ENERGY TECHNOLOGY SYSTEMS ANALYSIS PROGRAMME

Energy Technology Systems Analysis Programme

TIMES Version 3.9 User Note

Stochastic Programming and Tradeoff Analysis in TIMES

Authors:

Richard Loulou
KANLO Consultants, France

Antti Lehtila
VTT, Finland

Updated May 2016

Foreword

This report contains the full documentation on the implementation and use of the Stochastic Programming and Tradeoff Analysis facilities of the TIMES model generator.

The report is divided in five chapters. After the general introduction in Chapter 1, Chapter 2 presents a brief description of the mathematical approach taken with respect to stochastic programming and Chapter 3 the approach used for tradeoff analysis. Chapter 4 contains the description of the GAMS implementation of the new elements, along with the sets, parameters, variables, and equations that have been added to the TIMES model. Finally, Chapter 5 summarizes the usage notes in the form of a brief User's Manual for stochastic programming and tradeoff analysis in TIMES.

This document is a supplement to the main documentation of the TIMES model generator (Parts I–V), available on the IEA-ETSAP website.

Table of contents

1. INTRODUCTION

Stochastic Programming is a method for making optimal decisions under risk. The risk consists of uncertainty regarding the values of some (or all) of the LP parameters (cost coefficients, matrix coefficients, RHSs). Each uncertain parameter is considered to be a random variable, usually with a discrete, known probability distribution. The objective function thus becomes also a random variable and a criterion must be chosen in order to make the optimization possible. Such a criterion may be expected cost, expected utility, etc., as mentioned by Kanudia and Loulou (1998).

Uncertainty on a given parameter is said to be resolved, either fully or partially, at the *resolution time*, i.e. the time at which the actual value of the parameter is revealed. Different parameters may have different times of resolution. Both the resolution times and the probability distributions of the parameters may be represented on an event tree, such as the one of Figure 1, depicting a typical energy/environmental situation. In Figure 1, two parameters are uncertain: mitigation level, and demand growth rate. The first may have only two values (High and Low), and becomes known in 2010. The second also may have two values (High and Low) and becomes known in 2020. The probabilities of the outcomes are shown along the branches. This example assumes that present time is 2000. This example is said to have three stages (i.e. two resolution times). The simplest non-trivial event tree has only two stages (a single resolution time).

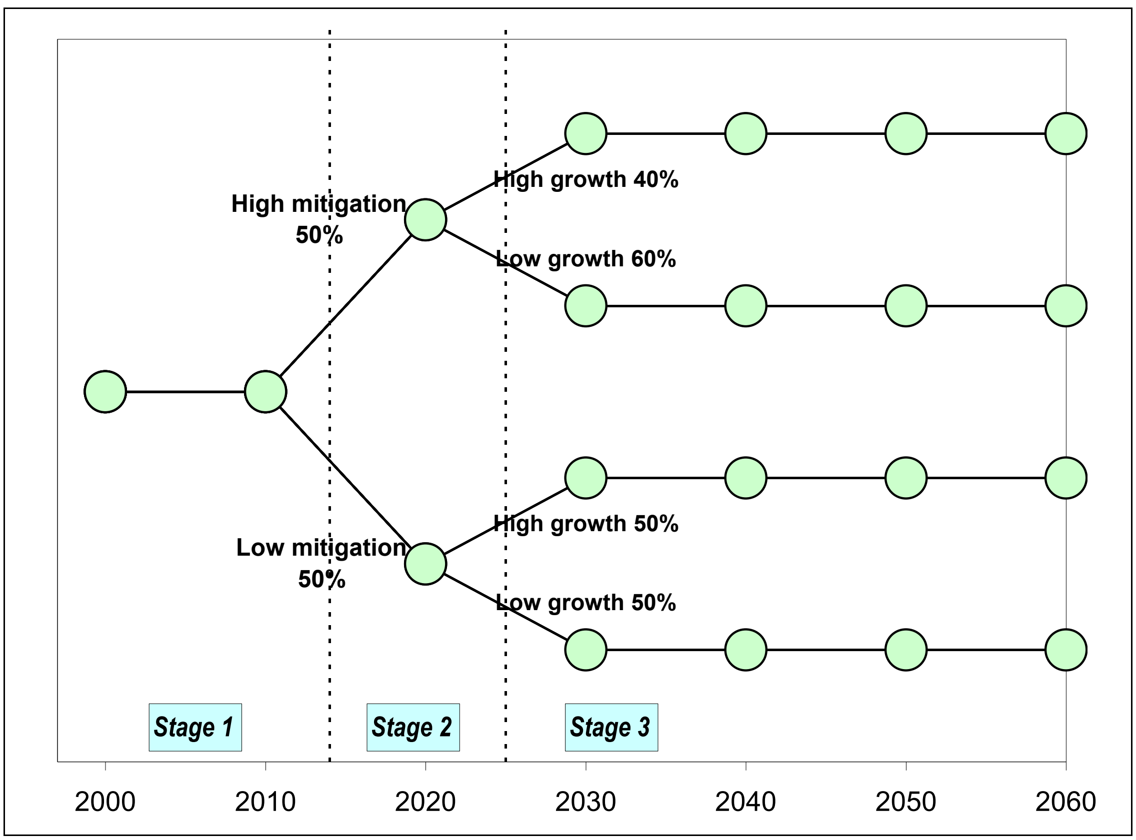

Figure 1. Event Tree for a three-stage stochastic TIMES example.

While stochastic programming is an advanced way to take into account uncertainties, a more common and very useful way to analyze the impact of uncertainties is ***sensitivity analysis***. In sensitivity analysis, the values of some important exogenous assumptions are varied, and a series of model runs is performed over a discrete set of combinations of these assumptions. Sensitivity analysis is often combined with ***tradeoff analysis***, where the tradeoff relation between several objectives is analyzed. The stochastic mode provides an efficient tool for both sensitivity and tradeoff analyses, because it enables the use of so-called uncertain attributes. The uncertain attributes are similar to the corresponding standard TIMES attributes, but they can be defined over a discrete set of states-of-the-world (SOW). In stochastic programming the SOWs correspond to the branches of the event tree, but they can equally well be used for sensitivity analysis, so that the model is sequentially run over the set of SOWs, using the corresponding values of the uncertain attributes in each individual run.

Figure 2 illustrates a few possible set-ups for sensitivity and tradeoff analyses in TIMES, all of which are supported by the model generator:

A. Simple sensitivity analysis over the set of SOWs.
B. Two-phase tradeoff analysis, where the model is first run once using a user-defined objective function, and then the solution from the first phase is used for defining additional constraints in a series of model runs in the second phase.
C. Two-phase tradeoff analysis, where the model is first run over a set of SOWs, each of which may have a different objective functions and different parameter attributes. In Phase 2 the solution for each SOW from Phase 1 is used for defining additional constraints for each SOW in Phase 2, where the standard objective function is used.
D. Two-phase tradeoff analysis, where the model is first run over a set of SOWs, each of which have a different objective function and optionally different UC RHS. In Phase 2 the solution for each SOW obtained from Phase 1 is used for defining an additional deviation constraint for each of the objectives used in Phase 1, and a single model is solved in Phase 2 optimizing the standard objective function.
E. Multiphase tradeoff analysis over N phases with different objective functions.

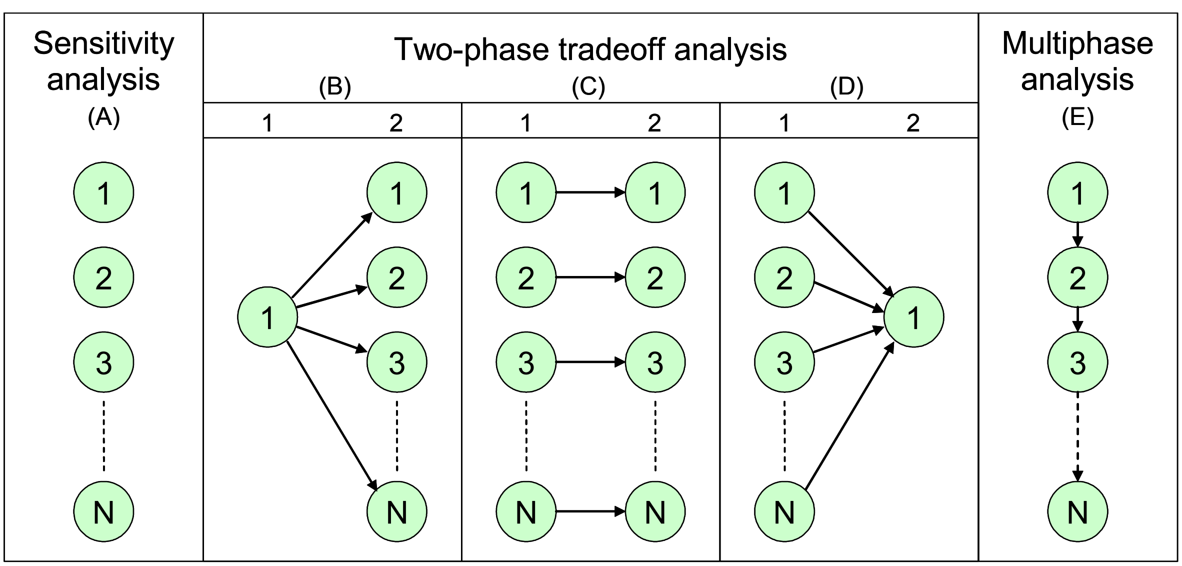

Figure 2. Possible set-ups for sensitivity / tradeoff analysis.

2. MULTI-STAGE STOCHASTIC PROGRAMMING

2.1 Problem formulation

In the context of stochastic programming, the key observation is that *prior to resolution time, the decision maker (and hence the model) does not know the eventual values of the random parameters, but still has to take decisions.* On the contrary, *after resolution, the decision maker knows with certainty the outcome of some event(s) and his decisions will be different depending of which outcome has occurred.*

For the example of Figure 1, in 2000 and 2010 there can be only one set of decisions, whereas in 2020 there will be two sets of decisions, contingent on which of the Mitigation outcomes (High or Low) has occurred, and in 2030, 2040, 2050 and 2060, there will be four sets of contingent decisions.

This remark leads directly to the following general multi-period, multi-stage stochastic program (1) – (3) below. The formulation described here is based on Dantzig (1955), Wets (1989), or Kanudia and Loulou (1998), and uses the expected cost criterion. Note that this is a LP, but its size is larger than that of the deterministic TIMES model.

