex sets is convex then $\Xi^*(\mathbf{x})$ is convex.

(2) Ξ^* is defined as the intersection of convex sets by formula $\Xi^* = \cap_x \Xi^*(\mathbf{x})$, and so, is convex.

(3) $C(\boldsymbol{\xi})$ is a convex and polyhedral set for given $\boldsymbol{\xi}$ because it is defined as the intersection of a finite number of closed halfspaces (cf. $\mathbf{C}(\boldsymbol{\xi}) = \{\mathbf{x} \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x} \geq \mathbf{b}(\boldsymbol{\xi}), \mathbf{x} \geq \mathbf{0}\}$).

(4) Again as in (2) the intersection of convex sets $C = \cap_{\xi \in \Xi} C(\boldsymbol{\xi})$ is a convex set.

Theorem about convexity of sets II: The next theorem (state and prove) is "Theorem about convexity of: (1) the set of $\boldsymbol{\xi}$ for which $C(\boldsymbol{\xi})$ is not empty and (2) the set of $\mathbf{x}$ for which $\Xi^*(\mathbf{x})$ is not empty", see page 6 and 6A.

Theorem: We assume that only $\mathbf{b}(\boldsymbol{\xi})$ is a random vector. So, we have $\min_x\{\mathbf{c}^\top\mathbf{x}(\boldsymbol{\xi}) \mid \mathbf{A}\mathbf{x}(\boldsymbol{\xi}) \geq \mathbf{b}(\boldsymbol{\xi}), \mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi} \in \Xi\}$. We assume that $\mathbf{b}(\boldsymbol{\xi})$ is linear in $\boldsymbol{\xi} \in \Xi$. We assume that $\Xi \subset \mathbb{R}^k$ is a convex set. Then:

(1) The set of those $\boldsymbol{\xi}$ for which $C(\boldsymbol{\xi}) \neq \emptyset$ is a convex set i.e. $\{\boldsymbol{\xi} \in \Xi \mid C(\boldsymbol{\xi}) \neq \emptyset\}$ is convex.

(2) The set of those $\mathbf{x} \geq \mathbf{0}$ for which $\Xi^*(\mathbf{x}) \neq \emptyset$ is a convex set i.e. $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \geq \mathbf{0}, \Xi^*(\mathbf{x}) \neq \emptyset\}$ is convex.

Proof: (1) Prove $\{\boldsymbol{\xi} \in \Xi \mid C(\boldsymbol{\xi}) \neq \emptyset\}$ is convex. We take $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \Xi$ such that $C(\boldsymbol{\xi}_1) \neq \emptyset$ and $C(\boldsymbol{\xi}_2) \neq \emptyset$. We want to show that for $\boldsymbol{\xi} = \lambda\boldsymbol{\xi}_1 + (1-\lambda)\boldsymbol{\xi}_2, \lambda \in [0,1]$ also $C(\boldsymbol{\xi}) \neq \emptyset$ to prove convexity. So, for $\boldsymbol{\xi}_1$ we get any $\mathbf{x}_1 \geq \mathbf{0}$ satisfying $\mathbf{A}\mathbf{x}_1 \geq \mathbf{b}(\boldsymbol{\xi}_1)$ and any $\mathbf{x}_2 \geq \mathbf{0}$

satisfying $\mathbf{A}\mathbf{x}_2 \geq \mathbf{b}(\boldsymbol{\xi}_2)$. They exist by the assumption. Hence, for $\mathbf{x} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ we obtain by summation of previous two inequalities and linear dependence of $\mathbf{b}(\boldsymbol{\xi})$ on $\boldsymbol{\xi}$: $\mathbf{A}\mathbf{x} \geq \lambda\mathbf{b}(\boldsymbol{\xi}_1) + (1-\lambda)\mathbf{b}(\boldsymbol{\xi}_2) = \mathbf{b}(\lambda\boldsymbol{\xi}_1 + (1-\lambda)\boldsymbol{\xi}_2)$. So $C(\boldsymbol{\xi}) \neq \emptyset$ and the proof is complete.

(2) Prove $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \geq \mathbf{0}, \Xi^*(\mathbf{x}) \neq \emptyset\}$ is convex when only $\mathbf{b}(\boldsymbol{\xi})$ is a random vector. Let $\mathbf{x}_1 \geq \mathbf{0}$ is such that $\mathbf{A}\mathbf{x}_1 \geq \mathbf{b}(\boldsymbol{\xi}_1)$, so satisfied for $\boldsymbol{\xi}_1 \in \Xi$. Similarly, $\mathbf{x}_2 \geq \mathbf{0}$ is such that $\mathbf{A}\mathbf{x}_2 \geq \mathbf{b}(\boldsymbol{\xi}_2)$, so satisfied for $\boldsymbol{\xi}_2 \in \Xi$. Then, $\mathbf{A}(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2)\mathbf{b}(\boldsymbol{\xi})$ has a solution $\boldsymbol{\xi}$. We assign $\boldsymbol{\xi} = \lambda\boldsymbol{\xi}_1 + (1-\lambda)\boldsymbol{\xi}_2$. Then $\boldsymbol{\xi} \in \Xi$ because Ξ is convex. Because of linearity of $\mathbf{b}(\boldsymbol{\xi})$ in $\boldsymbol{\xi}$ we may utilize $\lambda\mathbf{b}(\boldsymbol{\xi}_1) + (1-\lambda)\mathbf{b}(\boldsymbol{\xi}_2) = \mathbf{b}(\lambda\boldsymbol{\xi}_1 + (1-\lambda)\boldsymbol{\xi}_2)$ and we get the solution by summation of the original two inequalities.

LECTURE 20: CONVEXITY II

Set of minimizers: Define $X_{\min}(\boldsymbol{\xi})$ and understand it (see page 7, draw figures).

$X_{\min}(\boldsymbol{\xi}) = \mathrm{argmin}_{\mathbf{x}(\xi)}\{\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) = \mathbf{b}(\boldsymbol{\xi}), \mathbf{x}(\boldsymbol{\xi}) \geq \mathbf{0}\}$.

It is obviously convex under linearity assumptions. Its convexity comes from LP.

Example 66 Suggest own definition of the concept "permanently optimal solution" for the WS problem. Hint: Check your notes from the lectures! Why they might be important?

We denote $\forall\boldsymbol{\xi}^s \in \Xi : X_{\min}(\boldsymbol{\xi}^s) = \mathrm{argmin}_{\mathbf{x}(\xi^s)}\{\mathbf{c}^\top(\boldsymbol{\xi}^s)\mathbf{x}^\top(\boldsymbol{\xi}^s) \mid \mathbf{A}(\boldsymbol{\xi}^s)\mathbf{x}(\boldsymbol{\xi}^s) = \mathbf{b}(\boldsymbol{\xi}^s), \mathbf{x}(\boldsymbol{\xi}^s) \geq \mathbf{0}\}$. As it is a linear program solved for the fixed value of $\boldsymbol{\xi}$ then $X_{\min}(\boldsymbol{\xi})$ is a polyhedral convex set $\forall\boldsymbol{\xi} \in \Xi$. Then the set permanently optimal solutions is the intersection $\cap_{\boldsymbol{\xi}^s \in \Xi} X_{\min}(\boldsymbol{\xi}^s)$. If it is a nonempty set then it is a convex set not necessarily polyhedral. As some WS optimal solutions from the intersection are the same for any value $\boldsymbol{\xi}$ they are HN optimal solutions as well.

