CONTINGENT PORTFOLIO PROGRAMMING UNDER INCOMPLETE PROBABILITY INFORMATION

MAT-2108 INDEPENDENT RESEARCH PROJECT IN APPLIED MATHEMATICS

Juha Ojala
jsojala@cc.hut.fi

System Analysis Laboratory
Helsinki University of Technology
Table of Contents

1  Introduction .............................................................................................................. 4
2  Overview of Contingent Portfolio Programming .................................................... 6
   2.1  Main Elements ..................................................................................................... 6
       2.1.1  Time Axis ................................................................................................... 6
       2.1.2  Scenarios .................................................................................................... 6
       2.1.3  Resources .................................................................................................. 7
       2.1.4  Projects ..................................................................................................... 7
2.2  Objective of CPP .................................................................................................. 9
2.3  Summary of the Method ....................................................................................... 9
3  Modelling of Incomplete Information .................................................................... 10
   3.1  Problems with Data Requirements ..................................................................... 10
   3.2  Time Dependent and Independent Uncertainties .............................................. 11
   3.3  Modeling of Time Dependent Uncertainties ..................................................... 12
       3.3.1  Sibling Scenarios ....................................................................................... 13
       3.3.2  Example .................................................................................................... 13
   3.4  Modeling of Time Independent Uncertainties ................................................. 15
       3.4.1  Feasible Scenario Probabilities ................................................................. 15
       3.4.2  Extreme Points .......................................................................................... 16
4  Decision Making under Incomplete Information .................................................... 18
   4.1  Value Intervals of the Objective Function ....................................................... 18
   4.2  Quasiconvexity of the Objective Function ...................................................... 19
   4.3  Dominance Structures ....................................................................................... 21
   4.4  Decision Rules .................................................................................................. 21
       4.4.1  Maximax .................................................................................................... 24
4.4.2 Maximin ........................................................................................................25
4.4.3 Expected .......................................................................................................26
4.4.4 Minimax Regret ..........................................................................................26

5 Illustrative Example ........................................................................................27
5.1 Modeling Uncertainty ....................................................................................29
5.2 CPP Objective Function and Constraints .....................................................31
5.3 Applying Decision Rules ................................................................................34
  5.3.1 Maximax ....................................................................................................34
  5.3.2 Maximin ....................................................................................................35
  5.3.3 Expected ....................................................................................................35
  5.3.4 Minimax Regret .......................................................................................36

6 Discussion .........................................................................................................37
  6.1 Model Complexity ........................................................................................37
  6.2 Comparison of Decision Rules .....................................................................39
  6.3 Utilization of Presented Methods ..................................................................39

7 Conclusion ........................................................................................................40

Appendix ...............................................................................................................41
Convexity of the Set of Feasible Scenario Probabilities .................................41
Extreme Points of the Set of Feasible Scenario Probabilities .........................42

References ..........................................................................................................43
1 Introduction

A large variety of methods has been developed to facilitate the task of \textit{project portfolio selection}. The most simple methods can be used to estimate the value of a single project, whereas the more developed methods can also take account of project interdependencies and multiple time periods (Martino 1995). One general drawback of the methods is that they require the estimation of many parameters, such as risk-adjusted discount factors or probabilities of future scenarios (states of the world).

Project portfolio selection is an important part of \textit{project portfolio management}. It refers to the activity of selecting a portfolio from available project proposals and projects underway in order to meet or exceed the needs and expectations of an organization’s investment strategy. Selection must be made without exceeding available resources or violating other constraints (Archer and Ghasemzadeh 1999).

In broad terms, one can distinguish between three types of approaches that have been applied to the problem of project selection (Martino 1995). These are

1. \textit{scoring} (or \textit{multiple criteria decision making}, MCDM) \textit{methods} such as \textit{value trees},

2. \textit{optimization models} based on linear/nonlinear programming, and

3. \textit{dynamic programming models}, such as \textit{decision trees} and \textit{real options}.

Scoring models (see, e.g., Martino 1995) offer effective ways to estimate the value of a single project. They enable the consideration of any essential aspect of projects, such as future cashflows, strategic value, stakeholder satisfaction etc. However, scoring models are not very advisable in selecting a portfolio of projects, because they do not offer simple ways to consider the influence of project correlation and interaction.

Optimization methods (see, e.g., Luenberger 1998) do not consider projects in isolation but as a portfolio. They can, for example, take into account project interactions and resource and budget constraints. Shortcoming of many optimization models is that they do not model external uncertainties or project correlations. Furthermore, methods typically assume that every project involves only a single decision, the “go / no go” – decision at the beginning of the project.
The advantage of decision trees compared to scoring or optimizing methods is that they can model external uncertainties and projects with multiple decisions. The shortcomings of decision trees are similar to the shortcomings of scoring models, since they usually consider projects in isolation. Clemen (1996), for instance, presents examples about the use of decision trees.

Real options approach assumes that there is some underlying asset on which the outcome of the project depends. Furthermore, it is assumed that the asset follows continuous stochastic processes. These assumptions are extremely restrictive, thus the number of real option applications in project portfolio selection is limited. For examples about the use of real options, see Luenberger (1998) or Hull (1997).

In their recent paper Gustafsson and Salo (2001a) present a novel method – entitled contingent portfolio programming (CPP) – for the modeling of a portfolio of risky projects. CPP utilizes scenarios, decision trees and linear optimization, which enables many of the sophisticated aspects of the method, such as considering of project correlations and risk. However, the use of CPP calls for exact estimates about the probabilities of scenarios.

The need of exact estimates of parameters is a common attribute of decision making models. One method that allows the use of value intervals instead of point estimates in scoring models is the PRIME (preference ratios in multiattribute evaluation) method of Salo and Hämäläinen (see, e.g., Salo and Hämäläinen 1999 or Salo and Hämäläinen 2001). Also Salo and Bunn (1995) discuss the use of value intervals in scoring models.

The object of this study is to develop methods necessary to apply CPP when the probabilities of scenarios are not known exactly. This would ease the probability estimation process, increase the robustness of the method and offer ways to sensitivity analysis.

The rest of this paper is structured as follows. Section 2 provides an introduction to contingent portfolio programming. Sections 3 and 4 present methods to model uncertainties of probabilities and to apply CPP under uncertainty. An example is given in section 5 followed by discussion of the presented methods in section 6. Section 7 concludes the paper.
2 Overview of Contingent Portfolio Programming

In this section we summarize the concepts and methods presented in the paper of Gustafsson and Salo (2001).

The basic idea behind contingent portfolio programming is to combine the advantages of optimization models and decision trees. The possible future states of the world are modelled with a scenario tree, which enables the consideration of project correlations.

2.1 Main Elements

A contingent portfolio programming model is characterized by four main elements:

1. the time axis,
2. scenarios that represent alternative states of the world of one time period,
3. projects which involve multiple decisions, and
4. resources, such as money, labor and raw materials, that can be consumed and produced by the projects.

2.1.1 Time Axis

The use of time axis allows the modeling of consecutive scenarios and decisions. The set of all time periods under observation, the time frame, is denoted by $TT=[0,1,\ldots,T]$, where $T$ is the planning horizon.

2.1.2 Scenarios

The scenario tree of a CPP model is used to capture the risks of an independent project and also the correlations between different projects. See Figure 8 in page 28 for an example of a scenario tree. Table 1 summarizes some notation related to scenarios.

The probability, by which scenario $s$ succeeds its parent scenario $B(i)$, is defined by a conditional probability density function

$$p_{B(i)}(s|B(i)), p_{B(i)} : S_{B(i)} \to [0,1].$$

(1)

The nonconditional probabilities $p(i)$ can be calculated recursively with the help of conditional probabilities.
Table 1. Notation related to scenarios.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>scenario</td>
<td></td>
</tr>
<tr>
<td>$s_0$</td>
<td>base scenario, the first scenario of the scenario tree</td>
<td></td>
</tr>
<tr>
<td>$B(s)$</td>
<td>the parent scenario of scenario $s$</td>
<td>$B(s_0)=s_0$</td>
</tr>
<tr>
<td>$t(s)$</td>
<td>the time period of scenario $s$</td>
<td></td>
</tr>
<tr>
<td>$S_t$</td>
<td>the set of all scenarios in time period $t$</td>
<td></td>
</tr>
<tr>
<td>$S$</td>
<td>the set of all scenarios</td>
<td></td>
</tr>
</tbody>
</table>

A scenario path is temporally ordered sequence of successive scenarios beginning from the first period and ending in the last. So the general form of a scenario path is $\mathbf{p}=(s_0,s_1,s_2,\ldots,s_t)^T$. The probability of a scenario path is denoted with $p_{\text{path}}(\mathbf{p})$ and it is equal to the probability of its last element. That is, $p_{\text{path}}(\mathbf{p})=p(s_t)$. The set of all scenario paths is denoted by $\mathcal{SP}$.

2.1.3 Resources

In CPP-model, the term resources refers to all factors of production, such as money, manpower, raw material and equipment, that are consumed or produced by the projects. If appropriate, also intangible outputs of projects, such as strategic benefit, can be thought of as resource and taken into account in the objective function. The notation related to resources is presented in Table 2.

Table 2. Notation related to resources.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>resource</td>
<td></td>
</tr>
<tr>
<td>$R$</td>
<td>the set of all resources</td>
<td></td>
</tr>
<tr>
<td>$w_r$</td>
<td>represents the importance of the resource</td>
<td></td>
</tr>
<tr>
<td>$\delta_r(t)$</td>
<td>represents the relative importance of flows of the resource at different time periods</td>
<td></td>
</tr>
<tr>
<td>$a_r$</td>
<td>determines the proportion of unconsumed resources that are available in the next period</td>
<td></td>
</tr>
<tr>
<td>$b_r$</td>
<td>initial inflow of the resource in each scenario</td>
<td></td>
</tr>
</tbody>
</table>

2.1.4 Projects

The term project is used here to refer to any investment opportunities that are available to the decision maker, not just to projects in traditional meaning. In principle, a project can be any object that consumes or produces resources and involves some kind of decisions.
There is a number of decision points in each project. Actions taken in these points influence the resource flows generated by the project. Every action is associated with a decision variable \( X_{z,d,a} \) called action variable, that assumes the value equal to the number of selections of the action. A vector containing action variables associated with every decision point in project \( \zeta \) is called a project \( \zeta \) specific strategy, and it is denoted by \( \mathbf{X}_\zeta \). A Vector of all action variables in the entire portfolio is called a portfolio strategy and it is denoted by \( \mathbf{X} \). Some notation related to projects is introduced in Table 3.