Minimize

$$Z = \sum_{t \in T} \sum_{s \in S(t)} C(t,s) \times X(t,s) \times p(t,s) \tag{1}$$

Subject to:

$$A(t,s) \times X(t,s) \geq b(t,s) \qquad \forall t \in T, \forall s \in S(t) \tag{2}$$

$$\sum_{t \in T} D(t, g(t,s)) \times X(t, g(t,s)) \geq e(s) \qquad \forall s \in S(T) \tag{3}$$

where

t	=	time period
T	=	set of time periods
s	=	state index
$S(t)$	=	set of state indices for time period t; for Figure 1, we have: S(2000) = 1; S(2010) = 1; S(2020) = 1,2; S(2030) = 1,2,3,4; S(2040) = 1,2,3,4; S(2050) = 1,2,3,4; S(2060) = 1,2,3,4;
$S(T)$	=	set of state indices at the last stage (the set of *scenarios*). Set $S(T)$ is homeomorphic to the set of paths from period 1 to last period, in the event tree.
$g(t,s)$	=	a unique mapping from $\{(t,s) \mid s \in S(T)\}$ to $S(t)$, according to the event tree. $g(t,s)$ is the state at period t corresponding to scenario s.
$X(t,s)$	=	the column vector of decision variables in period t, under state s
$C(t,s)$	=	the cost row vector

$p(t,s)$	=	event probabilities
$A(t,s)$	=	the LP sub-matrix of single period constraints, in time period t, under state s
$b(t,s)$	=	the right hand side column vector (single period constraints) in time period t, under state s
$D(t,s)$	=	the LP sub-matrix of multi-period constraints under state s
$e(s)$	=	the right hand side column vector (multi-period constraints) under scenario s

Alternate formulation: The above formulation makes it a somewhat difficult to retrieve the strategies attached to the various scenarios. Moreover, the actual writing of the cumulative constraints (3) is a bit delicate. An alternate (but equivalent) formulation consists in defining one scenario per path from initial to terminal period, and to define distinct variables X(t,s) for each scenario and each time period. For instance, in this alternate formulation of the example, there would be four variables X(t,s) at every period t, (whereas there was only one variable X(2000,1) in the previous formulation).

Minimize

$$Z = \sum_{t \in T} \sum_{s \in S(t)} C(t,s) \times X(t,s) \times p(t,s) \qquad \textbf{(1')}$$

Subject to:

$$A(t,s) \times X(t,s) \geq b(t,s) \qquad \text{all t, all s} \qquad \textbf{(2')}$$

$$\sum_{t \in T} D(t,s) \times X(t,s) \geq e(s) \qquad \text{all t, all s} \qquad \textbf{(3')}$$

Of course, in this approach we need to add equality constraints to express the fact that some scenarios are identical at some periods. In the example of Figure 1, we would have:

X(2000,1)=X(2000,2)=X(2000,3)=X(2000,4),
X(2010,1)=X(2010,2)=X(2010,3)=X(2010,4),
X(2020,1)=X(2020,2),
X(2020,3)=X(2020,4).

Although this formulation is less parsimonious in terms of additional variables and constraints, many of these extra variables and constraints are in fact eliminated by the pre-processor of most optimizers. The main advantage of this new formulation is the ease of producing outputs organized by scenario.

In the current implementation of stochastic TIMES, the first approach has been used (Equations 1-3). The results are however reported for all scenarios in the same way as in the second approach.

In addition, in TIMES there is also an experimental variant for the modeling of recurring uncertainties with stochastic programming, described in Appendix A.

2.2 Alternative objective formulations

The preceding description of stochastic programming assumes that the policy maker accepts the expected cost as his optimizing criterion. This is equivalent to saying that he is risk neutral. In many situations, the assumption of risk neutrality is only an approximation of the true utility function of a decision maker.

Two alternative candidates for the objective function are:

- Expected utility criterion with linearized risk aversion
- Minimax Regret (Savage) criterion (Loulou and Kanudia, 1999[1])

Expected Utility Criterion with risk aversion

The first alternative has been implemented into the stochastic version of TIMES. This provides a feature for taking into account that a decision maker may be risk averse, by defining a new utility function to replace the expected cost.

The approach is based on the classical E-V model (an abbreviation for Expected Value-Variance). In the E-V approach, it is assumed that the variance of the cost is an acceptable measure of the risk attached to a strategy in the presence of uncertainty. The variance of the cost of a given strategy **k** is computed as follows:

$$Var(C_k) = \sum_j p_j \bullet (Cost_{j|k} - EC_k)^2$$

where $Cost_{j|k}$ is the cost when strategy k is followed and the j^{th} state of nature prevails, and EC_k is the expected cost of strategy k, defined as usual by:

$$EC_k = \sum_j p_j \bullet Cost_{j|k}$$

An E-V approach would thus replace the expected cost criterion by the following utility function to minimize:

$$U = EC + \lambda \cdot \sqrt{Var(C)}$$

where $\lambda > 0$ is a measure of the risk aversion of the decision maker. For $\lambda=0$, the usual expected cost criterion is obtained. Larger values of λ indicate increasing risk aversion.

[1] Loulou, R., and A. Kanudia (1999) "Minimax Regret Strategies for Greenhouse Gas Abatement: Methodology and Application", *Operations Research Letters,* 25, 219-230.

Taking risk aversion into account by this formulation would lead to a non-linear, non-convex model, with all its ensuing computational restrictions. These would impose serious limitations on model size.

Utility Function with Linearized Risk Aversion

To avoid non-linearities, it is possible to replace the semi-variance by the Upper-absolute-deviation, defined by:

$$UpAbsDev\ (Cost_k) = \sum_j p_j \bullet \{Cost_{j|k} - EC_k\}^+$$

where $\boldsymbol{y} = \{\boldsymbol{x}\}^+$ is defined by the following two *linear* constraints: $\boldsymbol{y} \geq \boldsymbol{x}$, and $\boldsymbol{y} \geq \boldsymbol{0}$, and the utility is now written via the following *linear* expression:

$$U = EC + \lambda \cdot UpsAbsDev(C)$$

This is the expected utility formulation implemented into the TIMES model generator.

Note that this linearized version of the risk averse utility function is not available in the MARKAL code.

2.3 Solving approaches

General multi-stage stochastic programming problems of the type described above can be solved by standard deterministic algorithms by solving the deterministic equivalent of the stochastic model. This is the most straightforward approach, which may be applied to all problem instances. However, the resulting deterministic problem may become very large and thus difficult to solve, especially if integer variables are introduced, but also in the case of linear models with a large number of stochastic scenarios.

Two-stage stochastic programming problems can also be solved efficiently by using a Benders decomposition algorithm (Kalvelagen, 2003). Therefore, the classical decomposition approach to solving large multi-stage stochastic linear programs has been nested Benders decomposition. However, a multi-stage stochastic program with integer variables does not, in general, allow a nested Benders decomposition. Consequently, more complex decompositions approaches are needed in the general case (e.g. Dantzig-Wolfe decomposition with dynamic column generation, or stochastic decomposition methods).

The current version of the TIMES implementation for stochastic programming is solely based on directly solving the equivalent deterministic problem. As this may lead to very large problem instances, stochastic TIMES models are in practice limited to a relatively small number of scenarios.

3. TRADEOFF ANALYSIS

Analyzing tradeoffs between the standard objective function and some other possible objectives (for which the market is not able to give a price) has not been possible in an effective way until TIMES version 2.5.0. The tradeoff analysis facility is available under the stochastic mode of TIMES, which provides the basic tool for making ***sensitivity analyses*** over a number of different cases. In addition to providing the means for specifying the parameters to be varied in the sensitivity analysis, the tradeoff facility provides a tool for making a two-phase or a multiphase tradeoff analysis using completely user-defined objective functions in the first phase, or even in several phases.

3.1 Two-phase tradeoff analysis

In the ***first phase*** of the TIMES two-phase tradeoff analysis facility, the objective function can be defined as a weighted sum of any number of objective components. All of the components will refer to the LHS value of a global user constraint (i.e. a user constraint that is summed over regions and periods). Each of the component UCs can be either fully non-constraining or constrained by upper/lower bounds on the LHS. The components are defined by the user by specifying non-zero weight coefficients for the UCs to be included in the objective. The original objective function (total discounted costs) is automatically pre-defined as a non-constraining user constraint with the name '***OBJZ***', and can therefore always be directly used as one of the component UCs, if desired.

Consequently, the first phase can be considered as representing a simple Utility Tradeoff Model, which can also be used as a stand-alone option. The resulting objective function to be minimized can be written as follows:

$$\min obj1 = \sum_{uc \in UC\_GLB} W(uc) \bullet LHS(uc)$$

where:

W(uc)	=	weight of objective component *uc* in Phase 1
LHS(uc)	=	LHS expression of user constraint *uc* according to its definition
UC_GLB	=	the set of all global UC constraints (including '***OBJZ***')

In the ***second phase*** of the TIMES two-phase tradeoff analysis facility the objective function is always the ***original objective function*** in TIMES, i.e. the total discounted system costs.

In addition, in the second phase the user can specify bounds on the proportional deviation in the LHS value of ***any user constraint***, in comparison to the optimal LHS value obtained in the first phase. Such deviation bounds can be set for both global and non-global constraints, and for both non-constraining and constrained UCs (any original absolute bounds are overridden by the deviation bounds). The ***objective function used in Phase 1*** is now available as an additional pre-defined UC, named '***OBJ1***', so that one can set either deviation bounds or absolute bounds on that as well, if desired. In addition, both the total and regional original objective function can be referred to by using the pre-defined UC name '***OBJZ***' in the deviation bound parameters.

The objective function to be minimized in the second phase, and the additional bounds on the LHS values of UCs, can be written as follows:

$$\min objz \quad = \quad LHS('OBJZ')$$

$$\left.\begin{array}{l} LHS(uc) \le (1 + maxdev(uc)) \bullet LHS^{*}(uc) \\ LHS(uc) \ge (1 - maxdev(uc)) \bullet LHS^{*}(uc) \end{array}\right\} \begin{array}{l} \text{for each } uc \text{ for which} \\ maxdev(uc) \text{ has been specified} \end{array}$$

where:

LHS('OBJZ')	=	the standard objective function (discounted total system costs)
LHS(uc)	=	LHS expression of user constraint *uc* according to its definition
LHS(uc)*	=	optimal LHS value of user constraint *uc* in Phase 1
maxdev(uc)	=	user-specified fraction defining the max. proportional deviation in the value of LHS(uc) compared to the solution in Phase 1

Remarks:

1. Use of the two-phase tradeoff analysis facility requires that a weight has been defined for at least one objective component in the first phase.
2. If no deviation bounds are specified, the second phase will be omitted.
3. Automatic discounting of any commodity or flow-based UC components is possible by using a new UC_ATTR option 'PERDISC' (see Section 4.4), which could be applied e.g. to the user-defined objective components in Phase 1.
4. The two-phase tradeoff analysis can be carried over a set of distinct cases, each identified by a unique SOW index.