Theorem about convexity III: The third "Theorem about convexity of the set of suitable $\boldsymbol{\xi}$ (giving $X_{\min}(\boldsymbol{\xi}) \neq \emptyset$)."

Theorem: (Copy the assumptions about WS SLP and linearity of coefficients with respect to $\boldsymbol{\xi}$ from previous theorems.) If only $\mathbf{c}(\boldsymbol{\xi})$ is stochastic then the set of those $\boldsymbol{\xi}$ for which $X_{\min}(\boldsymbol{\xi}) \neq \emptyset$ is convex.

Proof: If $\mathbf{c}^\top(\boldsymbol{\xi}_1)\mathbf{x}_1 \leq \mathbf{c}^\top(\boldsymbol{\xi}_1)\mathbf{x}$ $\forall\mathbf{x}$ feasible and also $\mathbf{c}^\top(\boldsymbol{\xi}_2)\mathbf{x}_2 \leq \mathbf{c}^\top(\boldsymbol{\xi}_2)\mathbf{x}$ $\forall\mathbf{x}$ feasible then $\forall\lambda \in [0,1] : \mathbf{c}(\lambda\boldsymbol{\xi}_1 + (1-\lambda)\boldsymbol{\xi}_2)^\top\mathbf{x} =$ (from linearity) $\lambda\mathbf{c}(\boldsymbol{\xi}_1)^\top\mathbf{x} + (1-\lambda)\mathbf{c}(\boldsymbol{\xi}_2)^\top\mathbf{x} \geq$ (from optimality) $\lambda\mathbf{c}^\top(\boldsymbol{\xi}_1)\mathbf{x}_1 + (1-\lambda)\mathbf{c}^\top(\boldsymbol{\xi}_2)\mathbf{x}_2$. Hence, the finite minimum exists (continuity, LP, bounded from below: also Weierstrass' Theorem :-)) also for the convex combination $\boldsymbol{\xi} = \lambda\boldsymbol{\xi}_1 + (1-\lambda)\boldsymbol{\xi}_2$ introduced above, and so $X_{\min}(\boldsymbol{\xi}) \neq \emptyset$ and proof is complete (as $\boldsymbol{\xi}$ belongs to the required set).

Theorem about convexity/concavity of z: State and prove the "Theorem about convexity about the convexity/concavity of the optimal objective function value of the WS deterministic equivalent", see pages 12–13. For clarity, I have replaced $\mathbf{b}$ by $\boldsymbol{\eta}$ but you may use any of these symbols.

Theorem: We have an underlying program: $z_{\min} = \min\{f(\mathbf{x}, \boldsymbol{\xi}) \mid \mathbf{g}(\mathbf{x}) \geq \boldsymbol{\eta}, \mathbf{x} \geq \mathbf{0}\}$. We assume that $\Xi \subset \mathbb{R}^m$ is a closed convex set. There are $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ random vectors with the support Ξ. Let $\mathbf{g} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ be continuous and concave in $\mathbf{x}$ in the interior of the feasible set and $f : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}$ be concave in $\boldsymbol{\xi} \in \Xi$ for each feasible $\mathbf{x}$ and convex in $\mathbf{x}$. Then:

(1) $z_{\min}$ is a convex function of $\boldsymbol{\eta}$.

(2) $z_{\min}$ is a concave and continuous function of $\boldsymbol{\eta}$ in the interior of Ξ.

Proof: (1) We assume that $\boldsymbol{\xi}$ is fixed. So, $z_{\min}(\boldsymbol{\eta}) = \min\{f(\mathbf{x}(\boldsymbol{\eta}), \boldsymbol{\xi}) \mid \mathbf{g}(\mathbf{x}(\boldsymbol{\eta})) \geq \boldsymbol{\eta}, \mathbf{x}(\boldsymbol{\eta}) \geq \mathbf{0}\}$. We take $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2 \in \Xi$. We get $z_{\min}(\boldsymbol{\eta}_1)$ and $z_{\min}(\boldsymbol{\eta}_2)$. Let $\boldsymbol{\eta} = \lambda\boldsymbol{\eta}_1 + (1-\lambda)\boldsymbol{\eta}_2$ for $\lambda \in [0,1]$ and minimum be obtained at $\bar{\mathbf{x}}$. We take $\mathbf{x} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$. Remember that $f(\mathbf{x}) \geq f(\bar{\mathbf{x}})$ We know from concavity $\mathbf{g}(\mathbf{x}) = \mathbf{g}(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \geq \lambda\mathbf{g}(\mathbf{x}_1) + (1-\lambda)\mathbf{g}(\mathbf{x}_2) \geq \lambda\boldsymbol{\eta}_1 + (1-\lambda)\boldsymbol{\eta}_2 = \boldsymbol{\eta}$, so $\mathbf{x}$ is feasible. In addition, from convexity of f at $\mathbf{x}$ $z_{\min}(\lambda\boldsymbol{\eta}_1 + (1-\lambda)\boldsymbol{\eta}_2) = z_{\min}(\boldsymbol{\eta}) = f(\bar{\mathbf{x}}) \leq f(\mathbf{x}) = f(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda)f(\mathbf{x}_2) = \lambda z_{\min}(\boldsymbol{\eta}_1) + (1-\lambda)z_{\min}(\boldsymbol{\eta}_2)$ and proof is complete.

(2) We assume that $\boldsymbol{\eta}$ is fixed. So, $z_{\min}(\boldsymbol{\xi}) = \min\{f(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}) \mid \mathbf{g}(\mathbf{x}(\boldsymbol{\xi})) \geq \boldsymbol{\eta}, \mathbf{x}(\boldsymbol{\xi}) \geq \mathbf{0}\}$. We take $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \Xi$. We get $\mathbf{x}_1$ and $\mathbf{x}_2$ minimizing $f(\mathbf{x}, \boldsymbol{\xi}_1)$ and $f(\mathbf{x}, \boldsymbol{\xi}_2)$ respectively. Then by concavity and min properties $\min f(\mathbf{x}, \lambda\boldsymbol{\xi}_1 + (1-\lambda)\boldsymbol{\xi}_2) \geq \min\{\lambda f(\mathbf{x}, \boldsymbol{\xi}_1) + (1-\lambda)f(\mathbf{x}, \boldsymbol{\xi}_2)\} \geq \min\{\lambda f(\mathbf{x}, \boldsymbol{\xi}_1)\} + \min\{(1-\lambda)f(\mathbf{x}, \boldsymbol{\xi}_2)\} \geq \lambda \min\{f(\mathbf{x}, \boldsymbol{\xi}_1)\} + (1-\lambda)\min\{f(\mathbf{x}, \boldsymbol{\xi}_2)\}$. Thus $\min f(\mathbf{x}, \boldsymbol{\xi})$ is a concave function of $\boldsymbol{\xi}$ in Ξ, and hence, continuous of $\boldsymbol{\xi}$ in the interior of Ξ.