**Table 3. Notation related to projects.**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \zeta )</td>
<td>single project</td>
<td></td>
</tr>
<tr>
<td>( Z )</td>
<td>the set of all available projects</td>
<td></td>
</tr>
<tr>
<td>( L_{\zeta} )</td>
<td>indicates how many times project can be selected</td>
<td></td>
</tr>
<tr>
<td>( X_{\zeta,d,a} )</td>
<td>action variable</td>
<td></td>
</tr>
<tr>
<td>( F_{\zeta,d,a} )</td>
<td>the set of feasible values of ( X_{\zeta,d,a} )</td>
<td></td>
</tr>
</tbody>
</table>

The decision points and actions of a project form a decision tree, in which every decision point except the first one has a parent action \( ap(d) \) and a parent decision point \( dp(d) \). For an example of a decision tree, see Figure 9 in page 28. The sets of all decision points and actions are denoted by \( D \) and \( A \), respectively.

The projects generate resource flows which are defined by resource flow functions,

\[
RF^\zeta_\zeta(\mathbf{X}_\zeta, s) = \sum_{d \in D_{\zeta}} \sum_{a \in A_d} \varepsilon_{z,d,a}^\zeta(s) \cdot X_{\zeta,d,a} ,
\]

where \( \varepsilon_{z,d,a}(\delta) \) denotes the flow of resource \( r \) implied by the action \( a \) of the decision point \( d \) when scenario is \( s \) and time is \( t(\delta) \). Portfolio value over the scenario path \( \varphi \) is the present value of the resource flows in the scenario path:

\[
V_\varphi(\mathbf{X}) = \sum_{r \in R} \sum_{t=1}^{T} \delta_r(t) \sum_{\zeta \in Z} RF^\zeta_\zeta(\mathbf{X}_\zeta, s) .
\]

Thus, the expected portfolio value can be calculated as

\[
EV(\mathbf{X}) = \sum_{\varphi \in \mathcal{X}} p_{\varphi} \cdot V_\varphi(\mathbf{X}) .
\]
2.2 Objective of CPP

The objective of CPP is to maximize the certainty equivalent of the portfolio value. In utility theory, the certainty equivalent of a random wealth variable \( X \) is defined to be the amount of a certain (risk-free) wealth that has utility level equal to the expected utility of \( X \) (Keeney and Raiffa 1976). However, CPP employs the more general form of certainty equivalent:

Certainty Equivalent = Expected Value – Risk Premium.

CPP applies risk premium that depends linearly on the mean-lower semi-absolute deviation (MLSAD) of the portfolio value. Mean-lower semi-absolute deviation of a discrete random variable \( X \) is defined by the equation

\[
\text{semiabsdev}[X] = \sum_{x < \mu} p(x)(\mu - x),
\]

(5)

where \( p(x) \) is the probability density function of \( X \) and \( \mu = E[X] \).

In CPP, deviation variables \( \Delta V^{\varphi} \) and \( \Delta V^{\varphi}_{ip} \) measure the amount by which the portfolio value over scenario path \( \varphi \) differs from the expected portfolio value. Thus, the risk premium employing MLSAD can be calculated with the equation

\[
RP(\Delta V^-) = k \cdot \sum_{i \in \Omega_p} p_{\varphi}(\varphi) \cdot \Delta V^{-}_{ip},
\]

(6)

where coefficient \( k \in [0,1] \) represents the risk attitude of the decision maker and \( \Delta V \) is the vector of all \( \Delta V^{-}_{ip} \)'s.

2.3 Summary of the Method

Contingent portfolio programming method can be summarized as follows:

\[
\max \left( EV(X) - RP(\Delta V^-) \right)
\]

such that

\[(1a) \quad \sum_{x \in D} X_{\varphi,d} = L_{\varphi} \quad \forall d \in D \text{ for which } ap(d) = "no parent" \]

\[(1b) \quad \sum_{x \in D} X_{\varphi,d} = X_{\varphi,\varphi(d)} \quad \forall d \in D \text{ for which } ap(d) \neq "no parent" \]
(2a) \[ b^r (s_0) + RF^r (X, s_0) - RS^r = 0 \quad \forall r \in R \]

(2b) \[ b^r (s) + RF^r (X, s) + \alpha_j RS^r_{b(j)} - RS^r = 0 \quad \forall s \neq s_0 \quad \forall r \in R \]

(3) \[ V_{ip} (X) - E V (X) - \Delta V_{ip}^+ + \Delta V_{ip}^- = 0 \quad \forall sp \in SP \]

(4) Optional constraints are adhered

(5) Decision variables are constrained as follows:

(5) \[ X_{\zeta, \varphi} \in F_{\zeta, \varphi} \quad \forall a \in A_{\theta} \quad \forall d \in D_{\zeta} \quad \forall \zeta \in Z \]

(6) \[ RS^r \geq 0 \quad \forall s \in S \quad \forall r \in R \]

(7) \[ \Delta V_{ip}^+ \geq 0 \quad \forall sp \in SP \]

(8) \[ \Delta V_{ip}^- \geq 0 \quad \forall sp \in SP \]

Constraints (1a) and (1b) assure the consistency of subsequent decisions, (2a) and (2b) assure that resource limitations are not exceeded and (3) is used to calculate the deviation variables. Optional constraints (4) may include, e.g., prerequisite constraints. For discussion about optional constraints see, e.g., Gustafsson and Salo (2001a) or Ghasemzadeh et al. (1999).

3 Modelling of Incomplete Information

3.1 Problems with Data Requirements

Contingent portfolio programming addresses many of the limitations of earlier project portfolio selection approaches. However, one of the challenges of CPP arises from this comprehensiveness: in order to take advantage of this method, the decision maker has to estimate a number of parameters. These parameters are \( w_i \)'s that reflect the relative importance of resources, \( k \) that reflects the decision maker’s risk attitude, and the probabilities of future scenarios, \( p(\delta) \)'s.

The estimation of the probabilities of future scenarios is a task that has to be done independently for every project selection problem, because the relevant future scenarios considered will usually change as time goes by and the project proposals change. It may also be time-consuming, depending on the nature and number of considered scenarios. In this
paper we present methods to ease the problem of estimating the probabilities of future scenarios by allowing the use of *probability intervals* instead of point estimates.

The estimation of weights $w_i$ and coefficient $k$ is an important task. However, once the decision maker has determined these parameters, he/she can use approximately the same values in all forthcoming problems that are similar enough to the original problem. For discussion about coefficient $k$ see Dietrich (2001).

### 3.2 Time Dependent and Independent Uncertainties

It is reasonable to assume, that if the current state of the world is known exactly and available resources for the determination process are unlimited, the probabilities of future scenarios can be accurately determined. Resources needed are, e.g., skilled manpower and time.

When applying CPP one is interested in conditional probabilities $p_{B(i)}(s|B(j))$. There are two kinds of main difficulties related to the estimation of these probabilities:

1. First kind of difficulty raises from the *definitional ambiguity of the scenarios*. That is, all future scenarios used in a model include a set of possible states of the world instead of one exactly defined state. For instance, scenario “oil price is from $25$ to $30$ in september 2002” is reasonable, but scenario “oil price is $27.4$ in september 2002” is not. Thus, because of the definitional ambiguity of some future scenario $B(i)$, there does not even exist exact conditional probabilities $p_{B(i)}(s|B(j))$. However, if the scenario $B(i)$ occurs, the state of the the world is known exactly (“the current oil price is $28.3$”) and the exact conditional probabilities of it’s child scenarios exist.

2. Difficulties raise also from resource limitations, i.e. the decision maker does not have enough resources to exactly determine the conditional probabilities $p_{B(i)}(s|B(j))$ even if the scenario $B(i)$ includes only one detailed state of the world.

For the two reasons presented above there always exists uncertainty about the conditional probability $p_{B(i)}(s|B(j))$ of a scenario. For modeling purposes, this uncertainty can be divided into two components.
1. Component that resolves as time goes by, called *time dependent uncertainty*. The resolving occurs because the time of $B(i)$ comes closer and also because the decision maker consumes time and other resources to get more accurate estimates.

2. Component of uncertainty that remains at the time of $B(i)$, called *time independent uncertainty*. This component does not resolve because the decision maker does not have resources to get a better estimate.

Typically in real world situations both kinds of uncertainties are present at the same time, because it is possible to include only a finite number of alternative scenarios in the model and also the available resources for the probability determination process are limited. Figure 1 illustrates how the accuracy of the estimate of conditional probability of some scenario $s$, occurring at time period four, increases as time goes by. However, at the time of $B(i)$ the time independent component of the uncertainty still remains.

![Figure 1](image-url)  
*Figure 1. Conditional probability of scenario $s$, occurring at time period 4, as a function of time.*

### 3.3 Modeling of Time Dependent Uncertainties

CPP can pretty straightforwardly handle time dependent uncertainties of scenario probabilities by using so called *sibling scenarios*. 

---

12
3.3.1 Sibling Scenarios

Typically the scenarios of the same time period are characterized by

1. different project resource flows and
2. different child scenarios with some conditional probabilities.

Sibling scenarios always have the same parent scenario and are characterized by

1. the same project resource flows and
2. the same child scenarios with different conditional probabilities.

Thus, sibling scenarios differ only with the probabilities of their child scenarios. Figure 2 illustrates the differences between ordinary and sibling scenarios.

![Table 1]

**Figure 2.** Characteristics of two scenarios that have the same parent scenario and the same child scenarios.

3.3.2 Example

Suppose that we have constructed a simple CPP model, the scenario tree of which is shown in the left side of Figure 3. Because of incomplete information, the conditional probabilities \( p_{h1}(s_{i1} | s_i) \) and \( p_{h1}(s_{i2} | s_i) \) in the scenario tree are not presented as point estimates but intervals. However, we know that at the time of scenario \( s_i \) we will get much more accurate estimates of these probabilities.
Because CPP can’t straightly handle probability intervals, we replace the scenario \( s_i \) with two sibling scenarios \( s_{ia} \) and \( s_{ib} \). The only difference between these scenarios is that the conditional probabilities of their child scenarios are different. The probabilities of \( s_{ia} \) and \( s_{ib} \) are set to 0.5, because we assess that both sibling scenarios are equally likely.

\[
\begin{align*}
S_0 &\rightarrow S_1 \\
S_1 &\rightarrow S_{11} \\
&\rightarrow S_{12}
\end{align*}
\]

\[
\begin{align*}
S_0 &\rightarrow S_{1a} \\
&\rightarrow S_{1a1} \\
S_{1a} &\rightarrow S_{1a2} \\
S_0 &\rightarrow S_{1b} \\
S_{1b} &\rightarrow S_{1b1} \\
&\rightarrow S_{1b2}
\end{align*}
\]

**Figure 3.** Modifying the scenario tree to represent time dependent uncertainties. Scenarios \( s_{ia} \) and \( s_{ib} \) are called sibling scenarios.

The method used here is not restricted in scenarios with two child scenarios. In the case of a scenario with more than two child scenarios, the decision maker has to employ a set of sibling scenarios that comprehensively enough represents the possible probabilities of the child scenarios. The optimal selection of sibling scenarios is an issue worth considering, but not in the scope of this paper.

The method can also be straightforwardly applied to CPP models with many possible future scenarios. However, specific software is necessary because the size of the model grows rapidly as the function of the number of possible scenarios with time dependent uncertainties.