3.2 Multiphase tradeoff analysis

The multiphase tradeoff analysis is otherwise similar to the two-phase analysis, but in this case the objective function can be defined in the same way as in the Phase 1 described above also in all subsequent phases. The different objective functions in each phase are distinguished by using an additional phase index (the SOW index). Deviation bounds can be specified in each phase, such that they will be in force over all subsequent phases (any user constraints), or only in some of the succeeding phases (any user constraints excluding OBJ1). The deviation bounds defined on any of the user-defined objectives OBJ1 will thus always be preserved over all subsequent phases.

Remark:

1. Although the multiphase tradeoff analysis allows the use of any user-defined objective functions in each phase, it is highly recommended that the original objective function be used in the last phase, so that the economic sense is maintained in the final solution.

3.3 Analyzing Risk Averse Decisions

3.3.1 General approach

In the context of TIMES, risky events may be associated with the possibility of failure in a particular energy corridor at a particular time. An important distinction must be made between failures of short and long duration. A short duration failure is one that lasts a few days, weeks or months, i.e. much less than the time resolution of the TIMES model. A long duration failure lasts several years, and often until the end of the planning horizon.

The explicit representation of short-term events, while theoretically possible, would in most approaches increase the model size so much that it becomes practically intractable. Short-term failures may thus be essentially dealt with only by preventative measures such as correct sizing of storages, and judicious choice of energy corridors, without modeling any remedial actions. Consequently, elemental risk parameters will be practically useful only when related to the longer-term failures, lasting at least several years. The TIMES input parameters for elemental risks facilitate an efficient way of formulating a multi-objective optimization model for analyzing risk-averse decisions.

The facilities for fixing the model variables to a previous solution, and for running the model with limited foresight, may provide useful additional tools for the analysis of energy system failures and unanticipated shocks, possible remedial actions and overall impacts of the events.

3.3.2 Multi Criterion Decision Making

The Risk Indicator of each fuel type **f** supplied via an energy corridor **p** can be given by:

$$RI(f) = Max_{reg,\,p}\ \{\ Rf(reg,p)*Q(reg,p,f)\ \}$$

Similarly, the risk indicator of fuel chain category **g** can be given by:

$$RI(g) = Max_{f}\ \{\ SUM_{reg,p}\ Rg(reg,p,g)*F(reg,p,f)\ \}$$

where:

reg = region
f = fuel
p = energy corridor / process
Q(reg,p,f) = supply of fuel **f** via process **p** in region **reg**
F(reg,p,f) = flow of fuel **f** in technology **p** in region **reg**
Rf(reg,p) = scalar-valued risk parameter for energy corridor process **p** in region **reg**
Rg(reg,p,g) = scalar-valued risk parameter of technology **p** in region **reg**, category **g**

The new (composite) objective function can be written as:

$$\text{Min } [CX + \alpha^* \text{ SUM}_f \text{ RI(f)} + \beta^* \text{ RI(g)}]$$

where α and β are user-provided weight parameters for the risk terms.

Minimizing the sum of RI(f) essentially penalizes excessive reliance on imports from a single risky source (or from a few large risky sources) while putting more emphasis on riskier corridors than on less risky ones. It has an even stronger effect than the sum of squares that is usually used in classical indexes of insecurity. It encourages diversification of importation corridors, weighted by their risk coefficients.

Minimizing the RI(g) indicator penalizes excessive reliance on a small number of technologies, with emphasis on the riskier ones. It encourages diversification of technologies in the same fuel chain. Again here, the Max operator has a stronger impact on diversification than a sum of squares.

In TIMES, the composite objective function can be directly used as a user-defined objective function of the model (see Section 4.4), by defining S_UCOBJ for all the main objective function components. However, the necessary estimation of the weight parameters α and β may prove to be the difficult part. Therefore, a two-phase trade-off analysis approach can also be considered. In the two-phase approach, the standard objective function would be used in the first phase, and in the second phase only the terms involving the risks would be minimized, subject to the additional constraint that the value of the original objective function can increase at most by a fraction of γ. In this case, only the ratio of the weights α and β needs to be estimated.

$$\text{Min } [\text{SUM}_f \text{ RI(f)} + \beta/\alpha * \text{RI(g)}], \text{ s.t. } CX \leq CX^* \cdot (1+\gamma)$$

The TIMES tradeoff analysis facility can be used for this two-phase solution approach as well.

New input parameters corresponding to the risk factors Rf(reg,p) and Rg(reg,p,g) described above have been implemented in TIMES, and are described syntactically in Chapter 4 (see Table 6, UC_ACTBET / UC_FLOBET).

4. GAMS IMPLEMENTATION

4.1 Overview

The handling of multi-stage stochastic programming has been implemented into the GAMS code of the TIMES model generator. The stochastic mode is activated by the following setting in the run file:

```
$ SET STAGES YES
```

All the required control and input data parameters must also be specified, as explained in the following sections. The stochastic results can be made available to the VEDA-BE report generator as explained in Section 5.6.

4.2 Stages, states of the world and scenarios

The predefined set J constitutes the domain of the stochastic stages. The members of this predefined set are named '1', '2', '3', ...'50'. Therefore, in principle, a maximum of 50 stages could be defined in the event tree. The actual number of stages in a model will be one larger than the sequence number of the last stage for which the number of sub-states **SW_SUBS** is specified (see below).

The predefined set **ALLSOW** constitutes the domain of possible states of the world. Currently it has been defined to include the members '1', '2', '3', ...'64'. In other words, the maximum number of states of the world is 64. Consequently, a binary event tree could include at most 7 stages, because 2^6=64. In each stage of the event tree, the states of the world are identified by sequential integers starting from 1. For example, if there are three states in the second stage, these are identified by the numbers 1, 2 and 3. If all these three states have two sub-states in the third stage, those will be numbered 1, 2, ..., 6, so that the states 1 and 2 in the third stage are sub-states of state 1 in the second stage. The states of the world defined for the final stage of the event tree constitute the actual set of different final states to be handled, also called *scenarios*. The set of final states, **SOW**, is then of course also a subset of the domain **ALLSOW**. The alias name **W** is defined for **SOW** and the name **WW** is defined for **ALLSOW**.

Internally, the states of the world are numbered differently in the intermediate stages. The internal numbering is obtained by enumerating all stages ***p,*** excluding the last stage, in reverse order. If the first sub-state of a certain state in stage ***p*** is ***k*** in stage ***p+1***, this state will be internally numbered ***k*** in stage ***p***, instead of the sequential number. However, actually the user does not need to know anything about this internal numbering, as all the input parameters will use the original numbering based on sequential numbers at each stage. The results, on the other hand, are for all periods reported for all of the states at the final stage, because the states at the final stage represent the unique scenarios across the periods. The mapping of the scenario indexes to the original state indexes in each period is left to the user.

4.3 Parameters for stochastic programming

4.3.1 User control parameters

All control parameters for stochastic programming are available in the VEDA-FE shell, where they may be specified by the user. All control parameter have a prefix 'SW_' in the GAMS code of the model generator. The parameters are listed in Table 1 and discussed in more detail below.

Table 1. Control parameters for Stochastic TIMES.

Parameter	Description
SW_START(j)	The year corresponding to the resolution of uncertainty at each stage j, and thus the last year of the hedging phase and the point from which the event tree fans out for each of the SOW.
SW_SUBS(j, w)	The number of sub-states of the world for each SOW at stage j.
SW_SPROB(j, w)	The conditional probability of each sub-state at stage j. These conditional probabilities can be overridden by SW_PROB.
SW_PROB(w)	The total probability of each SOW at the last stage. If specified, overrides the stage-specific conditional probabilities.
SW_LAMBDA	Risk aversion coefficient.

1. The parameter **SW_START** is used to indicate when each of the stochastic stages begins. For stage 1, the value **SW_START** is always assumed to be the first MILESTONYR. If any **SW_START** for subsequent stages is not equal to one of the milestone years, it will be replaced by the first MILESTONYR following it. If **SW_START** is not specified for some stage > 1, the MILESTONYR following the **SW_START** of the previous stage is assumed. In addition, stages can also be combined, see section 4.3.2.

2. The parameter **SW_SUBS** specifies the number of sub-states of the world for each SOW at stage j. If it is not specified for stage 1, the number is determined by using the following two rules:

   - If **SW_SUBS** or **SW_SPROB** is specified for any SOW at any stage, the largest ordinal number of the SOWs in stage 2 for which either **SW_SUBS** or **SW_SPROB** is specified is used, or 1 if none is specified at stage 2.

   - If neither **SW_SUBS** nor **SW_SPROB** is specified for any stage, the largest ordinal number of the SOWs for which **SW_PROB** is specified is used.

3. The parameter **SW_SPROB** can be used to specify the conditional probabilities of the sub-states of the world of each SOW at each stage j. Another way to specify the probabilities is to specify the total probabilities of each SOW (at the last stage), see **SW_PROB**.

4. The parameter SW_PROB can be used to specify the total probability of each SOW (at the last stage). If specified, the total probability will override the total probability derived from the stage-wise conditional probabilities. Another way to specify the probabilities is thus to specify the conditional probabilities for the sub-states of the world at each stage. If the resulting final total probabilities will not sum up to 1, they will be simply normalized over all SOWs.

5. The parameter SW_LAMBDA can be used to specify the risk aversion coefficient. If not specified, the objective function represents the expected total discounted system costs without risk aversion.

4.3.2 Combining stages

Stages of the event tree can also be combined, if deemed useful. Any successive stages will be combined into a single stage if the starting year of the succeeding stage is less than or equal to the preceding stage. For example, if in the example of Figure 1 the starting year of both stages 2 and 3 would be specified to be 2010, the stages 2 and 3 would be combined so that in 2010 the event tree is expanded directly from one state of the world to four states. By using this feature, stages can even be combined with the first stage, by specifying the same value of SW_START for both stage 1 and some subsequent stages. If all stages were combined with the first stage, the resulting model would optimize all the scenarios independently of each other. This feature can be used for making a deterministic run for each scenario. This can be done best by specifying a large value of SW_START for the first stage, and by leaving the other values intact.

Combined stages can be also useful for data management, for example, when the states at stage 2 should contain some combinations of uncertain parameters. In such cases it can be useful to define the scenarios for the uncertain parameters at successive stages so that cloning becomes possible. Then the combined scenarios for stage 2 can be formed by combining these successive stages.

4.3.3 Cloning parts of the event tree

If there are more than 2 stages in the event tree, the user can optionally utilize a cloning facility for both the specification of the event tree and the specification of uncertain parameters. At each stage, cloning can be used for those SOWs, for which the number of sub-states of the world is equal to the number of sub-states for the first SOW. Cloning can in this case be activated by leaving the number of sub-states unspecified for the SOWs to be cloned. The model generator will then assume the same number of sub-states as for the first SOW. For such cloned nodes of the event tree, both the conditional probabilities and the uncertain parameters of the sub-states will be copied from the sub-states of the first SOW, whenever they have not been specified by the user. The user can thus always override the cloning by simply specifying the probabilities and/or uncertain parameters explicitly.