Hints: All exercises and examples can be derived from the previous list ("define and explain", "state and prove", and "explain counterexample"). Be careful about mixing different theorems and proofs! Remember also definitions of the convex set and the convex function!

LECTURE 21: MEASURABILITY I

6.2 Measurability and counterexamples

Introduction: It begins on page/transparency 13. Schemes and concepts on pages 13–14 are interesting only for the understanding of the notation used later.

There is the key idea to understand measurability as the basic feature of random variables. Be able to explain this concept for the probability space $(\Omega, \mathcal{S}, P)$ and variable ξ using a figure (see you notes form the lectures).

There are two important theorems on page 15 (the proof on pages 24–26 and the supporting selection of results from the measure theory is on pages 22–23): the sixth (state and give the main ideas of the proof – try to find them yourself) "Theorem about measurability of the optimal objective function value":

Theorem – measurability z: For WS stochastic linear program $\inf\{\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) = \mathbf{b}(\boldsymbol{\xi}), \mathbf{x}(\boldsymbol{\xi}) \geq \mathbf{0}\ a.s.\}$, $z_{\inf}(\boldsymbol{\xi})$ is a Borel measurable extended value function (Reminder: We assume that all symbols involving $\boldsymbol{\xi}$ describe random variables as we have initially agreed).

Proof (Main steps):

1. Countable dense set $\mathcal{D}$ of vectors $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{x}_i \geq \mathbf{0}$ is taken.

2. $\forall i, k \in \mathbb{N}$: we define $z_{ik}(\boldsymbol{\xi}) = \mathbf{c}^\top \mathbf{x}_i$ if $\mathbf{A}(\boldsymbol{\xi}), \mathbf{b}(\boldsymbol{\xi})$ satisfy: $\|\mathbf{A}(\boldsymbol{\xi})\mathbf{x}_i - \mathbf{b}(\xi)\| \leq \frac{1}{k}\|$. Otherwise $z_{ik}(\boldsymbol{\xi}) = \infty$.

3. We want to show that $\sup_k \inf_i z_{ik}(\boldsymbol{\xi})$ is Borel measurable.

4. Then $z_{ik}(\boldsymbol{\xi})$ is Borel measurable because $\| \cdot \|$ is continuous, and therefore, Borel measurable.

5. Because sup and inf defined over a countable sets of functions are also Borel measurable, the composed function $\sup_k \inf_i z_{ik}(\boldsymbol{\xi})$ is also Borel measurable.

6. Now, we show that $z_{\min}(\boldsymbol{\xi}) = \sup_k \inf_i z_{ik}(\boldsymbol{\xi})$.

7. We begin with $z_{\min}(\boldsymbol{\xi}) \geq \inf_i z_{ik}(\boldsymbol{\xi}), \forall k \in \mathbb{N}$. We have to discuss 3 cases: $z_{\min}(\boldsymbol{\xi}) = \infty$, $z_{\min}(\boldsymbol{\xi}) = -\infty$, and $-\infty < z_{\min}(\boldsymbol{\xi}) < \infty$.

8. Then we show $z_{\min}(\boldsymbol{\xi}) = \sup_k \inf_i z_{ik}(\boldsymbol{\xi})$ again discussing 3 cases: $z_{\min}(\boldsymbol{\xi}) = \infty$, $z_{\min}(\boldsymbol{\xi}) = -\infty$, and $-\infty < z_{\min}(\boldsymbol{\xi}) < \infty$. This step completes the proof.

Example 67 (Counterexample) Review the example from the lecture where the fact that ξ (defined by mapping $\xi : \Omega \longrightarrow \mathbb{R}$) is not measurable (for the given $(\Omega, \mathcal{S}, P)$) does not allow us to derive probability distributions for $z_{\min}(\boldsymbol{\xi})$ and $\mathbf{x}_{\min}(\boldsymbol{\xi})$. Another example:

We have $\min\{\xi x(\xi) \mid -1 \leq x(\xi) \leq 1\}$, $(\Omega, \mathcal{S}, P)$ where $\Omega = \{B, G, R\}$, $\mathcal{S} = \{\{\}, \Omega, \{B\}, \{R, G\}\}$, $P(\{B\} = 0.5)$, and $\xi(B) = -1, \xi(G) = -1, \xi(R) = 1$. Then for $\xi = -1 : x_{\min}(-1) = 1$ and for $\xi = 1, x_{\min}(\xi) = -1$ and $z_{\min} = -1$. However, ξ is not a random variable, as $\xi^{-1}(-1) \notin \mathcal{S}$, and so, we do not know $P(x_{\min}(\xi) = 1)$. We can solve the problem redefining ξ, e.g., $\xi(G) = 1$.

Theorem about measurability of $\hat{\mathbf{x}}(\boldsymbol{\xi})$ State theorem about measurability "Theorem about measurability of the optimal solutions" (page 15 and the proof on pages 29–30 and the supporting selection of results from the measure theory is on pages 22–23:

Theorem: For WS stochastic linear program

$\inf\{\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) = \mathbf{b}(\boldsymbol{\xi}), \mathbf{x}(\boldsymbol{\xi}) \geq \mathbf{0}\ a.s.\}$

there is a Borel measurable transformation $\hat{\mathbf{x}}(\boldsymbol{\xi}) : \mathbb{R}^k \longrightarrow \bar{\mathbb{R}}^n$, which coincides with an optimal solution of the given WS SLP whenever the WS SLP has a solution. In addition $z_{\inf}(\boldsymbol{\xi}) = \mathbf{c}(\boldsymbol{\xi})^\top \hat{\mathbf{x}}(\boldsymbol{\xi})$ a.s.

Proof (Main steps): The main symbols $\mathcal{M}, \mathcal{N}, \mathcal{B}, \mathcal{C}_i, \mathcal{E}_i, \mathcal{D}$ are important as they are further used, so they must be understood!

1. We define 3 sets.

   $\mathcal{M} = \{\boldsymbol{\xi} \mid \text{WS SLP has no solution}\}$,

   $\mathcal{N} = \{\boldsymbol{\xi} \mid \text{WS SLP has a feasible solution and rank of } \mathbf{A}(\boldsymbol{\xi}) \text{ is } r(\mathbf{A}(\boldsymbol{\xi})) < m\}$, and

   $\mathcal{B} = \{\boldsymbol{\xi} \mid \text{WS SLP has a feasible solution and } r(\mathbf{A}) = m\}$.