For modeling point of view, the method of using sibling scenarios demands only the assessment of the conditional probabilities of the sibling scenarios. The uniform distribution may be the right choice unless there is some specific knowledge about the probabilities. Thus, the modeling task itself does not get any more demanding.
3.4 Modeling of Time Independent Uncertainties

In previous subsection we saw how time dependent uncertainties can be incorporated into CPP models. However, CPP does not have capabilities to deal with time independent uncertainties. In the next section we present methods for decision making under time independent uncertainties and in this section we introduce the additional notation and concepts needed.

For computational reasons it is assumed that the absolute number of future scenarios with uncertain conditional probability is order of ten at maximum.

3.4.1 Feasible Scenario Probabilities

The original CPP model presented in section 2 requires the exact determination of conditional probabilities $p_{B(b)}(x | B(b))$. The methods presented in the next section require only the set of feasible conditional probabilities of child scenarios for every scenario $s$. That is, for every scenario $s$ one must have the set

$$V_s(s) = \left\{ \left( q_s(x_1|s), q_s(x_2|s), \ldots, q_s(x_{last}|s) \right)^T \mid \begin{array}{l} q_s(x|s) \in [0,1] \ \forall x \in C(s), \\ \sum_{x \in C(s)} q_s(x|s) = 1, \ \ D_s \end{array} \right\},$$

(8)

where $C(s) = \{x_1, x_2, \ldots, x_{last}\}$ is the set of $s$’s child scenarios, the $q_s(x|s)$’s represent possible conditional probabilities of $x$’s and $D_s$ is a set of optional linear constraints on $q_s(x|s)$’s. The constraints $q_s(x|s) \in [0,1] \ \forall x \in C(s)$ and $\sum_{x \in C(s)} q_s(x|s) = 1$ guarantee that $q_s(x|s)$’s represent probabilities, and the optional constraints reflect the decision maker’s view on the probabilities of $x$’s.

The set $D_s$ may include, for instance, following kinds of constraints:

- point estimates, $q_s(x_i|s) = 0.6$ ,
- intervals, $q_s(x_i|s) \in [0.4,0.7]$, and
- comparisons, $q_s(x_1|s) \geq q_s(x_2|s) + q_s(x_3|s)$.

The optional constraints must be set so that the sets $V_s(s)$ contain at least one element. If the decision maker does not know anything about the conditional probabilities of $s$’s child
scenarios the set \( D \) is empty. For discussion about the probability elicitation process see, for instance, Clemen (1996).

Because the probability of the base scenario \( s_0 \) is always 1, the non-conditional probabilities \( q(x) \)'s corresponding to \( q(x|s) \)'s can be calculated recursively by the equation

\[
q(x) = q_s(x|s) \cdot q(s).
\]

(9)

Equivalently, we get the set of feasible non-conditional probabilities of scenario \( s \)'s child scenarios by multiplying the elements of the set \( V_s(\delta) \) with \( q(s) \),

\[
V(s) = \left\{ \left( q(x_1), q(x_2), \ldots, q(x_{|\delta|}) \right) \mid q(x) \in [0, q(s)] \quad \forall x \in C(s), \quad \sum_{x \notin C(s)} q(x) = q(s), \quad D' \right\},
\]

(10)

where \( D' \) denotes the constraints of the set \( D \) multiplied by \( q(s) \). The set \( D' \) may include, for example, following kinds of constraints:

- point estimates, \( q(x) = 0.6 \cdot q(s) \),
- intervals, \( q(x) \in [0.4 \cdot q(s), 0.7 \cdot q(s)] \), and
- comparisons, \( q(x_1) \geq q(x_2) + q(x_3) \).

The vector consisting of feasible \( q(\delta) \)'s of all scenarios is denoted by \( Q \) and the set of all \( Q \)'s by \( V_q \).

\[
V_Q = \left\{ \left( q(s_0), \ldots, q(s_{|\delta|}) \right) \mid q(s) \in [0, q(B(s))] \quad \forall s \neq s_0, \quad \sum_{s \in C(s)} q(x) = q(s) \quad \forall s \in S, \quad D' \quad \forall s \in S \right\},
\]

(11)

It should be clear that the set \( V_Q \) is convex. However, the proof is presented in Appendix.

### 3.4.2 Extreme Points

Because the set \( V_Q \) is convex and restricted by linear constraints, it's obvious that every \emph{extreme point} of \( V_Q \) (i.e. point that cannot be expressed as a convex combination of any two distinct points in the same set) is associated with an intersection point of two or more linear constraints. As stated in the beginning of this section, we assume that the number of scenarios with uncertain conditional probability is moderate, i.e., order of ten at maximum.
Then also the number of linear constraints on scenario probabilities is moderate and it’s computationally reasonable to identify all the extreme points of $V_Q$. This set of extreme points of $V_Q$ is denoted by $V_Q^\epsilon$.

One way to identify the set $V_Q^\epsilon$ is to exploit the extreme points of the sets $V_{\beta}(\sigma)$. With proof like the one in previous section, it is easy to show that the sets $V_{\beta}(\sigma)$ are convex. We denote the set of the extreme points of a set $V_{\beta}(\sigma)$ by $V_{\beta}^\epsilon(\sigma)$. Because the sets $V_{\beta}(\sigma)$ are convex and restricted by linear constraints, their every extreme point is associated with an intersection point of two or more constraints. Thus, the sets $V_{\beta}^\epsilon(\sigma)$ are relatively easy to determine. For example in the case of a scenario $s$ with two child scenarios $x_1$ and $x_2$, and with an optimal constraint $q_3(x_1|\sigma) \geq 3 \cdot q_3(x_2|\sigma)$, the set $V_{\beta}^\epsilon(\sigma)$ consists only of two items, $V_{\beta}^\epsilon(\sigma) = \{(1,0)^T, (0.75,0.25)^T\}$.

In the case of a scenario $s$ with three child scenarios $x_1$, $x_2$ and $x_3$, and with optional constraints $q_4(x_1|\sigma) \in [0.3,0.5]$, $q_4(x_2|\sigma) \in [0.35,0.56]$ and $q_4(x_3|\sigma) \in [0.1,0.21]$, the set $V_{\beta}^\epsilon(\sigma)$ consists of six items,

$$V_{\beta}^\epsilon(\sigma) = \left\{ \left(0.3,0.56,0.14\right)^T, \left(0.3,0.49,0.21\right)^T, \left(0.44,0.35,0.21\right)^T \right\}.$$

Graphically these points are the corner points of a plane in three-dimensional space as illustrated in Figure 4.

A cartes product of the sets $V_{\beta}^\epsilon(\sigma)$, $V_{\beta}^\epsilon = \prod_{\sigma \in \Sigma} V_{\beta}^\epsilon(\sigma)$, consists of every feasible combination of the extreme points of the sets $V_{\beta}^\epsilon(\sigma)$. Every element of this product equals to one element of the set $V_Q^\epsilon$. To transform an element of the set $V_{\beta}^\epsilon$ to an element of the set $V_Q^\epsilon$, one has to multiply the corresponding conditional probabilities of $V_{\beta}^\epsilon$, i.e.

$$q(s) = q_{\beta}(1) \cdot q_{\beta}(B(s) \mid B) \cdot q_{\beta}(B(s) \mid B) \cdots q(s_n),$$

where $q(s_n) = 1$. Thus, the set $V_Q^\epsilon$ can be identified by determining the sets $V_{\beta}^\epsilon(\sigma)$ and calculating their cartes product. For proof, see Appendix.
4 Decision Making under Incomplete Information

In this section we present methods for decision making under time dependent uncertainties. More specifically, we will introduce value intervals of the objective function, the concept of dominance structures and utilize certain decision rules (see e.g. Bunn 1984 or Salo and Hämäläinen 2001).

The methods presented in this section (excluding maximax criterion) are devised under the assumption that the number of considered future scenarios with uncertain conditional probability is small compared to the number of possible portfolio strategies. Otherwise the optimization problems presented are not reasonable. For computational reasons also the absolute number of future scenarios with uncertain conditional probability must be order of ten at maximum.

4.1 Value Intervals of the Objective Function

For convenience, let’s denote the objective function of CPP by $f(X,Q)$,
\[
f(\mathbf{X}, \mathbf{Q}) = EV(\mathbf{X}, \mathbf{Q}) - RP(\Delta V^-) = \sum_{\mathcal{P} \in \Omega} q_{\mathcal{P}}(s) \cdot V_{\mathcal{P}}(\mathbf{X}) - k \cdot \sum_{\mathcal{P} \in \Omega} q_{\mathcal{P}}(s) \cdot \Delta V_{\mathcal{P}}^- ,
\]

where \( q_{\mathcal{P}}(s) \) is the path probability of scenario path \( s \) implied by \( \mathbf{Q} \). We use the notation \( f(\mathbf{X}, \mathbf{Q}) \) instead of \( f(\mathbf{X}, \mathbf{Q}, \Delta V^-) \) because \( \Delta V^- \) is a function of \( \mathbf{X} \) and \( \mathbf{Q} \).

\( \mathbf{Q} \)'s represent possible scenario probabilities. In the original CPP model scenario probabilities are point estimates and the set of feasible \( \mathbf{Q} \)'s, \( \mathbf{V}_Q \), consists only of one item. When the uncertainties are introduced, the number of elements of \( \mathbf{V}_Q \) grows to infinity. With any feasible \( \mathbf{Q} \) it's possible to calculate the value of the objective function of CPP for every possible portfolio strategy \( \mathbf{X} \). So in addition to \( \mathbf{X} \) the value of the objective function depends also on \( \mathbf{Q} \).

Thus, for every portfolio strategy \( \mathbf{X} \) one can calculate infinite number of objective function values. These values constitute an interval, thus the problem of selecting the strategy with the highest objective function value transforms to the problem of selecting the strategy with the best interval of the CPP objective function. In general case, when there is no dominating alternative, there's no exact definition of the best interval. Thus, there exists only ways to determine the best interval in some sense.

### 4.2 Quasiconvexity of the Objective Function

Some of the methods we present require the assumption that the maximum of \( f(\mathbf{X}, \mathbf{Q}) \) over the set \( \mathbf{V}_Q \) is associated with one of the extreme points of \( \mathbf{V}_Q \). This is true if for every \( \mathbf{X} \) the function \( f(\mathbf{X}, \mathbf{Q}) \) is quasiconvex, i.e., if

\[
f(\mathbf{X}, (a \mathbf{Q}_1 + (1 - a) \mathbf{Q}_2)) \leq \max\{f(\mathbf{X}, \mathbf{Q}_1), f(\mathbf{X}, \mathbf{Q}_2)\} \quad \forall \mathbf{Q}_1, \mathbf{Q}_2 \in \mathbf{V}_Q.
\]

(13)

We could not prove the quasiconvexity of \( f(\mathbf{X}, \mathbf{Q}) \) over the set of \( \mathbf{V}_Q \). However, we could neither come up with a \( f(\mathbf{X}, \mathbf{Q}) \) that is not quasiconvex. Thus, we assume that \( f(\mathbf{X}, \mathbf{Q}) \) is either always or in most situations quasiconvex. In following we present examples that illustrate the forms of some \( f(\mathbf{X}, \mathbf{Q}) \)'s.