Cloning of the event tree can be convenient if the event tree is large, because then it can considerably reduce the amount of input data needed.

4.3.4 Uncertain input parameters

In this first version of the stochastic TIMES, the only a few uncertain input parameters have been implemented, as shown in Table 2. At a later stage more uncertain parameters may of course be added. All the uncertain input parameters have a prefix '**S_**'. The uncertain parameters can be divided into two types: absolute and relative:

Absolute parameters are applied in the same way as their deterministic counterparts, and they override the corresponding deterministic parameters in the parts of the event tree where they apply. Absolute parameters defined at a later stage of the event tree also override those defined at an earlier stage. All uncertain bound attributes are of absolute type.

Relative parameters are applied as multipliers to the corresponding deterministic baseline parameters. Relative parameters are also applied cumulatively over stages, so that any relative parameters defined at an earlier stage of the event tree are included in the combined multipliers at a later stage. Consequently, for any branches downstream in the event tree the current branch represents the baseline for which the multipliers in the succeeding stage will be applied. Uncertain demand projections have been implemented in TIMES as relative parameters. This means that the uncertain demands are expressed as multipliers applied to the baseline demand projection. The advantage of the relative parameters is that, when appropriate, they are easier to maintain than absolute parameters.

Table 2. Current set of uncertain input parameters for stochastic TIMES.

Parameter	**Description**	**Type**
S_COM_PROJ(r,y,c,j,w)	Demand projection	Rel.*
S_CAP_BND (r,y,p,l,j,w)	Bound on total installed capacity	Abs.
S_COM_CUMPRD(r,y,y,c,l,j,w)	Cumulative bound on commodity production	Abs.
S_COM_CUMNET(r,y,y,c,l,j,w)	Cumulative bound on commodity net prod.	Abs.
S_COM_FR(r,y,c,s,j,w)	Seasonal distribution of a commodity	Rel.*
S_FLO_CUM(r,p,c,y,y,l,j,w)	Cumulative bound on flow or activity	Abs.
S_FLO_FUNC(r,y,p,cg1,cg2,j,w)	Process transformation / efficiency	Rel.*
S_NCAP_COST(r,y,p,j,w)	Process investment cost	Rel.*
S_NCAP_AFS(r,y,p,s,bd)	Seasonal availability factor	Abs.*
S_UC_RHSxxx(…,l,j,w)	RHS constant of user constraint	Abs.
S_DAM_COST(r,y,c,cur,j,w)	Damage cost of net production of commodity	Abs.
S_CM_MAXC (y,item,j,w)	Bound on maximum level of climate variable	Abs.
S_CM_CONST(item,j,w)	Climate module constant (CS or SIGMA1)	Abs.

* Requires that corresponding deterministic attribute has also been defined

The user can also utilize the cloning facility described above for the specification of the uncertain parameters. As with the stage-wise conditional probabilities, cloning of uncertain parameters is done at some stage ***j*** if all of the following three conditions hold:

- The parameter has been specified for the sub-states of the first `SOW` at stage ***j***;
- The number of sub-states was left unspecified for some other `SOW` at stage ***j***;
- The uncertain parameter has not been specified for some of the sub-states of those other `SOW`, which are thus considered subject to cloning.

The last two indexes of all uncertain parameters are j (stage) and w (state of world). The stage index has been included in the parameters to ensure unambiguity; without the stage index there could easily be ambiguities in the parameter values for earlier stages. Note that demand projections are by default interpolated and extrapolated over all valid periods for each stage. Bound parameters are by default interpolated within periods only.

Example:

The event tree of the example shown in Figure 1 can be specified as follows:

```
PARAMETER SW_START  / 2 2010 /;
PARAMETER SW_SUBS   / 2.1 2, 2.2 2/;
PARAMETER SW_SPROB  / 2.1 0.5, 3.1 0.4, 3.3 0.5/;
```

Assume that High Mitigation in Figure 1 corresponds to a constant CO_2 concentration limit of 770 between 2010 and 2035, and Low Mitigation corresponds to the limits of 790 and 950 in 2010 and 2035, respectively. The mitigation parameters can then be specified as follows (the year index 0 is a placeholder for the interpolation control option):

```
PARAMETER S_CM_MAXCO2C / 0.2.1 1, 2010.2.1 770, 2035.2.1 770
                         0.2.2 1, 2010.2.2 790, 2035.2.2 950/;
```

Assume that High Growth is 5% higher than the baseline projection, and Low Growth is 5% lower. The High/Low growth parameters can then be specified as follows (assuming that the region is 'REG' and the demand is 'DEM':

```
PARAMETER S_COM_PROJ / REG.2015.DEM.3.1 1.05,
                       REG.2015.DEM.3.2 0.95,
                       REG.2015.DEM.3.3 1.05,
                       REG.2015.DEM.3.4 0.95 /;
```

4.3.5 Internal sets, parameters and control variables

The implementation uses a few internal sets and parameters. All the internal sets and parameters have a prefix '`SW_`'. Table 3 gives an overview of these sets and parameters. The implementation of the stochastic extension uses a number of GAMS control variables for renaming and adjusting the equations and variables for the additional dimension needed for stochastic programming. Table 4 summarizes the control variables.

Table 3. Internal sets and parameters for stochastic TIMES.

Set	**Description**
SW_ CHILD(j,w,w)	Child sub-states of the world for each SOW at each stage
SW_ COPY(j,w)	SOWs at each stage for which cloning is applied
SW_ MAP(t,w,j,w)	Mapping from period and internal SOW to stage and original SOW
SW_ STAGE(j,w)	Internal SOWs at each stage
SW_ T(t,w)	Valid internal SOWs in each period
SW_ TOS(w,t,w)	Mapping from redundant scenarios to unique SOW in each period t
SW_TREE(j,w,w)	Scenarios for each original SOW at each stage
SW_ TSTG(t,j)	Valid stages j for each period t
SW_ TSW(w,t,w)	Mapping from all scenarios to unique SOW in each period t
SW_ UCT(uc_n,t,w)	Valid internal SOWs in each period for period-wise user constraints
Parameter	**Description**
SW_ DESC(j,w)	Number of scenarios for each original SOW at each stage
SW_ TPROB(t,w)	Probability of each internal SOW in each period

Table 4. GAMS control variables for stochastic TIMES.

Control variable	**Value of control variable**	
	Standard	**Under stochastics**
EQ	"EQ"	"ES"
VAR	"VAR"	"VAS"
VART	"VAR"	"SUM(SW_TSW(SOW,T,W),VAS"
VARM	"VAR"	"SUM(SW_TSW(SOW,MODLYEAR,W),VAS"
VARV	"VAR"	"SUM(SW_TSW(SOW,V,W),VAS"
SOW	""	",SOW"
SWD	""	",WW"
SWTD	""	",T,WW"
SWT	""	",SW_T(T,SOW)"
SWS	""	",W)"

4.3.6 Reporting parameters

All standard reporting parameters have a prefix '**S**', and the first index is always the stochastic scenario index. Reporting parameters for the Climate Module have a prefix '**CM_S**'. The scenario index is always the first dimension of the parameters. The stochastic reporting parameters provide almost the same set of results as those that have been transferred to VEDA-BE from standard TIMES model runs, but now for each of the stochastic scenarios. However, there are a few small differences:

- All the undiscounted cost results from stochastic runs represent annualized costs, and they are divided into genuine costs and taxes/subsidies. Decommissioning costs are annualized over the same years as fixed costs.
- The activity costs and flow costs are reported at the ANNUAL level only, while in the standard reports they are reported in each timeslice.
- Reporting parameters for the levels of commodity balance and peak equations have been omitted from the stochastic reports, because the levels are normally zero anyway (except for demands). Only the marginals are thus reported.
- Under the stochastic mode, all user constraints are formulated by using slacks, and therefore the reporting parameters for user constraints represent the levels and marginals of these slack variables.

Table 5. Reporting parameters for stochastic TIMES.

Parameter	**Description**
Cost parameters	
SREG_WOBJ(w,r,item,cur)	Discounted objective value by region, type and currency
SCST_INVC(w,r,v,t,p)	Annualized undiscounted investment costs
SCST_INVX(w,r,v,t,p)	Annualized undiscounted investm. taxes and subsidies
SCST_DECC(w,r,v,t,p)	Annualized undiscounted decommissioning costs
SCST_FIXC(w,r,v,t,p)	Undiscounted fixed costs
SCST_FIXX(w,r,v,t,p)	Undiscounted fixed taxes and subsidies
SCST_ACTC(w,r,v,t,p)	Undiscounted activity costs
SCST_FLOC(w,r,v,t,p,c)	Undiscounted flow costs
SCST_FLOX(w,r,v,t,p,c)	Undiscounted flow taxes and subsidies
SCST_COMC(w,r,t,c)	Undiscounted commodity costs
SCST_COMX(w,r,t,c)	Undiscounted commodity taxes and subsidies
Level parameters	
SF_IN (w,r,v,t,p,c,s)	Flows into processes
SF_OUT (w,r,v,t,p,c,s)	Flows out of processes
SPAR_ACTL(w,r,v,t,p,s)	Activity levels of processes
SPAR_CAPL(w,r,t,p)	Total installed capacities of processes
SPAR_NCAPL(w,r,t,p)	Newly installed capacities of processes by period
SPAR_COMPRDL(w,r,t,c,s)	Commodity gross production levels
SPAR_COMNETL(w,r,t,c,s)	Commodity net production levels
SPAR_UCSL(w,uc,*,*,*)	Levels for the user constraint equations (slacks)
Marginal parameters	
SPAR_ACTM(w,r,v,t,p,s)	Marginals for the activity variables
SPAR_CAPM(w,r,t,p)	Marginals for the total installed capacity variables

Parameter	Description
SPAR_NCAPM(w,r,t,p)	Marginals for the new capacity variables
SPAR_COMPRDM(w,r,t,c,s)	Marginals for the commodity production variables
SPAR_COMNETM(w,r,t,c,s)	Marginals for the commodity net variables
SPAR_COMBALEM(w,r,t,c,s)	Marginals for the commodity balance equations (=E=)
SPAR_COMBALGM(w,r,t,c,s)	Marginals for the commodity balance equations (=G=)
SPAR_PEAKM(w,r,t,cg,s)	Marginals for the peak equations
SPAR_UCSM(w,uc,*,*,*)	Marginals for the user constraint equations (slacks)
Capacity bound parameters	
SPAR_CAPUP(w,r,t,p)	Upper bound on overall capacity in a period
SPAR_CAPLO(w,r,t,p)	Lower bound on overall capacity in a period
Climate module result parameters	
CM_SRESULT(w,item,t)	Basic results from stochastic Climate Module
CM_SMAXC_M(w,y)	Shadow price of climate variable constraint

The reporting parameters for the Climate Module correspond to the same ones in the standard mode, with the adjunction of the prefix CM_S.