2. Measurability of sets $\mathcal{M}, \mathcal{N}, \mathcal{B}$ is proven mainly using continuity of suitable transformations.

3. To define $\hat{\mathbf{x}}$ we partition $\mathcal{B}$ in sets belonging to extreme points (identified by basis matrices $\mathbf{B}_i(\boldsymbol{\xi})$) and extreme directions (identified by $\mathbf{B}_i(\boldsymbol{\xi})$ and columns $\mathbf{a}_j(\boldsymbol{\xi})$ of $\mathbf{A}(\boldsymbol{\xi})$) of the feasible set of SLP.
   $\forall i = 1, \ldots, r : \mathcal{C}_i = \{\boldsymbol{\xi} \mid \ |\mathbf{B}_i(\boldsymbol{\xi})| \neq 0, \mathbf{B}_i^{-1}(\boldsymbol{\xi})\mathbf{b}(\boldsymbol{\xi}) \geq \mathbf{0}\}$,
   We consider the set of points that cause descent extreme directions $\mathcal{D} \subset \mathcal{B}$. Further we split the set of these points: $\forall i = 1, \ldots, r, j = 1, \ldots, n : \mathcal{D}_{ij} = \{\boldsymbol{\xi} \mid (\boldsymbol{\xi} \in \mathcal{C}_i) \wedge (c_j - \mathbf{c}_{B_i}^\top(\boldsymbol{\xi})\mathbf{B}_i^{-1}(\boldsymbol{\xi})\mathbf{a}_j(\boldsymbol{\xi}) < 0) \wedge (\mathbf{B}_i^{-1}(\boldsymbol{\xi})\mathbf{a}_j(\boldsymbol{\xi}) \leq \mathbf{0})\} \setminus \cup_{k=1}^{j-1} D_{ik}$, and
   $\mathcal{E}_i = \{\boldsymbol{\xi} \mid (\boldsymbol{\xi} \in \mathcal{C}_i) \wedge (\mathbf{c}^\top(\boldsymbol{\xi}) - \mathbf{c}_{\mathbf{B}_i(\xi)}^\top \mathbf{B}_i^{-1}(\boldsymbol{\xi})\mathbf{A}(\boldsymbol{\xi}) \geq \mathbf{0}^\top)\} \setminus \cup_{k=1}^{i-1}\mathcal{E}_k$.

4. We have $\mathcal{B} = \cup_{i=1}^{r}(\mathcal{E}_i \cup_{j=1}^{n} \mathcal{D}_{ij})$. We then construct $\hat{\mathbf{x}}(\boldsymbol{\xi})$ classifying whether $\boldsymbol{\xi} \in \mathcal{M}$ or $\mathcal{N}$ or $\mathcal{B}$. Then for $\mathcal{B}$ we deal with $\boldsymbol{\xi} \in \mathcal{E}_i$ and $\boldsymbol{\xi} \in \mathcal{D}_{ij}$ separately. For each case, we utilize LP formulas (simplex method) to get $\hat{\mathbf{x}}(\boldsymbol{\xi})$. Because of linearity of all formulas, we get Borel measurable transformation. This step completes the proof.

LECTURE 22: MEASURABILITY II

Exercise 70 *Be able to rebuilt at least partially the famous example with the "rotating constraint" from the lectures. Identify sets $\mathcal{M}, \mathcal{N}, \mathcal{D}, \mathcal{B}, \mathcal{C}_i, \mathcal{E}_i$ for it!*

Example 68 (Counterexample II) It is **very important**. Explain by an example why $\hat{\mathbf{x}}(\boldsymbol{\xi})$ is needed. The key problem is that the WS LP may have for the given realization of $\boldsymbol{\xi}$ more than one optimal solution, so to talk about the mapping representing $\mathbf{x}_{\min}(\boldsymbol{\xi})$ would be meaningless, and so, the talk about probabilities of appearing of different $\mathbf{x}(\boldsymbol{\xi})$ values as optimal solutions (as their sum might be greater than 1! Look at your lecture notes). In addition, to consider $X_{\min}(\boldsymbol{\xi})$ sets as realizations and their sets as events is not a clear way at this level how to do it.

Therefore, for example ? $\in \mathrm{argmin}\{\xi x(\xi) \mid 0 \leq x(\xi) \leq 1\}$ we define $\xi \sim U[-1, 1]$. We may also introduce a slack variable $x_s(\xi) \geq 0 : x(\xi) + x_s(\xi) = 1$. So, we see that for $\xi < 0, x_{\min}(\xi) = 0$, for $\xi > 0, x_{\min}(\xi) = 1$, and for $\xi = 0, x_{\min}(\xi) \in [0, 1]$.

We follow the idea of the proof: $\mathcal{M} = \{\xi \mid \text{WS SLP has no solution}\} = \emptyset$ in our case,

$\mathcal{N} = \{\xi \mid$ WS SLP has a feasible solution and rank of $\mathbf{A}(\xi)$ is not full
i.e. $r(\mathbf{A}(\xi)) < m\} = \emptyset$ in our case.

$\mathcal{B} = \{\xi \mid$ WS SLP has a feasible solution and $r(\mathbf{A}(\xi)) = m\} = [-1, 1]$. As there are no extreme directions then $\mathcal{D} = \emptyset$. To define $\hat{x}$, we partition $\mathcal{B}$ in sets belonging to extreme points (identified by basis matrices $\mathbf{B}_i(\xi)$. $\forall i = 1, \ldots, r : \mathcal{C}_i = \{\xi \mid \ |\mathbf{B}_i(\xi)| \neq 0, \mathbf{B}_i^{-1}(\xi)\mathbf{b}(\xi) \geq \mathbf{0}\}$. So in our case we can choose, e.g., $\mathcal{C}_1 = [-1, 1]$ and $\mathcal{C}_2 = [-1, 1]$ as constraint does not depend on random variable.
However, now we deal with optimality and by

$\mathcal{E}_i = \{\xi \mid (\xi \in \mathcal{C}_i) \wedge (\mathbf{c}^\top(\xi) - \mathbf{c}_{\mathbf{B}_i(\xi)}^\top \mathbf{B}_i^{-1}(\xi)\mathbf{A}(\xi) \geq \mathbf{0}^\top)\} \setminus \cup_{k=1}^{i-1}\mathcal{E}_k$ we obtain $\mathcal{E}_1 = [-1, 0]$ and $\mathcal{E}_2 = (0, 1]$.