Suppose that we have a portfolio strategy \( \mathbf{X} \) that will yield $100m if scenario \( s_1 \) occurs and $0 if scenario \( s_2 \) occurs. Figure 5 illustrates the shape of the certainty equivalent of the portfolio value, \( f(\mathbf{X}, \mathbf{Q}) \), as the probability of \( s_1 \) changes. The value of coefficient \( k \) is 1.
Figure 5. One possible form of the CPP objective function.

Then suppose that also scenario $s_3$ is possible, in which case our portfolio would yield $50m. The probability of $s_3$ is known to be 0.5. Figure 6 illustrates the form of $f(X,Q)$ as the probability of $s_1$ changes. The probability of $s_2$ is now $q(s_2) = 0.5 - q(s_1)$.

Figure 6. One possible form of the CPP objective function.

As can be seen, both functions are strictly quasiconvex. Furthermore, linearity of the function grows as it turns more complex.
4.3 Dominance Structures

In literature, two kinds of dominance structures are widely used (see e.g. Gustaffsson et al. 2001b, Salo 1995 or Salo and Hämäläinen 2001). These dominance structures can be used to select the dominating interval from a set of intervals, or, if there does not exist one dominating interval, the structures can be used to discard the dominated intervals. Here we apply the absolute dominance and pairwise dominance structures to CPP objective function intervals.

Absolute dominance, which is more restrictive than pairwise dominance, is based on comparing the intervals of \( f(X,Q) \)'s. That is, strategy \( X \) is preferred to \( X' \) in the sense of absolute dominance if and only if the smallest value of \( f(X,Q) \) exceeds the largest value of \( f(X',Q) \), i.e.,

\[
X \succ^A X' \iff \min_Q f(X,Q) > \max_Q f(X',Q). \tag{14}
\]

According to pairwise dominance criterion, strategy \( X \) is preferred to \( X' \) if and only if the value of \( f(X,Q) \) exceeds the value of \( f(X',Q) \) for all \( Q \), i.e.,

\[
X \succ^P X' \iff \min_Q [f(X,Q) - f(X',Q)] > 0. \tag{15}
\]

However, dominance structures are not very useful if the number of possible portfolio strategies is great, as it usually is. There are two reasons for this. Firstly, when the number of possible strategies grow it’s unlikely that there will be an \( X \) that dominates all the other \( X \)'s. Secondly, the calculation of the maximum and minimum of \( f(X,Q) \) for all possible \( X \)'s may be too time-consuming. For these reasons the use of dominance structures is not discussed in more detail.

Anyway, it’s noteworthy that if \( X \) is preferred to \( X' \) according to either dominance criterion, \( X \) is preferred to \( X' \) also according to any of the decision rules presented in the following subsection.

4.4 Decision Rules

When selecting the best alternative from a set of intervals, it is customary to utilize some decision rules if dominance structures fail to give results (see e.g. Salo and Hämäläinen 2001). In general case, decision rules do not find the absolutely best interval, instead they
identify the best value interval in some sense. Therefore it is obvious that different rules may give different solutions. The decision rules utilized in this paper are:

1. **Maximax**: Choose the strategy \( X \) which largest value of \( f(X,Q) \) is largest over the set of all feasible \( Q \)'s, i.e. \( \max_Q f(X,Q) \geq \max_Q f(X',Q) \) \( \forall X' \neq X \).

2. **Maximin**: Choose the strategy \( X \) which least value of \( f(X,Q) \) is largest over the set of all feasible \( Q \)'s, i.e. \( \min_Q f(X,Q) \geq \min_Q f(X',Q) \) \( \forall X' \neq X \).

3. **Expected**: Choose the strategy \( X \) which expected value of \( f(X,Q) \), calculated over the extreme points of the set of feasible \( Q \)'s, i.e., over the set \( V_Q^e \), is greatest. That is,
\[
\frac{1}{|V_Q^e|} \sum_{Q \in V_Q^e} f(X,Q) \geq \frac{1}{|V_Q^e|} \sum_{Q' \in V_Q^e} f(X',Q) \forall X' \neq X,
\]
where \( |V_Q^e| \) denotes the number of elements in \( V_Q^e \).

4. **Minimax regret**: Choose the alternative \( X \) for which maximum regret, measured as the largest value difference between \( f(X,Q) \) and the value of other alternatives, is smallest, i.e.
\[
\max_{X' \neq X} \left[ f(X'',Q) - f(X',Q) \right] \leq \max_{X' \neq X} \left[ f(X'',Q) - f(X,Q) \right] \forall X' \neq X.
\]

Decision rules maximax, maximin and minimax regret are widely used in literature, while the rule expected is a kind of modification of the rule central values, according to which one should select the alternative \( X \) for which the midpoint of the value interval of \( f(X,Q) \) is greatest, i.e.,
\[
\left[ \max_Q f(X,Q) + \min_Q f(X,Q) \right] \geq \left[ \max_Q f(X',Q) + \min_Q f(X',Q) \right] \forall X' \neq X.
\]
However, this rule would lead to an optimization problem
\[
\max_X \left( \max_Q f(X,Q) + \min_Q f(X,Q) \right),
\]
which can’t be transformed into a linear optimization problem.

We motivate the presentation of expected decision rule with the following example. Assume that the probability \( q \) of a scenario \( s \) is symmetrically distributed random variable that may obtain values between \( a \) and \( b \). Assume also that with fixed strategy the value of a CPP
objective function $f(q)$ depends linearly on the probability of $s$. This situation is illustrated in the left side of Figure 7. We can calculate the expected value of $f(q)$ by

$$E[f(q)] = \int_a^b p(q) \cdot f(q) \, dq = \frac{1}{2} \left( f(a) + f(b) \right).$$

If the distribution of $q$ is almost symmetric and also the function $f(q)$ exhibits linear behaviour, following applies:

$$E[f(q)] = \int_a^b p(q) \cdot f(q) \, dq \approx \frac{1}{2} \left( f(a) + f(b) \right).$$

This situation is illustrated in the right side of Figure 7.

We can extend this method to models with more than two scenarios if the following conditions hold:

- function $f(Q)$ exhibits linear behaviour,
- the distribution of variable $Q$ is sufficiently symmetric, and
- the extreme points of the set $V_Q$ are distributed uniformly.

![Figure 7](image)

**Figure 7.** Possible forms of CPP objective function $f(q)$ and the distribution of $q, p(q)$.

As we argued in Section 4.2, the CPP objective function is quite linear. It is also reasonable to assume that the probabilities of scenarios are symmetrically distributed (e.g. Gaussian distribution). The distribution of the extreme points of the set $V_Q$ depends on the constraints set by the decision maker. For instance, in Figure 4 in p. 18 the extreme points are uniformly distributed. Notice that the extreme points of the set $V_Q$ are uniformly distributed if the extreme points of the sets $V_{i,j}$ are.
In the following subsections the use of these decision rules is presented in detail.

4.4.1 Maximax

Application of maximax decision rule leads to an optimization problem \( \max_{x,Q} f(X,Q) \), which can be presented as follows:

\[
\max \left( \sum_{q \in SP(\phi)} q_{path}(sp) \cdot V_{\phi}(X) - k \cdot \sum_{q \in SP(\phi)} q_{path}(sp) \cdot \Delta V_{\phi} \right),
\]

such that

(1-8)

The added decision variables \( q(\phi) \)'s are constrained as follows:

(9) \( \sum_{x \in C(\phi)} q(x) = q(\phi) \quad \forall \phi \in S \)

(10) \( q(\phi) \in [0, q(B(\phi)))] \quad \forall \phi \in S \)

(11) \( D'(\phi) \quad \forall \phi \in S \)

The notation (1-8) refers to constraints presented in the summary of the CPP method in section 2.3 and \( D'(\phi) \) is the set of optional linear constraints as explained in Section 3.4.1.

However, this optimization problem is quadratic, because there exists products of \( q(\phi) \)'s and \( X \)’s and products of \( q(\phi) \)'s and \( \Delta V_{\phi} \)'s in the objective function. If necessary, this problem can also be solved with linear optimization. Because the optimum is associated with some element of the set \( V'_{Q} \) (cf. section 4.2), one can find the optimum by solving the problem \( \max_{x} f(X,Q) \) with respect to every element of \( V'_{Q} \). After solving these problems, one can select the strategy that is associated with the highest objective function value. That is,

1. \( \forall Q \in V'_{Q} \):

\[
\nu_{Q} = \max \left( \sum_{q \in SP(\phi)} q_{path}(sp) \cdot V_{\phi}(X) - k \cdot \sum_{q \in SP(\phi)} q_{path}(sp) \cdot \Delta V_{\phi} \right),
\]

such that
(1-8)

2. Select $X$ that is associated with highest $\nu_Q$.

After solving the $\nu_Q$’s one can easily utilize minimax regret criterion (see section 4.4.4).

4.4.2 Maximin

Application of maximin criterion leads to an optimization problem

$$\max_x \left[ \min_Q f(X, Q) \right],$$

which can’t be directly solved using linear optimization. However, we can transform this problem into an equivalent problem (see e.g. Taha 1997):

$$\max \nu$$

such that

$$\nu \leq f(X, Q) \quad \forall Q \in V_Q$$

However, because the minimum of $f(X, Q)$ over $V_Q$ is identified by one of the extreme points of $V_Q$ (cf. section 4.2) we can replace the constraints by

$$\nu \leq f(X, Q) \quad \forall Q \in V_Q'.$$

Thus, the number of the constraints is equivalent to the number of the extreme points of $V_Q$. Now we can write the derived linear optimization problem as:

$$\max \nu$$

such that

$$\nu \leq \sum_{q \in \mathcal{SP}} q_{path}(sp) \cdot V_{q^*}(X) - k \cdot \sum_{q \in \mathcal{SP}} q_{path}(sp) \cdot \Delta V_{q^+} - \forall Q \in V_Q'$$

(1-2)

$$V_{q^*}(X) - EV(X, Q) - \Delta V_{q^+}(Q) + \Delta V_{q^-}(Q) = 0 \quad \forall sp \in \mathcal{SP} \quad \forall Q \in V_Q'$$

(3)

$$V_{q^*}(X) - EV(X, Q) - \Delta V_{q^+}(Q) + \Delta V_{q^-}(Q) = 0 \quad \forall sp \in \mathcal{SP} \quad \forall Q \in V_Q'$$

(4-8)

Notice that the third constraint of the original CPP method has to be written with respect to every extreme point of $V_Q$. 