4.4 Parameters for tradeoff analysis

The two-phase tradeoff analysis facility is available under the stochastic mode only. The stochastic mode should therefore be activated when using the tradeoff analysis facility. If no **SOW**s are explicitly defined by the user, the model will be run only once (**SOW**=1) using the two solution phases described above in Section 3. However, usually the user would like to estimate a full tradeoff curve, consisting of a number of discrete solution points (SOW=1,…,N). The number of points in the curve, i.e. the number of **SOW**s, should be defined by **SW_SUBS**('1', '1') = N;

The parameter attributes that can be varied in such sensitivity analyses are the same ***uncertain parameters*** that can also be used for multi-stage stochastic programming. The following uncertain attributes are perhaps the most important for tradeoff analyses:

- Uncertain RHS constants of user constraints
- Uncertain damage costs

The weight parameter *W*, which defines the coefficients for the user-defined objective components (see Section 3) can be specified by using the parameter **S_UCOBJ**, as described in Table **Error! Reference source not found.**. Optional discounting of any flow-based UC components can be activated by using the **UC_ATTR** option **'PERDISC'**. The two-phase solution procedure can be run over a maximum of 64 different cases (**SOW**s), each of which may have different values for any of the uncertain parameters. The deviation bounds to be

defined in Phase 2 can be specified with the **UC_RHSxxx** attributes, by using the **'N'** bound type. Any ***non-negative*** **'N'** value will be applied as a deviation bound in the second phase. The bound value represents the maximum proportional deviation allowed in the value of the **LHS** expression of the UC constraint, as described in Section 3. Negative **'N'** bounds are ignored, and therefore negative bound values can always be safely used for generating non-constraining user-defined equations for reporting purposes. By using the uncertain **S_UC_RHSxxx** attributes, the deviation bounds to be applied in Phase 2 can be varied over **SOW**s. The predefined UC names '***OBJZ***' and '***OBJ1***' can be used in deviation bounds to refer to the original or user-defined objective functions, respectively ('***OBJZ***' also by region). '***OBJZ***' can naturally also be used also in **S_UCOBJ**.

Remark: If the objective in Phase 1 is defined for only a single **SOW** (1), the same objective will also be used for any subsequent **SOW** points to be analyzed (according to the number of **SOW** as defined by **SW_SUBS**('1', '1')).

Table 6. New input parameters for two-phase tradeoff analysis in TIMES.

Parameter	**Description**
S_UCOBJ (uc_n,w)	Weight coefficients for the components of the objective function in the first phase of the tradeoff facility, and for each SOW to be analyzed (max. 64 different cases). Interpolation: Not available.
UC_ACTBET (uc_n,reg,datayear,p)	Can be used for assigning the risk of each energy corridor **p**. The parameter value will be multiplied by the activity of the corridor in user-specified period(s), i.e. the quantity supplied via the corridor, and the maximum of the resulting values over the corridors supplying the same fuel type is derived into the variable VAR_UCR(uc_n,r). The corridors supplying the same fuel type are identified by having the same **uc_n** name in the UC_ACTBET attribute (see Section 3.3).
UC_FLOBET (uc_n,reg,datayear,p,cg)	Can be used for assigning the risk of each technology **p**. and fuel group **cg**. The parameter value will be multiplied by the flow of fuels **cg** of the technology in user-specified period(s), and summed over all technologies prc having a UC_FLOBET attribute with the same **uc_n** and fuel group **cg**. The maximum of the resulting values over those having the same uc_n is derived into VAR_UCR(uc_n,r) (see Section 3.3).
UC_ATTR (r,uc_n,'LHS',grp,'PERDISC')	Flag indicating that discounting is to be applied to the periods in the LHS side of UC constraint. Applicable to UC components (grp) UC_ACT, UC_FLO, UC_IRE, UC_COMPRD, UC_COMNET and UC_COMCON, for any UC summed over periods.

4.5 Stochastic variables

As noted earlier, the variables that are used to model the stochastic programming version of TIMES are the same variables that make up the deterministic TIMES model, with two minor adjustments. The main difference is that the variables require another index corresponding to the state-of-the-world, **SOW**. To standardize the handling of this index it is always introduced after the period index, thus it is usually the second index (or the first if there is no period index) in the variable. To accommodate this requirement each standard model variable name is adjusted by replacing the standard prefix of the variable name, **VAR_**, by **VAS_**. So for example the capacity variable, **VAR_CAP**(r,t,p) becomes **VAS_CAP**(r,t,p,sow). During matrix generation the appropriate SOW index value is then entered into **VAS_CAP** according to the set **SW_T** and the period being worked on.

As there is thus essentially no redefinition of the variables for the stochastic formulation, other than the control of the instances of the variable according to the control sets **SW_T** and **SW_TSW**, the user is referred to Chapter 5 of the TIMES Reference Manual for details on the variables of the model. Below in Table 7 the variables strictly involved with the stochastic version are listed. However, as it is rather straightforward, the description of the variable details is not repeated here for the stochastic variables.

Table 7. Variables for stochastic TIMES.

Variable	Description
VAS_UC (‘OBJZ’,w)	The variable equal to the sum of the total discounted system cost associated with each SOW.
VAS_UC (‘OBJ1’,w)	The variable equal to the total objective function in Phase 1 of the Tradeoff Analysis facility (not used under multi-stage stochastics).
VAS_EXPOBJ	The variable equal to the expected value of the total discounted system cost.
VAS_UPDEV (w)	The upside deviation between the total system cost for each SOW and the expected value of the total system cost.

4.6 Equations

As noted earlier, and as is the case with the variables, the equations that are used to model stochastics are the same equations that make up the non-stochastic TIMES model with two minor adjustments. The main difference is that the equations require another index corresponding to the state-of-the-world, **SOW**. To standardize the handling of this index, either the **SOW** index as such or the set **SW_T**(t,sow) is introduced after all the other indexes of the equation. The set **SW_T** is used instead of **SOW** whenever the equation concerns a single period. To accommodate the required modifications, each standard model equation name is adjusted by replacing the standard prefix in the equation name, **EQ_**, by **ES_**. So, for example the capacity transfer equation, **EQ_CPT**(r,t,p) becomes **ES_CPT**(r,t,p,**SW_T**(t,sow)). During matrix generation the appropriate **SOW**

Table 8. Equations for stochastic TIMES.

Equation	**Description**
ES_SOBJ ('N',w)	The total discounted cost associated with each SOW.
ES_EXPOBJ	A) The expected value of the total discounted system cost, taking into consideration the probability of each event path. B) Under the two-phase Tradeoff Analysis, the user-defined objective function in the first Phase.
ES_UPDEV(w)	The upside absolute deviation between the total system cost for each SOW and the expected value of the total system cost. These equations are generated only when the risk aversion coefficient SW_LAMBDA has been specified.
EQ_OBJ	The multi-objective function, i.e. the full objective function obtained by adding to the expected total system cost a risk term obtained by multiplying the risk aversion intensity SW_LAMBDA by the upside absolute deviation (probability weighted sum of the upside deviations for every state of world).
ESG_UCMAX (uc_n,r,item,c,type)	Sets the maximum risk (over regions & corridors or over fuels) into the VAS_UC or VAS_UCR variable, according to the user-defined UC_ACTBET / UC_FLOBET attributes; *item* is internally set to either a process name or 'N'; ACTBET/FLOBET are distinguished according to the internally derived *type* index.
ESG_UCSUMAX (uc_n,w)	Sets the sum of regional maximum risk variables to an aggregate global risk variable.

index value is then entered into the **ES_CPT** equations according to the set **SW_T** and the period being worked on.

As there is thus essentially no redefinition of the equations for the stochastic formulation, other than the objective function and the control over the generation of the appropriate equations and variables according to the control sets **SW_T** and **SW_TSW** mentioned above, the user is referred to Chapter 5 of the TIMES Reference Manual for details on the core equations of the model. In Table 8, the few equations directly related to only stochastic version are listed and briefly described. The equations include the standard expected value for the stochastic objective function, the deviation equations, and, finally, the formula for the generalized objective function including the risk aversion penalties.

Note that the equation for the final objective function, **EQ_OBJ**, has the same name as in standard TIMES. Only the definition of this equation is different under the stochastic mode. Similarly, also the objective variable ObjZ has the same name as in standard TIMES.

4.7 Supported TIMES extensions

The implementation of stochastic programming has been extended to support the use of multi-stage stochastic programming also with the following TIMES extensions that are included in the standard distribution:

- The Climate Module (CLI)
- The Lumpy Investment extension (DSC)
- The Endogenous Technological Learning extension (ETL)
- The Damage Cost Functions (DAM)
- The IER extension of University of Stuttgart (IER)

The stochastic mode cannot be used with the TIMES-MACRO model variants.

4.8 Changes in model generator code

The implementation required extensive modifications to the existing code as well as a number of new components in the model generator. In total about 90 existing files were modified, and 11 new code files were added.

The new code components are listed in Table 9. The new files that are solely related to stochastic TIMES have the extension '.stc', with the exception of RPT_STC.cli, which is the report driver for the Climate Module under stochastic programming. A new general-purpose routine for annual costs was implemented, and it is used for generating the reporting cost parameters under stochastics (COST_ANN.rpt). To assist future changes in the code, a small helper routine was implemented for the inclusion of generalized variables in model equations (CAL_VAR.mod). This helper routine automates the

Table 9. New files in the TIMES model generator code.

File	Description
INITMTY.stc	Declarations for the stochastic extension
STAGES.stc	Preprocessing and management of stochastic stages
RPTMAIN.stc	Main driver for reporting results from stochastic runs
FILLSOW.stc	Cloning and processing of uncertain input parameters
RENAME.stc	Reporting of stochastic results in scenario files (used for ETL only)
SOLVE.stc	Solving stochastic model with possible direct decomposition
RECURRIN.stc	Macros and extra preprocessing for the SPINES mode
CLEARSOL.stc	Clears solution values (used in SPINES mode)
PEXTLEVS.stc	Calculate expected values (used in SPINES mode)
COST_ANN.rpt	Calculation of annual costs for stochastic results
EQDECLR.mod	Declaration of model equations (parameterization for stochastics)
CAL_VAR.mod	Helper for inclusion of stochastic variables in inter-period equations
RPT_PAR.cli	Calculation of the reporting parameters in the Climate Module
RPT_STC.cli	Driver for reporting of stochastic runs in the Climate Module

Table 10. Code files of the TIMES model generator with substantial changes.