So, we have $\mathcal{B} = \cup_{i=1}^{r} \mathcal{E}_i$. We then construct $\hat{\mathbf{x}}(\xi)$ and we deal with $\xi \in \mathcal{E}_i$. For each case, we utilize LP formulas (simplex method) to get unique value of $\hat{x}(\xi)$. Because of linearity of all formulas, we get Borel measurable transformation. In our case $\xi \in [-1, 0] \Rightarrow x^*(\xi) = 0$ and $\xi \in (0, 1] \Rightarrow x^*(\xi) = 1$.

and computing distributions

Exercise 71 *Follow the computations from the lectures and compute also the cumulative probability distribution function for $\hat{x}(\xi)$ and $z_{\inf}(\xi)$.*

LECTURE 23: COMPUTING DISTRIBUTION

Remark 91 (Inserted comparison with the parametric case) *Definition of validity region:* Define the region of validity $\Xi_{\min}(\mathbf{x})$ (page 8): $\Xi_{\min}(\mathbf{x}) = \{\boldsymbol{\xi} \mid \mathbf{x} \in X_{min}(\boldsymbol{\xi})\}$.

Theorem about convexity IV (of validity region): State and prove the fourth **omitted** "Theorem about convexity of the validity region."

Theorem: (Copy the assumptions about WS SLP and linearity of coefficients with respect to $\boldsymbol{\xi}$ from previous theorems.) If only $\mathbf{c}(\boldsymbol{\xi})$ is stochastic then the set $\Xi_{\min}(\mathbf{x})$ is convex $\forall \mathbf{x}$.
Proof: If $\mathbf{c}^\top(\boldsymbol{\xi}_1)\mathbf{x}_{\min} \leq \mathbf{c}^\top(\boldsymbol{\xi}_1)\mathbf{x}\ \forall \mathbf{x}$ feasible (i.e. $\boldsymbol{\xi}_1 \in \Xi(\mathbf{x}_{\min})$) and also $\mathbf{c}^\top(\boldsymbol{\xi}_2)\mathbf{x}_{\min} \leq \mathbf{c}^\top(\boldsymbol{\xi}_2)\mathbf{x}\ \forall \mathbf{x}$ feasible (i.e. $\boldsymbol{\xi}_1 \in \Xi(\mathbf{x}_{\min})$, so both LPs have the same solution $\mathbf{x}_{\min}$!) then $\forall \lambda \in [0, 1]$: it is clear that convex combination of previous inequalities gives: $\lambda \mathbf{c}(\boldsymbol{\xi}_1)^\top \mathbf{x}_{\min} + (1 - \lambda)\mathbf{c}(\boldsymbol{\xi}_2)^\top \mathbf{x}_{\min} =$ (because of linearity on both sides) $\mathbf{c}(\lambda \boldsymbol{\xi}_1 + (1 - \lambda)\boldsymbol{\xi}_2)^\top \mathbf{x}_{\min} \leq$ (because of optimality) $\mathbf{c}(\lambda \boldsymbol{\xi}_1 + (1 - \lambda)\boldsymbol{\xi}_2)^\top \mathbf{x}, \forall \mathbf{x}$ and hence $\mathbf{x}_{\min}$ also solves the problem for $\boldsymbol{\xi}$ that also belongs to $\Xi_{\min}(\mathbf{x})$ and proof is complete.
The discussion and counterexample on pages 8–9 related to the previous theorem is really only for the future A+ holders. For decision regions (of feasibility and optimality) on pages 10–12, your reading and basic understanding is useful for previous sets $(\mathcal{C}_i, \mathcal{E}_i)$.

Comparison of the introduced concepts with those sets used in the proof of the last measurability theorem is required. For WS problem, we write as usually:

$\min\{\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) = \mathbf{b}(\boldsymbol{\xi}), \mathbf{x}(\boldsymbol{\xi}) \geq \mathbf{0}\}$. We denote all $m \times m$ submatrices (of order m) of $\mathbf{A}(\boldsymbol{\xi})$ as $\mathbf{B}_l(\boldsymbol{\xi}), l = 1, \ldots, \binom{n}{m}$. We consider only those of them that are nonsingular (i.e. $\exists \mathbf{B}^{-1}(\boldsymbol{\xi})$). As in the LP case, for $\mathbf{B}_l^{-1}(\boldsymbol{\xi})\mathbf{b}(\boldsymbol{\xi}) \geq \mathbf{0}$ (i.e. $\boldsymbol{\xi}$ fixed), we call $\mathbf{B}_l(\boldsymbol{\xi})$ a feasible basis. Then:

(1) $\{\boldsymbol{\xi} \mid \mathbf{B}_l^{-1}(\boldsymbol{\xi})\mathbf{b}(\boldsymbol{\xi}) \geq \mathbf{0}\}$ is a feasible parameter region for $\mathbf{B}_l(\boldsymbol{\xi})$ (They may overlap).

The main feature is that for only $\mathbf{b}(\boldsymbol{\xi})$ stochastic, it is a convex polyhedral set $\forall l : \exists \mathbf{B}_l^{-1}(\boldsymbol{\xi})$.

(2) $\{\boldsymbol{\xi} \mid (\mathbf{B}_l^{-1}(\boldsymbol{\xi})\mathbf{b}(\boldsymbol{\xi}) \geq \mathbf{0}) \wedge (\mathbf{c}_B^\top(\boldsymbol{\xi})\mathbf{B}_l^{-1}(\boldsymbol{\xi})\mathbf{N}_l(\boldsymbol{\xi}) - \mathbf{c}_{N_l}^\top(\boldsymbol{\xi}) \leq \mathbf{0})\}$ is called a decision region. See transparencies for details.

Any simple example for explanations of these two concepts is acceptable, e.g., $\min\{\xi(x_1) \mid x_1 + x_2 = \xi, x_1, x_2 \geq 0\}, \xi \in \mathbb{R}$. This is the end of the remark.

Defect definition: It is said that the WS SLP objective function has no defect if $P(-\infty < z_{\min}(\boldsymbol{\xi}) < \infty) = 1$.

If $P(-\infty < z_{\min}(\boldsymbol{\xi}) < \infty) < 1$ then the objective function has a defect.

Example 69 (Defect) Be able to develop simple examples with and without defect. For example

$\min\{\xi x^2(\xi) \mid x \in \mathbb{R}\}, \xi \sim U[a,b]$.

Then if $[a,b] = [-2,2]$ causes the defect and for $[a,b] = [1,2]$ we get no defect.

Example 70 Define and explain the concept of 'defect' (see 16) and give the example of the SLP without the defect (use definition of PLP on page 16 or transportation problem on page 17). Skip Bereanu's Theorem (page 16–17).

It is said that the WS SLP objective function has a defect if $P(-\infty < z^{\mathrm{WS}}_{\min}(\boldsymbol{\xi}) < \infty) < 1$. Therefore, the probability distribution of $z^{\mathrm{WS}}_{\min}(\boldsymbol{\xi})$ is defined on the extended real line. However, because of the infinite values, usually certain characteristics cannot be computed. Infinite values may appear because of the infeasibility or unboundedness. As the transportation problem is defined for minimization of costs and the objective function coefficients and variables are obviously greater than zero, the only reason for the defect could be infeasibility. This case is obtained when the total supply does not satisfy the total demand (with non-zero probability).