4.4.3 Expected

Application of expected decision rule leads to a linear optimization problem

\[
\max_x \frac{1}{|V'\mid} \sum_{Q \in V'\cap Q_0} f(X, Q), \text{ which can be presented as follows:}
\]

\[
\max \frac{1}{|V'\mid} \sum_{Q \in V'\cap Q_0} \left( \sum_{e \in SP} q_{path}(e) \cdot V^+_e(X) - \sum_{e \in SP} q_{path}(e) \cdot \Delta V^-_{e} \right)
\]

such that

(1-2)

(3) \( V^+_e(X) - EV(X, Q) - \Delta V^+_e(Q) + \Delta V^-_{e} (Q) = 0 \) \( \forall e \in SP \) \( \forall Q \in V'\cap Q_0 \)

(4-8)

4.4.4 Minimax Regret

For every \( Q \in V'\cap Q_0 \) let \( \nu_Q \) be the solution of the original CPP optimization problem, i.e.,

\( \forall Q \in V'\cap Q_0 : \nu_Q = \max_x f(X, Q) \). Notice that these \( \nu_Q \)'s are identical to the \( \nu_Q \)'s of the maximax criterion.

Application of minimax regret criterion leads to following optimization problem:

\[
\min_x \left[ \max_Q (\nu_Q - f(X, Q)) \right],
\]

where the \( \nu_Q \)'s are solutions of ordinary CPP optimizing problems, \( \nu_Q = \max_x f(X, Q) \).

Notice that these \( \nu_Q \)'s are identical to the \( \nu_Q \)'s of the maximax criterion. This problem can be transformed into linear optimization problem (cf. with maximin criterion).

\[
\min \nu
\]

such that

\[
\nu \geq \nu_Q - f(X, Q) \quad \forall Q \in V'\cap Q_0
\]

Thus, the minimax regret criterion can be presented as follows:
1. \( \forall Q \in V_Q^\gamma \):  

\[
\nu_Q = \max \left( \sum_{i \in SP} q_{path}(s_i) \cdot V_{\gamma}^{\gamma}(X) - k \cdot \sum_{i \in SP} q_{path}(s_i) \cdot \Delta V_{\gamma}^{\gamma} \right),
\]

such that  
(1.8)  

2. \( \min \nu \)  

such that  

\[
\nu \geq \nu_Q - \left( \sum_{i \in SP} q_{path}(s_i) \cdot V_{\gamma}(X) - k \cdot \sum_{i \in SP} q_{path}(s_i) \cdot \Delta V_{\gamma}^{\gamma} \right) \forall Q \in V_Q^\gamma
\]

(1.2)  

(3)  

\[
V_{\gamma}(X) - EV(X, Q) - \Delta V_{\gamma}^{\gamma}(Q) + \Delta V_{\gamma}^{\gamma}(Q) = 0 \quad \forall s_i \in SP \quad \forall Q \in V_Q^\gamma
\]

(4.8)  

5 Illustrative Example  

In this section we give a numerical example that demonstrates the use of different decision rules and also the differences between them.  

Suppose that one has the possibility to undertake two projects, A and B, in which investments can be made in two stages. Both projects require and produce only monetary resources and any surplus that is not invested can be deposited with 10\% risk-free interest rate. The total budget available is $10m.  

Both projects last three years and during the first two ones they only generate expenses. During the third year they may generate profit, contingent on the scenario occurring on that time.  

The scenario tree is shown in Figure 8. Notice that the conditional probabilities of the future scenarios are not point estimates, so we can not directly use CPP. The decision trees of projects A and B in Figures 8 and 9, respectively, illustrate the decisions that have to be
made. Thus, we have to decide whether to start the projects and whether to continue them after one year. The estimated project cash flows are also shown in decision trees.

**Figure 8.** Scenario tree of the example.

**Figure 9.** Decision tree of the project A of the example.
Using value 0.5 for the risk aversion coefficient $k$, we will determine the optimal portfolio using the four decision rules presented, maximax, maximin, expected and minimax regret.

5.1 Modeling Uncertainty

As shown in the scenario tree of Figure 8, we have the following additional information about the conditional probabilities of scenarios:

\[ D_{10} = \{ q_{x_1}(s_1 | s_{10}) \in [0.4, 0.6], q_{x_2}(s_2 | s_{10}) \in [0.4, 0.6] \} \]

\[ D_{11} = \{ q(s_1 | s_{11}) \in [0.3, 0.5], q(s_{12} | s_{11}) \in [0.1, 0.2], q(s_{13} | s_{11}) \in [0.35, 0.55] \} \]

\[ D_{20} = \{ q(s_2 | s_{20}) \in [0.5, 0.8], q(s_{22} | s_{20}) \in [0.2, 0.5] \} . \]

Thus, the sets $V^j(i)$ are
\[
V_1(r_0) = \left\{ \left( q_{r_0}(s_1|r_0), q_{r_0}(s_2|r_0) \right) \mid q_{r_0}(s_1|r_0) \in [0.4,0.6], q_{r_0}(s_2|r_0) \in [0.4,0.6], \right\}
\]

\[
V_1(r_1) = \left\{ \left( q_{r_1}(s_{11}|r_1), q_{r_1}(s_{12}|r_1), q_{r_1}(s_{13}|r_1) \right) \mid q_{r_1}(s_{11}|r_1) \in [0.3,0.5], q_{r_1}(s_{12}|r_1) \in [0.1,0.2], q_{r_1}(s_{13}|r_1) \in [0.35,0.55], q_{r_1}(s_{11}|r_1) + q_{r_1}(s_{12}|r_1) + q_{r_1}(s_{13}|r_1) = 1 \right\}
\]

\[
V_1(r_2) = \left\{ \left( q_{r_2}(s_{21}|r_2), q_{r_2}(s_{22}|r_2) \right) \mid q_{r_2}(s_{21}|r_2) \in [0.5,0.8], q_{r_2}(s_{22}|r_2) \in [0.2,0.5], q_{r_2}(s_{21}|r_2) + q_{r_2}(s_{22}|r_2) = 1 \right\}
\]

The sets of the extreme points of the sets \( V_1(r) \) can be easily calculated to be

\[
V_1^{\text{ext}}(r_0) = \{(0.6,0.4), (0.4,0.6)\}
\]

\[
V_1^{\text{ext}}(r_1) = \{(0.5,0.15,0.35),(0.5,0.1,0.4),(0.3,0.2,0.5),(0.45,0.2,0.35),(0.3,0.15,0.55),(0.35,0.1,0.55)\}
\]

\[
V_1^{\text{ext}}(r_2) = \{(0.5,0.5),(0.8,0.2)\}.
\]

**Table 4. Elements of the Cartesian product of the sets \( V_1^{\text{ext}}(r) \).**

|   | \( q(s_1|r_0) \) | \( q(s_2|r_0) \) | \( q(s_{11}|r_1) \) | \( q(s_{12}|r_1) \) | \( q(s_{13}|r_1) \) | \( q(s_{21}|r_2) \) | \( q(s_{22}|r_2) \) |
|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | 0.6           | 0.4           | 0.5           | 0.15          | 0.35          | 0.5           | 0.5           |
| 2 | 0.6           | 0.4           | 0.5           | 0.1           | 0.4           | 0.5           | 0.5           |
| 3 | 0.6           | 0.4           | 0.3           | 0.2           | 0.5           | 0.5           | 0.5           |
| 4 | 0.6           | 0.4           | 0.45          | 0.2           | 0.35          | 0.5           | 0.5           |
| 5 | 0.6           | 0.4           | 0.3           | 0.15          | 0.55          | 0.5           | 0.5           |
| 6 | 0.6           | 0.4           | 0.35          | 0.1           | 0.55          | 0.5           | 0.5           |
| 7 | 0.6           | 0.4           | 0.5           | 0.15          | 0.35          | 0.8           | 0.2           |
| 8 | 0.6           | 0.4           | 0.5           | 0.1           | 0.4           | 0.8           | 0.2           |
| 9 | 0.6           | 0.4           | 0.3           | 0.2           | 0.5           | 0.8           | 0.2           |
|10 | 0.6           | 0.4           | 0.45          | 0.2           | 0.35          | 0.8           | 0.2           |
|11 | 0.6           | 0.4           | 0.3           | 0.15          | 0.55          | 0.8           | 0.2           |
|12 | 0.6           | 0.4           | 0.35          | 0.1           | 0.55          | 0.8           | 0.2           |
|13 | 0.4           | 0.6           | 0.5           | 0.15          | 0.35          | 0.5           | 0.5           |
|14 | 0.4           | 0.6           | 0.5           | 0.1           | 0.4           | 0.5           | 0.5           |
|15 | 0.4           | 0.6           | 0.3           | 0.2           | 0.5           | 0.5           | 0.5           |
|16 | 0.4           | 0.6           | 0.45          | 0.2           | 0.35          | 0.5           | 0.5           |
|17 | 0.4           | 0.6           | 0.3           | 0.15          | 0.55          | 0.5           | 0.5           |
|18 | 0.4           | 0.6           | 0.35          | 0.1           | 0.55          | 0.5           | 0.5           |
|19 | 0.4           | 0.6           | 0.5           | 0.15          | 0.35          | 0.8           | 0.2           |
|20 | 0.4           | 0.6           | 0.5           | 0.1           | 0.4           | 0.8           | 0.2           |
|21 | 0.4           | 0.6           | 0.3           | 0.2           | 0.5           | 0.8           | 0.2           |
|22 | 0.4           | 0.6           | 0.45          | 0.2           | 0.35          | 0.8           | 0.2           |
|23 | 0.4           | 0.6           | 0.3           | 0.15          | 0.55          | 0.8           | 0.2           |
|24 | 0.4           | 0.6           | 0.35          | 0.1           | 0.55          | 0.8           | 0.2           |
For the application of maximin, expected and minimax regret criterion, we need the set of extreme points of the set of feasible \( Q \)'s, i.e., the set \( V'_Q \). As explained in section 3.4.2, we get this set with the help of cartesian product of the sets \( V'_p(q) \). The 24 elements of the cartesian product are presented in the rows of Table 4, and the respective non-conditional probabilities (elements of \( V'_Q \)) in the rows of Table 5.