File	Description
BND_SET.mod	Uncertain bound parameters for capacities
BNDMAIN.mod	Handling of stochastic indexes for bounds
CAL_CAP.mod	Dynamic equations for capacity related flows
CAL_NCOM.mod	Dynamic equations for investment and decommissioning flows
EQCAPACT.mod	Dynamic capacity utilization equations
EQCOMBAL.mod	Uncertain demand parameters
EQCPT.mod	Dynamic capacity transfer equations
EQCUMCOM.mod	Dynamic cumulative commodity equations
EQDAMAGE.mod	Dynamic equations for objective component of damages
EQMAIN.mod	Handling of basic equation differences under stochastic TIMES
EQOBJ.mod	Objective functions for stochastic programming
EQOBJELS.mod	Dynamic equations for objective component
EQOBJFIX.mod	Dynamic equations for objective component
EQOBJINV.mod	Dynamic equations for objective component
EQOBJVAR.mod	Dynamic equations for objective component
EQOBSALV.mod	Dynamic equations for objective component
EQPEAK.mod	Dynamic capacity/peaking equations
EQSTGIPS.mod	Dynamic storage equations
EQUSERCO.mod	Dynamic and cumulative user constraints
MAINDRV.mod	Handling of the main control variables for stochastic TIMES
MOD_EQUA.mod	Parameterized equation declarations
RPTMAIN.mod	Handling of report generation under stochastic TIMES
ATLEARN.etl	Reports from the ETL extension under stochastic TIMES
EQU_EXT.etl	Dynamic learning equations (former EQUETL.etl)
EQCAFLAC.vda	Dynamic capacity utilization equations
EQU_EXT.cli	Dynamic carbon balance equations, dynamic concentration bounds
RPT_EXT.cli	Handling of report generation under stochastic TIMES

parametrization of the variables so that they will be correctly dimensioned and mapped, if the model is run under stochastics.

The TIMES code files that were most substantially changed during the implementation are listed in Table 10. All files that involve dynamic equations are here classified to have undergone substantial changes, because dynamic equations require special handling of the stochastic variables.

In addition, the file EQMAIN.mod was divided into two parts during the implementation (EQDECLR.mod and EQMAIN.mod). Moreover, some files related to the ETL extension were renamed to conform to the standard conventions for TIMES extensions.

5. USER'S REFERENCE

5.1 Activating the stochastic mode

The stochastic mode can be activated by using the following setting in the run file:

```
$  SET STAGES YES
```

If the stochastic mode is used for sensitivity analysis only, it can be alternatively activated also by using the following setting:

```
$  SET SENSIS YES
```

All the control and input parameters of the stochastic extension are only available when using either of these settings. When using the **SENSIS** setting for sensitivity analysis, the model will be solved sequentially in each of the stochastic scenarios, using the basis information from each run as a starting basis for the next run.

5.2 Specification of states of the world and scenarios

The user does not need to specify the stages, states of the world, or scenarios explicitly. The predefined domain for the stages is the set J, which contains the elements '1','2',...,'999', sufficient for any conceivable stochastic TIMES model. The predefined domain for the states is the set ALLSOW, which by default contains the elements '1','2',...,'64'. Consequently, a maximum of 64 states (at any given stage) can be used in the specification of a stochastic model. The same maximum amount applies, of course, also to the scenarios, which are the states of the last stage. If necessary, the maximum amount can be increased to any number NN<1000, by the following control setting:

```
$  SET MAXSOW NN
```

5.3 Specification of input parameters

5.3.1 Specification of control parameters

1. Use the parameter **SW_START** to indicate when each of the stochastic stages begins. **SW_START** is optional. The following rules apply:

   - For stage 1 no value needs to be specified (unless some stages are combined into the first stage), because the first stage always starts in the first period.
   - If any **SW_START** specified for subsequent stages is not equal to one of the milestone years, it will be automatically replaced by the first milestone year following it.
   - If **SW_START** is not specified for some stage, the milestone year following the **SW_START** of the previous stage is assumed.

- Stages can be combined by specifying equal or decreasing values of SW_START for successive stages.
- Equivalent deterministic runs for each scenario can be made by specifying for the first stage a value of SW_START larger than any milestone year.

2. Use the parameter SW_SUBS to specify the number of sub-states of the world for each SOW at each stage j, if any. The use of SW_SUBS is required if more than two stochastic stages are modeled. The following rules apply:

    - If SW_SUBS is not specified for stage 1, the number of states in stage 2 is determined by the model generator from the other control parameters (for details see section 4.3.1).
    - For any subsequent stages that have sub-states, SW_SUBS must be specified for at least the first SOW. For those SOW that SW_SUBS is left unspecified, the number of the first SOW is assumed, and these SOWs will be subject to cloning.

3. Use the parameter SW_SPROB to specify the conditional probabilities of the sub-states of each SOW at each stage j. Alternatively, the total probabilities of each SOW (at the last stage) can be specified, by using SW_PROB. The use of SW_SPROB is the recommended method, for which the following rules apply:

    - If the parent SOW is subject to cloning, SW_SPROB can be left unspecified, and it will then inherit the probabilities from the sub-states of the first SOW at the previous stage.
    - If SW_SPROB is not specified and the parent SOW is not subject to cloning, SW_SPROB will be automatically assigned a probability UP/m, where UP is the unassigned probability among the sub-states of the parent SOW, and m is the number of remaining sub-states for which probabilities are to be assigned. An even probability distribution among the sub-states can thus be specified without using any SW_SPROB (for non-cloned SOW), or by specifying the probability for the first sub-state only (for cloned SOW).

4. Use the parameter SW_LAMBDA to specify the risk aversion coefficient. If not specified, the objective function represents the expected total discounted system costs without risk aversion.

A quick reference of the use of the control parameters is given in Appendix B.

5.3.2 Specification of uncertain parameters

The uncertain parameters shown in Table 11 can be currently used in stochastic TIMES. A quick reference of the use of the input parameters is given in Appendix C. Apart from the special aspects concerning the ***relative type***, the use of the uncertain parameters is basically similar to the corresponding deterministic parameters. However, the index of both the stage and state of the world has to be specified when using these parameters. The same default interpolation rules are applied to the uncertain parameters as to the

Table 11. Current set of uncertain input parameters for stochastic TIMES.

Parameter	**Description**	**Type**
S_COM_PROJ(r,y,c,j,w)	Demand projection	Rel.*
S_CAP_BND (r,y,p,l,j,w)	Bound on total installed capacity	Abs.
S_COM_CUMPRD(r,y,y,c,l,j,w)	Cumulative bound on commodity production	Abs.
S_COM_CUMNET(r,y,y,c,l,j,w)	Cumulative bound on commodity net prod.	Abs.
S_COM_FR(r,y,c,s,j,w)	Seasonal distribution of a commodity	Rel.*
S_FLO_CUM(r,p,c,y,y,l,j,w)	Cumulative bound on flow or activity	Abs.
S_FLO_FUNC(r,y,p,cg1,cg2,j,w)	Process transformation / efficiency	Rel.*
S_NCAP_AFS(r,y,p,s,bd)	Seasonal availability factor	Abs.*
S_NCAP_COST(r,y,p,j,w)	Process investment cost	Rel.*
S_UC_RHSxxx(…,l,j,w)	RHS constant of user constraint	Abs.
S_DAM_COST(r,y,c,cur,j,w)	Damage cost of net production of commodity	Abs.
S_CM_MAXC(y,item,j,w)	Bound on maximum level of climate variable	Abs.
S_CM_CONST(item,j,w)	Climate module constant (CS or SIGMA1)	Abs.

* Requires that corresponding deterministic parameter is also defined

deterministic counterparts. Relative parameters are applied as multipliers to the corresponding deterministic baseline parameters, and cumulatively over stages. Any relative parameters defined at an earlier stage of the event tree are included in the combined multipliers at a later stage.

The user can also utilize the cloning facility for the automatic copying of uncertain parameters from the sub-states of the first state at each stage to the sub-states of other states. As mentioned above, these other states are subject to cloning only if **SW_SUBS** was left unspecified for them.

5.4 Sensitivity analyses

The stochastic mode can also be used for a series of deterministic runs. This can be accomplished by combining all (or some of) the stochastic stages with the first stage. An easy way to do this is to specify for the first stage a value of **SW_START** larger than all (or some of) the subsequent **SW_START**. Consequently, the subsequent stages will have a value of **SW_START** less than or equal to the first stage, and are therefore combined with the first stage. In effect, this will mean that all those branches of the event tree that are distinct already at the first stage will be solved independently of each other. If all stages are combined, the stochastic scenarios will be run fully independently. The SENSIS setting described above will accomplish this without the need for setting **SW_START**.

Solving a set of deterministic scenarios in this way can be very useful for the following different purposes:

- For comparing the results from the stochastic model to the results of individual deterministic scenarios;
- For making standard sensitivity analysis with different values for the uncertain parameters.

When the model generator detects that all scenarios are disjoint already at the first stage, it uses the straightforward scenario decomposition approach to solving the problem. Thus, each scenario is in this case solved separately, one after another. The results are still reported in the same way as in a standard stochastic run.

5.5 Example: Five-stage stochastic model

In section 4.3.4 a simple example of specifying a stochastic model was already given. In this section a somewhat larger and more complete example is given, which however uses the same uncertain parameters as the earlier example.

In the run file, the stochastic mode should be activated as follows:

```
* Define the model as stochastic:
$  SET STAGES YES
```

The specification of the event tree consists of the definition of the starting years of the stages, the number of sub-states of each SOW, and the probabilities. We are defining a five-stage event tree, illustrated in Figure 3. In this example, we define the second stage to start in 2010 and the third stage in 2020. We leave the start years of the fourth and fifth stage unspecified, which means that they start in the milestone year succeeding the previous stage (default definition).

```
PARAMETER SW_START  / 2 2010, 3 2020 /;
```

We wish to have three states at the second stage, and all states will have two sub-states at each subsequent stage. Here we can utilize the cloning facility, and therefore we need to specify the number of sub-states for the first state at each stage only (and not for each state of each stage).

```
PARAMETER SW_SUBS   / 1.1 3, 2.1 2, 3.1 2, 4.1 2/;
```

If we wish to specify the conditional probabilities for the sub-states so that the distribution is the same under each parent state, we can use the cloning facility quite effectively for the specification of probabilities. For the second stage, we need to define the probabilities for the first two states (the third is derived automatically). For the remaining event tree, we only need to define the probability for the first state at each stage (the second is derived):

```
PARAMETER SW_SPROB  / 2.1 0.33, 2.2 0.34,
                      3.1 0.60, 4.1 0.55, 5.1 0.5 /;
```

However, if we wish to override the cloning of some probabilities, we can do that by specifying the probabilities explicitly. Below, the probabilities of the sub-states of the last branch at stage 4 are specified explicitly:

```
PARAMETER SW_SPROB  / 4.12 0.7 /;
```