Computing probability distributions – general case: In the context of the main ideas of the proof of the last theorem be able to state and explain the general formula for the distribution function of the WS problem (see page 30 and compare with proof 29–30 and similarities to pages 19–21). It may help you to see the formula working with the results of the example on pages 31–32.

Under the assumption that WS SLP

$$z_{\min}(\boldsymbol{\xi}) = \min_{\mathbf{x}(\boldsymbol{\xi})}\{\mathbf{c}^\top(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) \mid \mathbf{A}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) = \mathbf{b}(\boldsymbol{\xi}), \mathbf{x}(\boldsymbol{\xi}) \geq \mathbf{0}\ a.s.\}$$

has no defect i.e. $P(-\infty < z_{\min}(\boldsymbol{\xi}) < \infty) = 1$ and

$P(\{\boldsymbol{\xi} \mid \text{WS SLP has a feasible solution and } r(\mathbf{A}) < m\}) = 0$ then:

$$F_{z_{\min(\xi)}}(t) = P(z_{\min}(\boldsymbol{\xi}) \leq t) = \sum_{i=1}^{r} P(\boldsymbol{\xi} \in \mathcal{E}_i \wedge \mathbf{c}^\top_{\mathbf{B}_i(\xi)}\mathbf{B}_i^{-1}(\boldsymbol{\xi})\mathbf{b}(\boldsymbol{\xi}) \leq t), \quad \text{where}$$

$$\mathcal{E}_i = \{\boldsymbol{\xi} \mid (\boldsymbol{\xi} \in \mathcal{C}_i) \wedge (\mathbf{c}^\top(\boldsymbol{\xi}) - \mathbf{c}^\top_{\mathbf{B}_i(\xi)}\mathbf{B}_i^{-1}(\boldsymbol{\xi})\mathbf{A}(\boldsymbol{\xi}) \geq \mathbf{0}^\top)\} \qquad \text{and}$$

$$\forall i = 1, \ldots, r : \mathcal{C}_i = \{\boldsymbol{\xi} \mid \ |\mathbf{B}_i(\boldsymbol{\xi})| \neq 0, \mathbf{B}_i^{-1}(\boldsymbol{\xi})\mathbf{b}(\boldsymbol{\xi}) \geq \mathbf{0}\} \setminus \cup_{k=1}^{i-1} C_k.$$

There are all basis matrices of $\mathbf{A}(\boldsymbol{\xi})$ denoted as $\mathbf{B}_i(\boldsymbol{\xi}), i = 1, \ldots, r$.

Skip the rest of pages 32–33.

Computing probability distributions – simpler case: It is required only in the sense that you have to know how to apply it for the following example!

You are asked to compute simple integrals with density $f(t)$ like $P(\xi \in B) = \int_B f(t)dt$ see bottom of 27. Then, to sort results (see the RHS of 27) and at the end to apply theory explained on pages 19–21 (this part of theory is not required for other purposes) to use formula on page 27. Computations of the expectation on page 28 are based on the general formula for computing moments for random variables (but see also 19–21).

So we denote $z_{\inf}(\boldsymbol{\xi})$ the optimal objective function of WS SLP as before. Let $\boldsymbol{\xi}$ be absolutely continuous ($f(\mathbf{t})$ is a density function). All measurability assumptions related to the program's coefficients are satisfied. No defect appears (check by the graph of the parametric description of $z_{\inf}(\boldsymbol{\xi})$) that is usually given. There are $\mathbf{B}_l(\boldsymbol{\xi})$ submatrices (we assume that $\exists\boldsymbol{\xi} : \mathbf{A}(\boldsymbol{\xi})$ has a full rank). So for that $\mathbf{B}_l(\boldsymbol{\xi})$ that are almost nonosingular, we have got:

$$\mathcal{A}_l = \{\boldsymbol{\xi} \mid \boldsymbol{\xi} \in \Xi, (\mathbf{B}_l^{-1}(\boldsymbol{\xi})\mathbf{b}(\boldsymbol{\xi}) \geq \mathbf{0}) \wedge (\mathbf{c}_{B_l}^\top(\boldsymbol{\xi})\mathbf{B}_l^{-1}(\boldsymbol{\xi})\mathbf{N}_l(\boldsymbol{\xi}) - \mathbf{c}_{N_l}^\top(\boldsymbol{\xi}) \leq \mathbf{0}^\top)\}, l = 1, \ldots, q.$$

Then we define:

$$\mathcal{B}_l = \mathcal{A}_l - \cup_{j=1}^{l-1}\mathcal{A}_j.$$

The key formula is:

$$F_{z_{\inf}(\xi)}(v) = \sum_{l=1}^{q}\int_{\mathcal{C}_l(v)} f_\xi(\mathbf{t})d\mathbf{t},$$

where $\mathcal{C}_l(v) = \{\boldsymbol{\xi} \mid \boldsymbol{\xi} \in \mathcal{B}_l \wedge z_l(\boldsymbol{\xi}) = \mathbf{c}_{B_l}^\top(\boldsymbol{\xi})\mathbf{B}_l^{-1}(\boldsymbol{\xi})\mathbf{b}(\boldsymbol{\xi}) < v\}$. For computations, replace $\boldsymbol{\xi}$ by $\mathbf{t}$.

Example 71 (Parametric LP-based example) For Dupačová's example (pages 27 and 28, use also your notes from the lectures and 2003 year exam paper Question 4 b.) repeat computations for different set up.

However assume that, $\mathcal{A}_i$ are given (different intervals), ξ is an variable with the uniform probability distribution, and parametric formulas for $z_{\min}(\xi)$ are also given.

Exercise 72 *Be able to compute very simple examples (n =1, 2, k = 1, 2) to identify probability distributions of $z_{min}(\boldsymbol{\xi})$ and $\mathbf{x}_{min}(\boldsymbol{\xi})$ for unconstrained nonlinear WS programs (linear regression examples, etc.). For example:*

(1) Minimize sum of squares of deviations (to get the mean as the solution).

(2) Minimize maximum absolute deviation, sum of absolute deviations (rewrite as LPs).

(3) Minimize sum of squares for regression function $y = \beta x$.

Exercise 73 *Utilize formulas for transformations of random variables, see your notes from lectures (calculus-like examples, etc.). For example:*

Identify probability distribution for $z_{min}(\boldsymbol{\xi})$ and $\mathbf{x}_{min}(\boldsymbol{\xi})$:

$\min\{x_1^2(\xi) - 2x_1(\xi) + x_2^2(\xi) - 2\xi^2 x_2(\xi) + \xi^4 \mid x_1(\xi), x_2(\xi) \in \mathbb{R}\}$, *where* $\xi \sim U[0,1]$.

Compare it with the last year exam paper Question 5 a.