**Table 5. Non-conditional probabilities corresponding to the probabilities of Table 4.**

<table>
<thead>
<tr>
<th>#</th>
<th>( q(r_1) )</th>
<th>( q(r_2) )</th>
<th>( q(r_{11}) )</th>
<th>( q(r_{12}) )</th>
<th>( q(r_{13}) )</th>
<th>( q(r_{21}) )</th>
<th>( q(r_{22}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6</td>
<td>0.4</td>
<td>0.3</td>
<td>0.09</td>
<td>0.21</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td>0.4</td>
<td>0.3</td>
<td>0.09</td>
<td>0.24</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>0.6</td>
<td>0.4</td>
<td>0.18</td>
<td>0.12</td>
<td>0.3</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td>0.4</td>
<td>0.27</td>
<td>0.12</td>
<td>0.21</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>5</td>
<td>0.6</td>
<td>0.4</td>
<td>0.18</td>
<td>0.09</td>
<td>0.33</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td>0.4</td>
<td>0.21</td>
<td>0.06</td>
<td>0.33</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>7</td>
<td>0.6</td>
<td>0.4</td>
<td>0.3</td>
<td>0.09</td>
<td>0.21</td>
<td>0.32</td>
<td>0.08</td>
</tr>
<tr>
<td>8</td>
<td>0.6</td>
<td>0.4</td>
<td>0.3</td>
<td>0.06</td>
<td>0.24</td>
<td>0.32</td>
<td>0.08</td>
</tr>
<tr>
<td>9</td>
<td>0.6</td>
<td>0.4</td>
<td>0.18</td>
<td>0.12</td>
<td>0.3</td>
<td>0.32</td>
<td>0.08</td>
</tr>
<tr>
<td>10</td>
<td>0.6</td>
<td>0.4</td>
<td>0.27</td>
<td>0.12</td>
<td>0.21</td>
<td>0.32</td>
<td>0.08</td>
</tr>
<tr>
<td>11</td>
<td>0.6</td>
<td>0.4</td>
<td>0.18</td>
<td>0.09</td>
<td>0.33</td>
<td>0.32</td>
<td>0.08</td>
</tr>
<tr>
<td>12</td>
<td>0.6</td>
<td>0.4</td>
<td>0.21</td>
<td>0.06</td>
<td>0.33</td>
<td>0.32</td>
<td>0.08</td>
</tr>
<tr>
<td>13</td>
<td>0.4</td>
<td>0.6</td>
<td>0.2</td>
<td>0.06</td>
<td>0.14</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>14</td>
<td>0.4</td>
<td>0.6</td>
<td>0.2</td>
<td>0.04</td>
<td>0.16</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>15</td>
<td>0.4</td>
<td>0.6</td>
<td>0.12</td>
<td>0.08</td>
<td>0.2</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>16</td>
<td>0.4</td>
<td>0.6</td>
<td>0.18</td>
<td>0.08</td>
<td>0.14</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>17</td>
<td>0.4</td>
<td>0.6</td>
<td>0.12</td>
<td>0.06</td>
<td>0.22</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>18</td>
<td>0.4</td>
<td>0.6</td>
<td>0.14</td>
<td>0.04</td>
<td>0.22</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>19</td>
<td>0.4</td>
<td>0.6</td>
<td>0.2</td>
<td>0.06</td>
<td>0.14</td>
<td>0.48</td>
<td>0.12</td>
</tr>
<tr>
<td>20</td>
<td>0.4</td>
<td>0.6</td>
<td>0.2</td>
<td>0.04</td>
<td>0.16</td>
<td>0.48</td>
<td>0.12</td>
</tr>
<tr>
<td>21</td>
<td>0.4</td>
<td>0.6</td>
<td>0.12</td>
<td>0.08</td>
<td>0.2</td>
<td>0.48</td>
<td>0.12</td>
</tr>
<tr>
<td>22</td>
<td>0.4</td>
<td>0.6</td>
<td>0.18</td>
<td>0.08</td>
<td>0.14</td>
<td>0.48</td>
<td>0.12</td>
</tr>
<tr>
<td>23</td>
<td>0.4</td>
<td>0.6</td>
<td>0.12</td>
<td>0.06</td>
<td>0.22</td>
<td>0.48</td>
<td>0.12</td>
</tr>
<tr>
<td>24</td>
<td>0.4</td>
<td>0.6</td>
<td>0.14</td>
<td>0.04</td>
<td>0.22</td>
<td>0.48</td>
<td>0.12</td>
</tr>
</tbody>
</table>

### 5.2 CPP Objective Function and Constraints

The CPP objective function for the problem is

\[
f(X,Q) = EV(X,Q) - 0.5 \cdot (q_{path}(\phi_{11}) \cdot \Delta V_{\phi_{11}}^- + q_{path}(\phi_{12}) \cdot \Delta V_{\phi_{12}}^- + q_{path}(\phi_{13}) \cdot \Delta V_{\phi_{13}}^- + q_{path}(\phi_{21}) \cdot \Delta V_{\phi_{21}}^- + q_{path}(\phi_{22}) \cdot \Delta V_{\phi_{22}}^-),
\]

where \( EV(X,Q) \) denotes the expected net present value,

\[
EV(X,Q) = q_{path}(\phi_{11}) \cdot V_{\phi_{11}}(X) + q_{path}(\phi_{12}) \cdot V_{\phi_{12}}(X) + q_{path}(\phi_{13}) \cdot V_{\phi_{13}}(X) + q_{path}(\phi_{21}) \cdot V_{\phi_{21}}(X) + q_{path}(\phi_{22}) \cdot V_{\phi_{22}}(X).
\]
Values over the scenario paths can be calculated as

\[
V_{\theta_1}(X) = -2 \cdot X_{ABY} - 1 \cdot X_{BY} + 1/1.1 \cdot \left( -2 \cdot X_{ACY} - 2 \cdot X_{BCY} \right) + \\
+ 1/1.1^2 \cdot 10 \cdot X_{ACY}
\]

\[
V_{\theta_2}(X) = -2 \cdot X_{ABY} - 1 \cdot X_{BY} + 1/1.1 \cdot \left( -2 \cdot X_{ACY} - 2 \cdot X_{BCY} \right) + \\
+ 1/1.1^2 \cdot \left( 7 \cdot X_{ACY} + 2 \cdot X_{BCY} \right)
\]

\[
V_{\theta_3}(X) = -2 \cdot X_{ABY} - 1 \cdot X_{BY} + 1/1.1 \cdot \left( -2 \cdot X_{ACY} - 2 \cdot X_{BCY} \right) + \\
+ 1/1.1^2 \cdot \left( 2 \cdot X_{ACY} + 3 \cdot X_{BCY} \right)
\]

\[
V_{\theta_4}(X) = -2 \cdot X_{ABY} - 1 \cdot X_{BY} + 1/1.1 \cdot \left( -2 \cdot X_{ACY} - 2 \cdot X_{BCY} \right) + \\
+ 1/1.1^2 \cdot \left( 5 \cdot X_{ACY} + 8 \cdot X_{BCY} \right)
\]

\[
V_{\theta_5}(X) = -2 \cdot X_{ABY} - 1 \cdot X_{BY} + 1/1.1 \cdot \left( -2 \cdot X_{ACY} - 2 \cdot X_{BCY} \right) + \\
+ 1/1.1^2 \cdot 12 \cdot X_{BCY}
\]

The subindices of the scenario paths \( \theta \) refer to the last scenario of that path.

The deviation constraints can be expressed as

\[
V_{\theta_1}(X) - EV(X, Q) - \Delta V^+_{\theta_1} + \Delta V^-_{\theta_1} = 0
\]

\[
V_{\theta_2}(X) - EV(X, Q) - \Delta V^+_{\theta_2} + \Delta V^-_{\theta_2} = 0
\]

\[
V_{\theta_3}(X) - EV(X, Q) - \Delta V^+_{\theta_3} + \Delta V^-_{\theta_3} = 0
\]

\[
V_{\theta_4}(X) - EV(X, Q) - \Delta V^+_{\theta_4} + \Delta V^-_{\theta_4} = 0
\]

\[
V_{\theta_5}(X) - EV(X, Q) - \Delta V^+_{\theta_5} + \Delta V^-_{\theta_5} = 0
\]

When applying the decision rules, we calculate the CPP objective function values associated with the elements of the set \( V_Q \). For example, with respect to the first element of \( V_Q \), the CPP objective function can be written as

\[
f(X, Q_1) = EV(X, Q_1) - 0.5 \cdot \left( 0.3 \cdot \Delta V^+_{\theta_1}(Q_1) + 0.09 \cdot \Delta V^-_{\theta_2}(Q_1) + 0.11 \cdot \Delta V^+_{\theta_1}(Q_1) + \\
0.2 \cdot \Delta V^+_{\theta_3}(Q_1) + 0.2 \cdot \Delta V^-_{\theta_2}(Q_1) \right).
\]

\( EV(X, Q_1) \) denotes the expected net present value when the scenario probabilities are \( Q_1 \),

32
\[ EV(\mathbf{X}, \mathbf{Q}_i) = 0.3 \cdot V_{\varphi_1}(\mathbf{X}) + 0.09 \cdot V_{\varphi_2}(\mathbf{X}) + 0.11 \cdot V_{\varphi_3}(\mathbf{X}) + 0.2 \cdot V_{\varphi_4}(\mathbf{X}) + 0.2 \cdot V_{\varphi_5}(\mathbf{X}). \]

The deviation variables for \( f(\mathbf{X}, \mathbf{Q}_i) \) can be calculated with constraints

\[ V_{\varphi_1}(\mathbf{X}) - EV(\mathbf{X}, \mathbf{Q}_i) - \Delta V^{+}_{\varphi_1}(\mathbf{Q}_i) + \Delta V^{-}_{\varphi_1}(\mathbf{Q}_i) = 0 \]
\[ V_{\varphi_2}(\mathbf{X}) - EV(\mathbf{X}, \mathbf{Q}_i) - \Delta V^{+}_{\varphi_2}(\mathbf{Q}_i) + \Delta V^{-}_{\varphi_2}(\mathbf{Q}_i) = 0 \]
\[ V_{\varphi_3}(\mathbf{X}) - EV(\mathbf{X}, \mathbf{Q}_i) - \Delta V^{+}_{\varphi_3}(\mathbf{Q}_i) + \Delta V^{-}_{\varphi_3}(\mathbf{Q}_i) = 0 \]
\[ V_{\varphi_4}(\mathbf{X}) - EV(\mathbf{X}, \mathbf{Q}_i) - \Delta V^{+}_{\varphi_4}(\mathbf{Q}_i) + \Delta V^{-}_{\varphi_4}(\mathbf{Q}_i) = 0 \]
\[ V_{\varphi_5}(\mathbf{X}) - EV(\mathbf{X}, \mathbf{Q}_i) - \Delta V^{+}_{\varphi_5}(\mathbf{Q}_i) + \Delta V^{-}_{\varphi_5}(\mathbf{Q}_i) = 0. \]

Decision tree constraints of the problem can be expressed as

\[ X_{ASY} + X_{ABN} = 1 \]
\[ X_{ACY_1} + X_{ACN_1} = X_{ASY} \]
\[ X_{ACY_2} + X_{ACN_2} = X_{ASY} \]
\[ X_{RSY} + X_{RSN} = 1 \]
\[ X_{BCY_1} + X_{BCN_1} = X_{RSY} \]
\[ X_{BCY_2} + X_{BCN_2} = X_{RSY} \]

and the resource constraints as

\[ -2 \cdot X_{ASY} - 1 \cdot X_{RSY} + 10 - R S_{t_0} = 0 \]
\[ -2 \cdot X_{ACY_1} - 2 \cdot X_{BCY_1} + 1.1 \cdot R S_{t_0} - R S_{t_1} = 0 \]
\[ -2 \cdot X_{ACY_2} - 2 \cdot X_{BCY_2} + 1.1 \cdot R S_{t_0} - R S_{t_2} = 0 \]
\[ 10 \cdot X_{ACY_1} + 1.1 \cdot R S_{t_1} - R S_{t_11} = 0 \]
\[ 7 \cdot X_{ACY_1} + 2 \cdot X_{BCY_1} + 1.1 \cdot R S_{t_1} - R S_{t_2} = 0 \]

33
2 \cdot X_{ACY_1} + 3 \cdot X_{BCY_1} + 1.1 \cdot RS_{t_1} - RS_{t_3} = 0

5 \cdot X_{ACY_2} + 8 \cdot X_{BCY_2} + 1.1 \cdot RS_{t_2} - RS_{t_3} = 0

12 \cdot X_{BCY_2} + 1.1 \cdot RS_{t_2} - RS_{t_3} = 0.