The uncertainty concerning climate change mitigation is assumed to be resolved at the first stage. We assume a high, medium and low mitigation scenarios. In this example, we assume that the high scenario corresponds to a CO_2 concentration limit of 900 GtC in 2080, the medium scenario to 1150 and the low scenario to 1400 GtC in 2100, respectively. For all scenarios, we assume that the limit evolves linearly from the value of 900 GtC in 2010. The mitigation parameters can then be specified as follows (the year index 0 is a placeholder for the interpolation control option):

```
PARAMETER S_CM_MAXCO2C /0.2.1 1, 2010.2.1 900, 2080.2.1 900
                        0.2.2 1, 2010.2.2 900, 2080.2.2 1150
                        0.2.3 1, 2010.2.3 900, 2080.2.3 1400/;
```

The uncertainty concerning demand is assumed to be resolved gradually at the subsequent stages. At each stage we assume a high, and low growth scenario. At the third stage, high growth is 7% higher and low growth is 4% lower than the baseline. At the

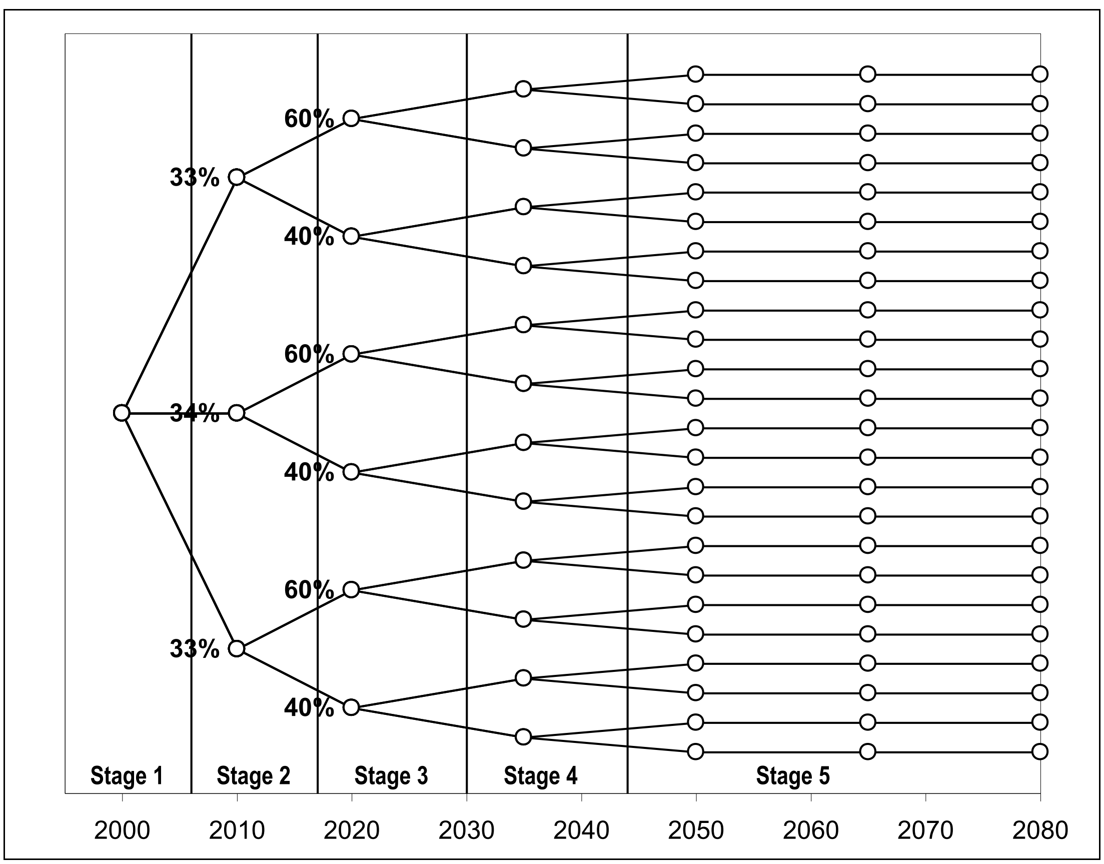

Figure 3. Event Tree for a five-stage stochastic TIMES example.

fourth stage high growth is 5% higher and low growth is 5% lower than the adjusted baseline scenarios at previous stage. Finally, at the fifth stage high growth is 4% higher and low growth is 7% lower than the adjusted baseline scenarios at the previous stage.

The cloning facility can again be effectively utilized for the demand data. Instead of specifying the demand scenarios for all of the 42 nodes at the third, fourth and fifth stage, we can specify the data for only six nodes corresponding to the sub-states of the first nodes at the second, third and fourth stage. Note that the demand parameters are by default interpolated and extrapolated over all valid periods for each stage, and thus only one data point is needed. The demand scenarios can thus be fully specified as follows:

```
PARAMETER S_COM_PROJ / REG.2015.DEM.3.1 1.07,
                       REG.2015.DEM.3.2 0.96,
                       REG.2015.DEM.4.1 1.05,
                       REG.2015.DEM.4.2 0.95,
                       REG.2015.DEM.5.1 1.04,
                       REG.2015.DEM.5.2 0.93 /;
```

After running the stochastic model, the user might wish to compare the results with equivalent deterministic scenarios. This can be accomplished by changing the start years so that the first stage has a later start year than all other stages (see section 4.4):

```
PARAMETER SW_START  / 1 9999, 2 2010, 3 2020 /;
```

5.6 Tradeoff analyses

5.6.1 Activating the tradeoff analysis mode

The tradeoff analysis facility has to be activated by either one of the following settings in the run file:

```
$   SET STAGES YES
$   SET SENSIS YES
```

The first alternative simply activates the stochastic mode, which is required when using the tradeoff facility. The second alternative additionally enables the use of the active basis information successively between solving each point of the sensitivity analysis, utilizing the so-called ***warm start*** method. In some cases, this may significantly improve the solution speed, because the solution of each point can start from the optimal solution of the previous point. However, in cases where the successive model instances to be solved differ considerably from each other, this may not be an efficient option. The judgment of whether to use the ***warm start*** or not has to be made by the user. The set-ups D and E described in the introduction are only possible with the setting `STAGES YES`, i.e. ***without*** the setting `$SET SENSIS YES`.

When using the stochastic mode for tradeoff analysis, there is no need to specify any other stochastic control parameters than the number of analysis points, which can be done by specifying the number of SOW: SW_SUBS('1','1') = N;

Therefore, the ***minimal specifications*** required to use the two-phase tradeoff analysis are the following:

- The stochastic mode is activated ($SET STAGES/SENSIS YES)
- The weight parameter S_UCOBJ is defined for some UC and for either a single or several SOW=1,…,N (see Section 3 for syntax);
- If the analysis is to be carried ***over several SOW***, the user should additionally specify the number N of the SOWs explicitly, by setting SW_SUBS('1','1') = N; (default=1).

In addition, the user can define any of the uncertain parameters over the SOW to be analyzed, and define deviation bounds by using either the deterministic or uncertain UC_RHSxxx attributes, using the 'N' bound type and a non-negative bound value. The predefined UC names '***OBJZ***' and '***OBJ1***' can be used in the ***RHS*** parameters to refer to the original or user-defined objective functions, respectively ('***OBJZ***' also by region).

The set-up B described in the Introduction corresponds to the case where S_UCOBJ is defined only for the first SOW. The corresponding set-ups D and E, which both have only a single SOW in the final phase, can only be accomplished by activating the stochastic mode by "$SET STAGES YES", and by indicating the single terminal SOW by setting S_UCOBJ('OBJ1','1'), where an explicit zero value (EPS) indicates that set-up D is selected and any non-zero value indicates that set-up E is to be used. In addition, the use of set-up D requires that no other uncertain parameters apart from S_UCOBJ and S_UC_RHSxxx are specified, and deviation bounds can be specified only for OBJ1. In the multiphase set-up E, all uncertain attributes can be freely used. However, in the multiphase case the deviation bounds in each phase can be only be specified by using the uncertain S_UC_RHSxxx parameters. The deviation bounds defined for each OBJ1_i are in the multiphase case always preserved over all subsequent phases SOW = i+1,…,N. Any other deviation bounds defined for SOW = i are also preserved, unless explicitly canceled in any subsequent phase SOW = k (k > i), by using the 'N' bound type, in which case the bounds remain in force in phases SOW = i+1,…,k.

5.6.2 Possible uses of the tradeoff facility

The simple facility described above can be used for a number of tradeoff analysis tasks. Below, just a few possible set-ups are briefly described:

1. One can optimize the problem with respect to a generic N row (phase 1), and then relax the optimal value of this auxiliary OBJ by n% and go back to the solution of an economic model by re-optimizing with the original objective function, i.e. the total

discounted costs or surplus (phase 2); by iterating over N discrete values of increasing n% one can build full tradeoff curves and calculate the supply (cost) curve of the public good reflected by row N;

2. One can preliminarily set an upper bound to the cost objective (OBJZ); then, as in point one above, one can optimize the problem with respect to a generic N row (phase 1), and then relax the optimal value of this auxiliary OBJ by n% and go back to the economic equilibrium by re-optimizing with the original objective function, i.e. the total discounted costs or surplus (phase 2); by iterating over N different rows, it is possible to identify the equivalence space of different public goods;

3. One can use uncertain damage costs to define the tradeoffs between the externalities and the standard system costs. In this way one can also introduce threshold levels for the externalities, as well as non-linearity in the value of the public good.

4. One can also use more complex, user-defined objective functions in the first phase;

5. One can use phase 1 only, by specifying a linear combination of OBJZ and some other criterion. By varying the weight of that criterion, one may obtain a full parametric analysis of the trade-offs between OBJZ and the other criterion. It is easily shown using Duality theory that this is mathematically equivalent to the sensitivity analysis described in 1 above.

6. One can optionally set an upper bound to the cost objective (OBJZ) and optimize the problem separately with respect to a set of N different generic N rows (phase 1), and then relax the optimal value of each of the auxiliary objectives OBJ(i) by n(i)% and go back to the economic equilibrium by re-optimizing with the original objective function, i.e. the total discounted costs or surplus (phase 2);

7. One can use the multiphase tradeoff analysis for complex multi-objective analyses.

5.7 Tradeoff analysis examples

Example 1a. Single two-phase tradeoff analysis (Possible use 1 above, ***without*** iteration over different deviation bounds). Assume that the generic row to be optimized in Phase 1 is ***UCEXT***, which should be ***minimized***.

- Activate stochastic mode:
  $SET STAGES YES
- Define objective function to be minimized in Phase 1:
  S_UCOBJ('UCEXT','1') = 1;
- Define a 10% deviation bound to be applied in Phase 2:
  UC_RHS('UCEXT','N') = 0.1;

Example 1b. Two-phase ***tradeoff curve*** analysis (Possible use 1 above, with iteration over 5 different ***deviation bounds***). Assume that the generic row to be optimized in Phase 1 is ***UCEXT***, which should be ***maximized***.