7 Two-stage programs

LECTURE 24: TWO-STAGE BASICS

Compare with copies from B-L book, pages 83–93.

Two-stage linear programs: Give the specification of general two-stage programs with recourse:
Two-stage program with random recourse:

$$\min_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} + \mathcal{Q}(\mathbf{x}) \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \quad \text{where}$$
$$\mathcal{Q}(\mathbf{x}) = E_\xi\{Q(\mathbf{x}, \boldsymbol{\xi})\}, \quad \text{and}$$
$$Q(\mathbf{x}, \boldsymbol{\xi}) = \min_{\mathbf{y}(\xi)}\{\mathbf{q}^\top(\boldsymbol{\xi})\mathbf{y}(\boldsymbol{\xi}) \mid \mathbf{W}(\boldsymbol{\xi})\mathbf{y}(\boldsymbol{\xi}) = \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}(\boldsymbol{\xi})\mathbf{x}, \mathbf{y}(\boldsymbol{\xi}) \geq \mathbf{0}\ a.s.\}.$$

Also a nested form is a correct answer (develop the SB version alone!):

$$\min_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} + E_\xi\{\min_{\mathbf{y}(\xi)}\{\mathbf{q}^\top(\boldsymbol{\xi})\mathbf{y}(\boldsymbol{\xi}) \mid \mathbf{W}(\boldsymbol{\xi})\mathbf{y}(\boldsymbol{\xi}) = \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}(\boldsymbol{\xi})\mathbf{x}, \mathbf{y}(\boldsymbol{\xi}) \geq \mathbf{0}\ a.s.\}\}$$
$$\mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}.$$

Exercise 74 *Draw various figures illustrating occurrence of randomness in different vectors and matrices of the second stage program. Review figures from the lecture.*

LECTURE 25: SCENARIOS

Scenario-based two-stage stochastic linear program: It (cf. with Investment application). The SB TS SLProg with implicit nonanticipativity constraints can be written as follows:

$$? \in \operatorname*{argmin}_{\mathbf{x},\mathbf{y}_s : s \in \mathcal{S}}\{\mathbf{c}^\top\mathbf{x} + \sum_{s=1}^{S} p_s \mathbf{q}_s^\top \mathbf{y}_s \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{W}_s\mathbf{y}_s = \mathbf{h}_s - \mathbf{T}_s\mathbf{x}, \mathbf{y}_s \geq \mathbf{0}, s \in \mathcal{S}\} \tag{55}$$

This description is more compact than the next one. It is suitable for efficient algorithms, mainly, for Bender's decomposition.

SB TS SLProg – explicit nonanticipativity constraints:

$$? \in \operatorname*{argmin}_{\mathbf{x}_s,\mathbf{y}_s : s \in \mathcal{S}}\{\sum_{s=1}^{S} p_s(\mathbf{c}^\top\mathbf{x}_s + \mathbf{q}_s^\top\mathbf{y}_s) \mid \tag{56}$$
$$\mathbf{A}\mathbf{x}_s = \mathbf{b}, \mathbf{x}_s \geq \mathbf{0}, \mathbf{W}_s\mathbf{y}_s = \mathbf{h}_s - \mathbf{T}_s\mathbf{x}_s, \mathbf{y}_s \geq \mathbf{0}, s \in \mathcal{S}, \forall r, u \in \mathcal{S} : \mathbf{x}_r = \mathbf{x}_u\},$$

The model with explicit nonanticipativity constraints is easy to derive from the deterministic program and it is suitable for decomposition algorithms based on scenarios. Why? Because it si built by using IS reformulations with extra constraints connecting the first stage variables.

LECTURE 26: RECOURSE TYPES

Types of recourse: Two-stage program with fixed recourse: $\mathbf{W}(\boldsymbol{\xi}) = \mathbf{W}$ i.e. the recourse matrix is constant.

Two-stage program with relatively complete recourse: Iff $\mathbf{W}$ is constant and satisfies: $\forall \mathbf{x} \in \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \forall \boldsymbol{\xi} \in \Xi : \exists \mathbf{y}(\boldsymbol{\xi}) \geq \mathbf{0} : \mathbf{W}\mathbf{y}(\boldsymbol{\xi}) = \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}(\boldsymbol{\xi})\mathbf{x}$ (or

$\mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}(\boldsymbol{\xi})\mathbf{x} \in \text{pos}\mathbf{W}$ i.e. the RHS belongs to the positive cone generated by columns of $\mathbf{W}$).

Two-stage program with complete recourse: Iff $\mathbf{W}$ is constant and satisfies: $\forall \mathbf{t} \in \mathbb{R}^{m_2}$: $\exists \mathbf{y} \geq \mathbf{0} : \mathbf{W}\mathbf{y}(\boldsymbol{\xi}) = \mathbf{t}$ i.e. $\forall \mathbf{t} \in \mathbb{R}^{m_2} : \{\mathbf{y} \mid \mathbf{W}\mathbf{y} = \mathbf{t}, \mathbf{y} \geq \mathbf{0}\} \neq \emptyset$.

Two-stage program with simple recourse: Iff $\mathbf{W} = (\mathbf{I}, -\mathbf{I})$. You may also use $\mathbf{y}^+$ and $\mathbf{y}^-$ version.

Two-stage program with separable simple recourse:
Iff $\mathbf{W} = (\mathbf{I}, -\mathbf{I}), \mathbf{q}(\boldsymbol{\xi}) = (\mathbf{q}^+, \mathbf{q}^-)$ and $\mathbf{q}^+ + \mathbf{q}^- \geq \mathbf{0}, \mathbf{T}(\boldsymbol{\xi}) = \mathbf{T}, \mathbf{h}(\boldsymbol{\xi}) = \boldsymbol{\xi}$.

Exercise 75 *Be able to explain these concepts by using "our" trajectory-based figures in the requirement space (see your own lecture notes). You may even do some basic computations to draw trajectories for $2 \times n$ LPs!*

Comparison: Compare these concepts using basic properties (see Birge-Louveaux).

Exercise 76 *Be able to include the recourse action in the knapsack as we did during lectures.*

Exercise 77 *What is the relation between EEV, VSS, EVPI values (their finiteness) and different types recourse?*

LECTURE 27: FEASIBLE SET CONVEXITY

K sets: "Theorem about convexity of K_2^P set" (State and prove), see Theorem 1 in B-L 86–87 and 2003 exam paper Question 6 d. for details.

Define K_2 set and K_2^P sets by B-L. Be able to list all their important properties. Be able to compare them – what is the difference between them? How their properties depend on the recourse type?