Decision variables \( X \)'s, \( RS \)'s and \( \Delta V \)'s are constrained as follows:

All \( X \)'s are binary

All \( RS \)'s are continuous and non-negative

All \( \Delta V \)'s are continuous and non-negative

### 5.3 Applying Decision Rules

#### 5.3.1 Maximax

The objective is to find the portfolio strategy, for which maximum CPP objective function value is greatest over the set of feasible scenario probabilities, i.e. to maximize the objective function value over the sets of feasible portfolio strategies and feasible scenario probabilities,

\[
\max_{X, Q} f(X, Q).
\]

Additional decision variables \( q(s) \)'s are constrained as follows

\[
q(s_{11}) = 1
\]

\[
q(s_1) \in [0.4, 0.6], \quad q(s_2) \in [0.4, 0.6]
\]

\[
q(s_{11}) \in [0.3q(s_1), 0.5q(s_1)], \quad q(s_{12}) \in [0.1q(s_1), 0.2q(s_1)], \quad q(s_{13}) \in [0.35q(s_1), 0.55q(s_1)]
\]

\[
q(s_{21}) \in [0.5q(s_2), 0.8q(s_2)], \quad q(s_{22}) \in [0.2q(s_2), 0.5q(s_2)]
\]

\[
q(s_1) + q(s_2) = q(s_{11})
\]

\[
q(s_{11}) + q(s_{12}) + q(s_{13}) = q(s_1)
\]

\[
q(s_{21}) + q(s_{22}) = q(s_2).
\]

When this model is solved, its found out that the highest possible CPP objective function value is $22.1m, and it is associated with scenario probabilities \( q(s_1)=0.4, q(s_2)=0.6, q(s_{11})=0.2, \)

34
\( q(s_{12}) = 0.06, \ q(s_{13}) = 0.14, \ q(s_{51}) = 0.48, \ q(s_{52}) = 0.12. \) Maximax decision rule also suggests the initiation of both projects, but one should not continue project B if scenario \( s_1 \) occurs. That is, action variables \( X_{ASY}, X_{A\overline{CY}1}, X_{A\overline{CY}2}, X_{R\overline{SY}}, X_{R\overline{CN}1} \) and \( X_{R\overline{CY}2} \) are ones and all the others are zeros. For clarity, we denote this strategy by \( X_p. \)

5.3.2 Maximin

The objective of maximin decision rule is to select the strategy which minimum CPP objective function value over the set of feasible \( Q \)'s is highest. Thus, the objective is to

\[
\text{max} \ v
\]

such that \( v \) is the smallest CPP objective function value associated with the selected strategy. That is, \( v \) must be smaller than the CPP objective function value associated with the selected strategy and any point of the set \( V_Q^\epsilon. \) Thus, we need 24 additional constraints,

\[
v \leq f(X, Q_i), \quad i = 1, \ldots, 24.
\]

Equivalently for the rest 23 elements. Note that the calculation of deviation variables \( \Delta V(Q) \)'s requires also \( 24 \cdot 5 = 120 \) constraints, as explained in previous subsection.

Additional decision variable \( v \) is constrained as follows,

\[ v \text{ is continuous.} \]

When this problem is solved, it is found out that only the project B should be initiated. Furthermore, if scenario \( s_1 \) occurs B should get cancelled. That is, action variables \( X_{ASY}, X_{R\overline{SY}}, X_{R\overline{CN}1} \) and \( X_{R\overline{CY}2} \) are ones and all the others are zeros. For clarity, we denote this strategy by \( X_p. \)

The minimum CPP objective function value associated with this strategy is \$5.3m, and it is associated with probabilities \( Q_{17} \cdot Q_{12}. \)

5.3.3 Expected

When applying expected decision rule, one tries to maximize the expected CPP objective function value, which is calculated over the set \( V_Q^\epsilon. \) That is, the object is to

\[
\text{max} \ \frac{1}{24} \sum_{i=1}^{24} f(X, Q_i).
\]
When this model is solved, it is found out that expected criterion suggests the same strategy as maximax criterion, i.e., to initiate both projects and cancel B if scenario \( s_1 \) occurs. The expected value of the CPP objective function is then $13.6m.

### 5.3.4 Minimax Regret

When applying minimax regret criterion one has to first solve ordinary CPP problem with respect to every element of the set \( V' \), i.e.

\[
\nu_{Q_i} = \max_x f(X, Q_i), \quad i = 1, \ldots, 24.
\]

After solving the 24 ordinary CPP optimizing problems, the object is to select strategy \( X \) such that the maximum regret associated with that strategy is as small as possible. This can be accomplished by solving the problem

\[
\min \nu
\]

such that

\[
\nu \geq \nu_{Q_i} - f(X, Q_i), \quad i = 1, \ldots, 24.
\]

Thus, one needs additional 24 constraints for \( \nu \) and 24\times5=120 constraints for the calculation of the deviation variables \( \Delta V(Q)'s \).

The maximums of ordinary CPP problems are presented in Table 6. The maximums are associated with either strategy \( X_a \) or strategy \( X_v \). Notice that Table 6 also includes all the information needed for the use of maximax criterion. The minimax regret criterion suggests the use of strategy \( X_v \). For this strategy, the maximum regret is $7.0m and it is associated with probability \( Q_5 \).

Thus, decision rules maximax, expected and minimax regret suggest the use of strategy \( X_v \).

However, the differences between strategies are pretty small. The expected value of strategy \( X_a \) can be calculated to be $12.6m ($13.6m for \( X_s \), the maximum regret $9.1m ($7.0m for \( X_s \)) and the maximum CPP objective function value $20.9m ($22.1m for \( X_s \)). Also for \( X_s \), the minimum CPP objective function value can be calculated to be $4.2m ($5.3 for \( X_s \)). Anyway, if there is no additional information, strategy \( X_v \) should be selected.

36
Table 6. CPP objective function maximums with respect to every element of $V_\mathcal{Q}^e$.

<table>
<thead>
<tr>
<th>$Q_i$</th>
<th>$\max f(X, Q_i)$</th>
<th>$\chi$</th>
<th>$Q_i$</th>
<th>$\max f(X, Q_i)$</th>
<th>$\chi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13.5 a</td>
<td></td>
<td>13</td>
<td>20.9 b</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>11.9 a</td>
<td></td>
<td>14</td>
<td>20.9 b</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8.0 b</td>
<td></td>
<td>15</td>
<td>20.9 b</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>12.8 a</td>
<td></td>
<td>16</td>
<td>20.9 b</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>8.0 b</td>
<td></td>
<td>17</td>
<td>20.9 b</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8.0 b</td>
<td></td>
<td>18</td>
<td>20.9 b</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>14.4 a</td>
<td></td>
<td>19</td>
<td>22.1 a</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>12.7 a</td>
<td></td>
<td>20</td>
<td>20.9 a</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6.6 a</td>
<td></td>
<td>21</td>
<td>16.3 a</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>13.6 a</td>
<td></td>
<td>22</td>
<td>21.5 a</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>5.3 b</td>
<td></td>
<td>23</td>
<td>16.2 b</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5.6 a</td>
<td></td>
<td>24</td>
<td>16.2 b</td>
<td></td>
</tr>
</tbody>
</table>

6 Discussion

In this section we compare the different decision rules and discuss the computational complexity of the presented methods. We also discuss different situations where proposed methods could be used.

6.1 Model Complexity

The optimizing problems of different decision rules have some common decision variables and constraints. The numbers of these constraints are presented in Table 7. The numbers of additional decision variables and constraints of the optimization problems of criterions maximax and expected, are presented in Tables 8 and 9, respectively. The optimization problems of criterions maximin and minimax regret have the same number of additional decision variables and constraints, which are presented in Table 10. The notation $|\cdot|$ stands for the number of elements in that set and $\mathcal{S}^e$ is the set of scenarios with child scenarios with uncertain conditional probabilities.

Note that the maximax criterion optimizing problem can be utilized also by solving $|V_\mathcal{Q}^e|$ ordinary CPP optimizing problems. The solving of these problems is also required before one can apply the minimax regret criterion.
Table 7. Decision variables and constraints common to optimization problems of all decision rules.

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>Number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action variables (X’s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource surplus variables (R’s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision tree constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional constraints</td>
<td>O</td>
<td></td>
</tr>
</tbody>
</table>

Table 8. Additional decision variables and constraints of maximax criterion.

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>Number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deviation variables ((\lambda)’s)</td>
<td>2 (</td>
<td>SP</td>
</tr>
<tr>
<td>Probabilities ((q)’s)</td>
<td>(</td>
<td>S</td>
</tr>
<tr>
<td>Constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deviation constraints</td>
<td>(</td>
<td>SP</td>
</tr>
<tr>
<td>Probability constraints</td>
<td>(</td>
<td>S</td>
</tr>
<tr>
<td>Optional probability constraints</td>
<td>(P) typically (P &gt;</td>
<td>S^*</td>
</tr>
</tbody>
</table>

Table 9. Additional decision variables and constraints of expected criterion.

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>Number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deviation variables ((\lambda)’s)</td>
<td>2 (</td>
<td>SP</td>
</tr>
<tr>
<td>Constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deviation constraints</td>
<td>(</td>
<td>SP</td>
</tr>
</tbody>
</table>

Table 10. Additional decision variables and constraints of maximin and minimax regret criterions.

<table>
<thead>
<tr>
<th>Decision variables</th>
<th>Number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Deviation variables ((\lambda)’s)</td>
<td>2 (</td>
<td>SP</td>
</tr>
<tr>
<td>Constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deviation constraints</td>
<td>(</td>
<td>SP</td>
</tr>
<tr>
<td>Constraints for (r)</td>
<td>(</td>
<td>V^\omega Q</td>
</tr>
</tbody>
</table>

As we can see, the complexity of proposed methods, except from maximax criterion, depends linearly on the number of elements in \(V^\omega Q\). This number, in turn, depends heavily on the number of elements in set \(S^*\). If scenario \(s\) has child scenarios with uncertain conditional probabilities, then the set \(V^\omega (\hat{S})\) contains at least two elements. We can calculate the size of the set \(V^\omega Q\) as a product of the sizes of the sets \(V^\omega (\hat{S})\)’s,
\[ |V_Q^\epsilon| = \prod_{r \in \mathcal{S}} |V_Q^\epsilon(r)| \leq \prod_{r \in \mathcal{S}} 2^{1^r} = 2^{|\mathcal{S}|}. \]

Thus, the size of the set \( V_Q^\epsilon \), and also the complexity of the methods maximin, expected and minimax regret depends exponentially on the size of the set \( \mathcal{S}^\epsilon \). However, the number of constraints in maximax method depends only linearly on the size of the set \( \mathcal{S}^\epsilon \). Thus, maximax criterion works also with larger models.