LECTURE 28: Q FUNCTIONS CONVEXITY

Q functions: "Theorem about convexity of $Q(\mathbf{x}, \boldsymbol{\xi})$ and $\mathcal{Q}(\mathbf{x})$ for fixed recourse" (State and prove, B-L 89–90):
Theorem: For two-stage stochastic program with fixed recourse assuming that
$Q(\mathbf{x}, \boldsymbol{\xi}) \neq -\infty, \forall \mathbf{x}, \boldsymbol{\xi}$ we have:
(1) $Q(\mathbf{x}, \boldsymbol{\xi})$ is a convex function in $\mathbf{x}$ for all $\mathbf{x}$ feasible i.e. $\mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ and $\mathbf{x}$ such that $\mathcal{Q}(\mathbf{x}) < \infty$.
(2) $\mathcal{Q}(\mathbf{x})$ is a convex function in $\mathbf{x}$ under the same conditions.

Proof: (1) For $\boldsymbol{\xi}$ fixed we may write $\mathbf{q}$ and $\mathbf{y}$ instead of $\mathbf{q}(\boldsymbol{\xi})$ and $\mathbf{y}(\boldsymbol{\xi})$. For $\mathbf{t} = \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}(\boldsymbol{\xi})\mathbf{x}$ it is equivalent to prove that $\varphi(\mathbf{t}) = \min\{\mathbf{q}^\top\mathbf{y} \mid \mathbf{W}\mathbf{y} = \mathbf{t}, \mathbf{y} \geq \mathbf{0}\}$ is a convex in $\mathbf{t}$ (changing with $\mathbf{x}$ change). So, $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{R}^{m_2}$ and $\mathbf{t}_\lambda$ their convex combination. $\mathbf{y}_{1,\min}$ and $\mathbf{y}_{2,\min}$ are optimal solutions for $\mathbf{t}_1$ and $\mathbf{t}_2$. Then $\mathbf{y}_\lambda = \lambda\mathbf{y}_{1,\min} + (1-\lambda)\mathbf{y}_{2,\min}$ is feasible for $\min\{\mathbf{q}^\top\mathbf{y} \mid \mathbf{W}\mathbf{y} = \mathbf{t}_\lambda, \mathbf{y} \geq \mathbf{0}\}$. For $\mathbf{y}_{\lambda,\min}$ its solution, we get: $\varphi(\mathbf{t}_\lambda) = \mathbf{q}^\top\mathbf{y}_{\lambda,\min} \leq \mathbf{q}^\top\mathbf{y}_\lambda = \mathbf{q}^\top(\lambda\mathbf{y}_{1,\min} + (1-\lambda)\mathbf{y}_{2,\min}) = \lambda\varphi(\mathbf{t}_1) + (1-\lambda)\varphi(\mathbf{t}_2)$ and proof is complete.

(2) Because $\mathcal{Q}(\mathbf{x}) = E_\xi\{Q(\mathbf{x}, \boldsymbol{\xi})\}$ then for $\mathbf{x}_1$ and $\mathbf{x}_2$ we obtain $\mathcal{Q}(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) = E_\xi\{Q(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2, \boldsymbol{\xi})\} \le E_\xi\{\lambda Q(\mathbf{x}_1, \boldsymbol{\xi}) + (1-\lambda)Q(\mathbf{x}_2, \boldsymbol{\xi})\}$ and linearity of E_ξ completes the proof.

Separable simple recourse theorem: It is **omitted**. "Fundamental theorem for separable simple recourse" is included for completeness:
Theorem: For two-stage program with separable simple recourse

$$\min_{\mathbf{x}}\{\mathbf{c}^\top\mathbf{x} + \mathcal{Q}(\mathbf{x}) \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}, \quad \text{where}$$
$$\mathcal{Q}(\mathbf{x}) = E_\xi\{Q(\mathbf{x}, \boldsymbol{\xi})\}, \quad \text{and}$$
$$Q(\mathbf{x}, \boldsymbol{\xi}) = \min_{\mathbf{y}(\xi)}\{\mathbf{q}^\top\mathbf{y}(\boldsymbol{\xi}) \mid \mathbf{W}\mathbf{y}(\boldsymbol{\xi}) = \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}\mathbf{x}, \mathbf{y}(\boldsymbol{\xi}) \ge \mathbf{0}\ a.s.\}.$$

where $\mathbf{q} \ge \mathbf{0}$ and $\mathbf{W} = (\mathbf{I}, -\mathbf{I})$, the function $Q(\mathbf{x}, \boldsymbol{\xi})$ may be represented as a separable function in $\boldsymbol{\chi} = \mathbf{T}\mathbf{x}$.

Proof: We denote components of $\mathbf{q}$ and $\mathbf{y}(\boldsymbol{\xi})$ according columns of $\mathbf{W}$ as $\mathbf{q}^+$ and $\mathbf{y}^+(\boldsymbol{\xi})$ for $\mathbf{I}$ submatrix and as $\mathbf{q}^-$ and $\mathbf{y}^-(\boldsymbol{\xi})$ for $-\mathbf{I}$. $Q(\mathbf{x}, \boldsymbol{\xi}) = \sum_{i=1}^{m} Q_i(\chi_i, \boldsymbol{\xi})$ where

$$Q_i(\chi_i, \boldsymbol{\xi}) = \begin{cases} (h_i(\boldsymbol{\xi}) - \chi_i)q_i^+ & \text{if } \chi_i < h_i(\boldsymbol{\xi}) \\ -(h_i(\boldsymbol{\xi}) - \chi_i)q_i^- & \text{if } \chi_i \ge h_i(\boldsymbol{\xi}) \end{cases}.$$

As $Q(\mathbf{x}, \boldsymbol{\xi})$
$= \min_{\mathbf{y}^+(\xi), \mathbf{y}^-(\xi)}\{\mathbf{q}^{+\top}\mathbf{y}^+(\boldsymbol{\xi}) + \mathbf{q}^{-\top}\mathbf{y}^-(\boldsymbol{\xi}) \mid \mathbf{y}^+(\boldsymbol{\xi}) - \mathbf{y}^-(\boldsymbol{\xi}) = \mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}\mathbf{x}, \mathbf{y}^+(\xi), \mathbf{y}^-(\xi) \ge \mathbf{0}\}$
then by duality theorem:
$Q(\mathbf{x}, \boldsymbol{\xi}) = \max_{\mathbf{u}(\boldsymbol{\xi})}\{(\mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}\mathbf{x})\mathbf{u}(\boldsymbol{\xi}) \mid \mathbf{q}^- \le \mathbf{u}(\boldsymbol{\xi})\mathbf{q}^+\}$. So, we see that

$$u_{i\min}(\boldsymbol{\xi}) = \begin{cases} q_i^+ & \text{if } (\mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}\mathbf{x})_i > 0 \text{ i.e. } \chi_i < h_i(\boldsymbol{\xi}) \\ q_i^- & \text{if } (\mathbf{h}(\boldsymbol{\xi}) - \mathbf{T}\mathbf{x})_i \le 0 \text{ i.e. } \chi_i \ge h_i(\boldsymbol{\xi}) \end{cases}.$$

So we may substitute $u_{i\min}(\boldsymbol{\xi})$ and we have the separable function in $\boldsymbol{\chi}$.