### 6.2 Comparison of Decision Rules

As we saw in previous section, different decision rules give different strategies. Generally speaking, maximax criterion gives risky strategies and maximin criterion gives strategies that are risk-averse. Expected and especially the minimax regret criterions give more robust strategies.

It's easy to construct a model where the maximax strategy performs perfectly with some probabilities \( Q \), but poorly with most probabilities. Thus, maximax rule should never be the only decision rule used. Also, maximin criterion gives strategies that never perform poorly, but there may exist strategies that perform much better in most situations.

Because maximax and maximin rules may give such extreme strategies, we recommend the minimax regret decision rule. Also the expected criterion is advisable, if it's presumptions (see section 4.4) are fulfilled.

### 6.3 Utilization of Presented Methods

It's clear that the presented methods can't be used to solve problems with tens of uncertain probabilities, because the size of the optimizing problems grows exponentially. However, methods are completely practical if the model used includes only a few scenarios with uncertain conditional probabilities.

For instance, methods could be useful with a problem, where two or more experts are used to assess the probabilities of future scenarios. If there are some scenarios which probabilities the experts don't agree about, the use of weighted averages or some other kind of compromises may lead to wrong estimates. Thus, it might be wiser and easier to use probability intervals. For example, minimax regret criterion could be used to get a strategy that works well despite of the actual probabilities.
Methods could be also used in sensitivity analysis. For the selected strategy $X^*$ one can calculate the maximum regret $\max_{x,Q} [f(X,Q) - f(X^*,Q)]$, or the minimum possible CPP objective function value, $\min_Q f(X^*,Q)$. Both optimization problems need additional nonlinear constraints $\Delta V^+ \cdot \Delta V^- = 0 \forall \Delta V$, because otherwise the solutions would not converge. The problems are almost similar to the maximax problem presented, and can thus be applied also to larger models with many uncertain probabilities.

7 Conclusion

The contingent portfolio programming method presented by Gustafsson and Salo (2001a) addresses many of the limitations of the earlier project portfolio selection methods. However, the need of exact probabilities of future scenarios is a requirement that hinders the use of this method. In this paper we presented methods to overcome this requirement by allowing the use of probability intervals instead of point estimates.

Specifically, there are four main findings in this paper. Firstly, it was demonstrated how probability uncertainties can be divided into time dependent and time independent components.

Secondly, we demonstrated how time dependent uncertainty can be included in CPP models. This was done through the use of sibling scenarios.

Thirdly, we presented a way to model time independent uncertainties. It was assumed that instead of point estimates, one has a set of feasible conditional probabilities for child scenarios of every scenario. Only requirement for these sets is that they have to be linearly constrained.

Finally, we demonstrated how different decision rules could be used in selecting the best portfolio under uncertainties of probabilities. The presented methods also offer tools for sensitivity analysis. However, the methods face two major drawbacks. Firstly, the size of the optimizing problems of maximin, expected and minimax regret criterions grows exponentially as the function of uncertainties. Secondly, the theoretical basis of the methods maximin and minimax regret is not solid, because we could not prove the quasiconvexity of
the CPP objective function. However, we could neither come up with a sample function that is not quasiconvex.

The voluminous use of probability intervals in CPP models requires still some theoretical work, at least developing of more efficient ways to apply the decision rules. This may also require studying the quasiconvexity of the objective function. One interesting issue is also the relationship between the \( k \) coefficient and different decision rules. If one do not consider this, it may lead to double-counting the risks. On the practical side, the use of presented methods demands specific software.

**Appendix**

**Convexity of the Set of Feasible Scenario Probabilities**

**Lemma.**

The set \( V_Q \) of all feasible \( Q \)'s is convex.

**Proof.**

We prove the lemma in the case of the three optional constraints introduced in previous subsection. Proof with arbitrary \( D' \) (see, e.g., Taha 1977) would require additional notation that is out of the scope of this paper.

Define \( Q' \) as the convex combination of two distinct points, \( Q' \) and \( Q'' \), in \( V_Q \), i.e.

\[
Q' = \lambda Q' + (1 - \lambda) Q'' , \quad 0 \leq \lambda \leq 1.
\]

Then, \( V_Q \) is convex if and only if \( Q' \) also lies in \( V_Q \). For this to be true we have to show that \( q' \) satisfies the following constraints if \( q' \) and \( q'' \) satisfy them. Remember that the constraints (3-5) are optional, i.e., not every scenario has to satisfy them.

1. \( q(\lambda) \in [0, q(s)], \lambda \in C(s), \)

2. \( \sum_{\lambda \in C(s)} q(\lambda) = q(s), \)

3. \( q(\lambda) = a \cdot q(s), \lambda \in C(s), \)

4. \( q(\lambda) \in [b \cdot q(s), c \cdot q(s)], \lambda \in C(s), \) and
(5) \( q(x_1) \geq d \cdot q(x_2) + e \cdot q(x_3), \) \( B(x_1) = B(x_2) = B(x_3), \)

where \( a, b, c, d \) and \( e \) are positive constants. The constraint (1) is included in (4) with \( b=0 \) and \( c=1 \), and also the constraint (3) is included in (4) with \( b=c=a \). Thus, it’s enough to show that (2), (4) and (5) applies.

2. For arbitrary \( s \),
\[
\sum_{x \in \mathcal{C}(t)} q^*(x) = \sum_{x \in \mathcal{C}(t)} \left[ \lambda q'(x) + (1 - \lambda)q''(x) \right]
= \lambda \sum_{x \in \mathcal{C}(t)} q'(x) + (1 - \lambda) \sum_{x \in \mathcal{C}(t)} q''(x) = \lambda q'(s) + (1 - \lambda)q''(s) = q^*(s).
\]

4. For arbitrary \( x \),
\[
q^*(x) = \lambda q'(x) + (1 - \lambda)q''(x) \in \lambda \cdot \left[ b \cdot q'(s), c \cdot q''(s) \right] + (1 - \lambda) \cdot \left[ b \cdot q''(s), c \cdot q''(s) \right]
= \left[ b \cdot (\lambda q'(s) + (1 - \lambda)q''(s)), c \cdot (\lambda q'(s) + (1 - \lambda)q''(s)) \right] = \left[ b \cdot q^*(s), c \cdot q^*(s) \right]
\]

5. For arbitrary \( x_1, x_2 \) and \( x_3 \) with the same parent scenario,
\[
q^*(x_1) = \lambda q'(x_1) + (1 - \lambda)q''(x_1)
\geq \lambda \cdot \left( d \cdot q'(x_2) + e \cdot q'(x_3) \right) + (1 - \lambda) \left( d \cdot q''(x_2) + e \cdot q''(x_3) \right)
= d \cdot (\lambda q'(x_2) + (1 - \lambda)q''(x_2)) + e \cdot (\lambda q'(x_3) + (1 - \lambda)q''(x_3))
= d \cdot q^*(x_2) + e \cdot q^*(x_3).
\]

**Extreme Points of the Set of Feasible Scenario Probabilities**

**Lemma.**

The sets \( V_Q^\varepsilon \) and \( V_Q^\varepsilon = \prod_{s \in S} V_C^\varepsilon(s) \) are identical in the sense that every element of \( V_Q^\varepsilon \) can be transformed to an element of \( V_Q^\varepsilon \) and vice versa.

**Proof.**

To clarify the presentation, we denote an element of \( V_C^\varepsilon(s) \) by
\[
q_i = \left( q_{i_1}(s_1), q_{i_2}(s_2), \ldots, q_{i_{|S|}}(s_{|S|}) \right)^T, \text{ an element of } V_C^\varepsilon \text{ by}
\]
\[
Q = \prod_{s \in S} q_i, \quad q_s \in V_C^\varepsilon(s) \forall s \in S, \text{ and an element of } V_Q \text{ by } Q.
\]

Every element \( Q \) of the set \( V_Q^\varepsilon \) equals to an element \( Q \) of the set \( V_Q^\varepsilon \).
Assume that we have the probabilities $(q(s_0),\ldots,q(s_j))^\top$ of the scenarios from $s_0$ to some scenario $s_j$ such that these probabilities are extreme. If we take the conditional probabilities of any point $q_i \in V_{c}^\epsilon (s_j)$, we get the probabilities $(q(s_0),\ldots,q(s_j),q(x_1),\ldots,q(x_{\text{last}}))^\top$ of the scenarios from $s_i$ to $s_j$'s child scenarios ($x$'s). It is clear that these probabilities are also extreme. Beginning from the base scenario and following the presented procedure, we can construct vector $Q$ by using any combination of the extreme $q_i$'s. Thus, any combination of the extreme $q_i$'s equals to some element of $V_Q^\epsilon$.

Every element $Q$ of the set $V_Q^\epsilon$ equals to an element $Q_e$ of the set $V_i^\epsilon$.

Proof by contradiction. Assume that we have probabilities $(q(s_0),\ldots,q(s_j),q(x_1),\ldots,q(x_{\text{last}}))^\top$ that are extreme. Assume also that $\{q(x_1),\ldots,q(x_{\text{last}})\} = \{q(s_j) \cdot q_i \cdot (x_1 | s_j),\ldots,q(s_j) \cdot q_i \cdot (x_{\text{last}} | s_j)\} = q(s_j) \cdot q_i$ such that $q_i \not\in V_{c}^\epsilon (s_j)$. Then, we can express $q_i$ as a convex combination of two distinct points in $V_{c}^\epsilon (s_j)$, $q_i = \lambda q'_i + (1-\lambda)q''_i$, $0 < \lambda < 1$, and the non-conditional probabilities as a convex combination of two distinct points,

$$(q(s_0),\ldots,q(s_j),q(s_j) \cdot q_i)^\top = \lambda \cdot (q(s_0),\ldots,q(s_j),q(s_j) \cdot q'_i)^\top + (1-\lambda) \cdot (q(s_0),\ldots,q(s_j),q(s_j) \cdot q''_i)^\top.$$  

Then, by definition, $(q(s_0),\ldots,q(s_j),q(x_1),\ldots,q(x_{\text{last}}))^\top$ is not an extreme point.

Combining the two results, we get that the sets $V_Q^\epsilon$ and $V_i^\epsilon$ are equal.

References


J. Gustafsson and A. Salo, Managing Risky Projects with Contingent Portfolio Programming, unpublished manuscript, 2001a.