- Activate stochastic mode, and define number of tradeoff points:
  $SET STAGES YES
  SW_SUBS('1','1') = 5;
- Define objective function to be used in Phase 1 (applied to all 5 points):
  S_UCOBJ('UCEXT','1') = –1;
- Define the deviation bounds to be applied in Phase 2 (10%–50%):
  S_UC_RHS('UCEXT','N','2','1') = 0.1;
  S_UC_RHS('UCEXT','N','2','2') = 0.2;
  S_UC_RHS('UCEXT','N','2','3') = 0.3;
  S_UC_RHS('UCEXT','N','2','4') = 0.4;
  S_UC_RHS('UCEXT','N','2','5') = 0.5;

Example 1c. Two-phase ***tradeoff curve*** analysis (Possible use 1 above, with iteration over 5 different ***slopes***, or Lambda values). Assume that the generic externality is represented by ***UCEXT***, which should be ***minimized***.

- Activate stochastic mode, and define number of tradeoff points:
  $SET STAGES YES
  SW_SUBS('1','1') = 5;
- Define ***composite objective functions*** to be minimized in Phase 1:
  S_UCOBJ('OBJZ','1') = 1;
  S_UCOBJ('UCEXT','1') = Lambda1;
  S_UCOBJ('OBJZ','2') = 1;
  S_UCOBJ('UCEXT','2') = Lambda2;
  S_UCOBJ('OBJZ','3') = 1;
  S_UCOBJ('UCEXT','3') = Lambda3;
  S_UCOBJ('OBJZ','4') = 1;
  S_UCOBJ('UCEXT','4') = Lambda4;
  S_UCOBJ('OBJZ','5') = 1;
  S_UCOBJ('UCEXT','5') = Lambda5;
- Define deviation bounds to be applied in Phase 2. Here, as an example, we wish to ***retain the optimal value of the composite objective*** obtained in Phase 1, while minimizing the total system costs in Phase 2:
  UC_RHS('OBJ1','N') = EPS;

Example 2. Two-phase tradeoff ***equivalence space*** analysis (Possible use 2 above, with iteration over 5 different objectives). Assume that the generic rows to be optimized in Phase 1 are ***UCEXT1***, ***UCEXT2***, ***UCEXT3***, ***UCEXT4***, ***and UCEXT5***, of which the even-numbered should be maximized.

- Activate stochastic mode, and define number of tradeoff points:
  $SET STAGES YES
  SW_SUBS('1','1') = 5;

- Define an upper bound for OBJZ in Phase 1:
  UC_RHS('OBJZ','UP') = 7e6;
- Define objective functions to be used in Phase 1:
  S_UCOBJ('UCEXT1','1') = 1;
  S_UCOBJ('UCEXT2','2') = –1;
  S_UCOBJ('UCEXT3','3') = 1;
  S_UCOBJ('UCEXT4','4') = –1;
  S_UCOBJ('UCEXT5','5') = 1;
- Define a 10% deviation bound to be applied in Phase 2 (in this example, the same 10% deviation bound is applied separately to each objective):
  S_UC_RHS('UCEXT1','N','2','1') = 0.1;
  S_UC_RHS('UCEXT2','N','2','2') = 0.1;
  S_UC_RHS('UCEXT3','N','2','3') = 0.1;
  S_UC_RHS('UCEXT4','N','2','4') = 0.1;
  S_UC_RHS('UCEXT5','N','2','5') = 0.1;

Example 3. Tradeoff curve analysis using uncertain damage costs (Possible use 3 above). The two-phase tradeoff facility is not necessary here, but a deterministic sensitivity analysis is sufficient. Assume that the externality is represented by commodity ***COMEXT***.

- Activate sensitivity mode, and define number of analysis points:
  $SET SENSIS YES
  SW_SUBS('1','1') = 5;
- Define different damage costs to be analyzed (assumed constant over T):
  S_DAM_COST(R,T,'COMEXT',CUR,'2','1') = 20;
  S_DAM_COST(R,T,'COMEXT',CUR,'2','2') = 40;
  S_DAM_COST(R,T,'COMEXT',CUR,'2','3') = 60;
  S_DAM_COST(R,T,'COMEXT',CUR,'2','4') = 80;
  S_DAM_COST(R,T,'COMEXT',CUR,'2','5') = 100;

This Example 3 will also define a tradeoff curve between the externality and system costs, using different assumptions for the marginal damage costs.

Example 4. Tradeoff analysis optimizing separately with several different objectives in the first phase, and using deviation bounds for each of them in a single run in the second phase (Possible use 6 above). Assume that the generic rows to be optimized in Phase 1 are ***UCEXT1***, ***UCEXT2***, ***UCEXT3***, ***UCEXT4***, ***and UCEXT5***, of which the even-numbered should be maximized.

- Activate stochastic mode, and define number of tradeoff points:
  $SET STAGES YES
  SW_SUBS('1','1') = 5;
- Define an upper bound for OBJZ in Phase 1:
  UC_RHS('OBJZ','UP') = 7e6;

- Define objective functions to be used for each SOW:

  S_UCOBJ('UCEXT1','1') = 1;
  S_UCOBJ('UCEXT2','2') = –1;
  S_UCOBJ('UCEXT3','3') = 1;
  S_UCOBJ('UCEXT4','4') = –1;
  S_UCOBJ('UCEXT5','5') = 1;

- Set a flag indicating that a single model is to be used in the second (or last) Phase. Any non-zero value for the flag would activate the ***multiphase set-up E***, thereby optimizing UCEXT5 in the last phase under deviation bounds on the other objectives. However, in this example, the ***two-phase set-up D*** is desired, and therefore an explicit zero value has to be specified for this flag:

  S_UCOBJ('OBJ1', '1') = EPS;

- Define deviation bounds to be applied in Phase 2. In this example, 10% or 20% deviation bounds are applied to odd and even numbered objectives, respectively).:

  S_UC_RHS('OBJ1','N','2','1') = 0.1;
  S_UC_RHS('OBJ1','N','2','2') = 0.2;
  S_UC_RHS('OBJ1','N','2','3') = 0.1;
  S_UC_RHS('OBJ1','N','2','4') = 0.2;
  S_UC_RHS('OBJ1','N','2','5') = 0.1;

5.8 Exporting results to VEDA-BE

For reporting the results of the stochastic models, the attributes listed in Table 12 have been added for the transfer of results into VEDA_BE:

Table 12. Reporting parameters for stochastic TIMES.

Parameter	**Description**
Cost parameters	
SREG_WOBJ(w,r,item,cur)	Discounted objective value by region, type and currency
SCST_INVC(w,r,v,t,p)	Annualized undiscounted investment costs
SCST_INVX(w,r,v,t,p)	Annualized undiscounted investm. taxes and subsidies
SCST_DECC(w,r,v,t,p)	Annualized undiscounted decommissioning costs
SCST_FIXC(w,r,v,t,p)	Undiscounted fixed costs
SCST_FIXX(w,r,v,t,p)	Undiscounted fixed taxes and subsidies
SCST_ACTC(w,r,v,t,p)	Undiscounted activity costs
SCST_FLOC(w,r,v,t,p,c)	Undiscounted flow costs
SCST_FLOX(w,r,v,t,p,c)	Undiscounted flow taxes and subsidies
SCST_COMC(w,r,t,c)	Undiscounted commodity costs
SCST_COMX(w,r,t,c)	Undiscounted commodity taxes and subsidies
Level parameters	

Table 12. Reporting parameters for stochastic TIMES.

Parameter	**Description**
SF_IN (w,r,v,t,p,c,s)	Flows into processes
SF_OUT (w,r,v,t,p,c,s)	Flows out of processes
SPAR_ACTL(w,r,v,t,p,s)	Activity levels of processes
SPAR_CAPL(w,r,t,p)	Total installed capacities of processes
SPAR_NCAPL(w,r,t,p)	Newly installed capacities of processes by period
SPAR_COMPRDL(w,r,t,c,s)	Commodity gross production levels
SPAR_COMNETL(w,r,t,c,s)	Commodity net production levels
SPAR_UCSL(w,uc,*,*,*)	Levels for the user constraint equations (slacks)
Marginal parameters	
SPAR_ACTM(w,r,v,t,p,s)	Marginals for the activity variables
SPAR_CAPM(w,r,t,p)	Marginals for the total installed capacity variables
SPAR_NCAPM(w,r,t,p)	Marginals for the new capacity variables
SPAR_COMPRDM(w,r,t,c,s)	Marginals for the commodity production variables
SPAR_COMNETM(w,r,t,c,s)	Marginals for the commodity net variables
SPAR_COMBALEM(w,r,t,c,s)	Marginals for the commodity balance equations (=E=)
SPAR_COMBALGM(w,r,t,c,s)	Marginals for the commodity balance equations (=G=)
SPAR_PEAKM(w,r,t,cg,s)	Marginals for the peak equations
SPAR_UCSM(w,uc,*,*,*)	Marginals for the user constraint equations (slacks)
Capacity bound parameters	
SPAR_CAPUP(w,r,t,p)	Upper bound on overall capacity in a period
SPAR_CAPLO(w,r,t,p)	Lower bound on overall capacity in a period
Climate module result parameters	
CM_SRESULT(w,item,t)	Basic results from stochastic Climate Module
CM_SMAXC_M(w,item,y)	Shadow price of climate variable constraint

Under the stochastic mode, user constraints are modeled using slack variables, with no loss in generality. Therefore, the reporting parameter `SPAR_UCSL` contains the levels of the slack variables, and the parameter `SPAR_UCSM` represents the marginals of the slack variables (undiscounted when the constraint is region-specific).

In stochastic and sensitivity analyses the full results are produced for all `SOW`. In the two-phase tradeoff analysis the results are included for all `SOW` in the second phase. In the multiphase analysis the results are reported for the single `SOW` in each phase, unless the user explicitly turns out the reporting for certain `SOW` by setting a negative value for `S_UCOBJ('OBJ1',SOW)`. However, results for any terminal `SOW` are always reported.

Table 13. Basic Climate Module reporting items for stochastic TIMES.

Item name	Climate module variable/parameter	Description
CO2-PPM	VAS_CLITOT(CO2-ATM)	CO_2 concentration by milestone year (ppm)
CO2-GTC	VAS_CLITOT(CO2-GTC)	Total CO_2 emissions by milestone year (GtC)
CO2-ATM	VAS_CLIBOX(CO2-ATM)	Mass of CO_2 in the atmosphere (GtC)
CO2-UP	VAS_CLIBOX(CO2-UP)	Mass of CO_2 in the upper ocean layer (GtC)
CO2-LO	VAS_CLIBOX(CO2-LO)	Mass of CO_2 in the deep ocean layer (GtC)
CH4-PPB	VAS_CLIBOX(CH4-ATM)	CH_4 concentration by milestone year (ppb)
CH4-MT	VAS_CLITOT(CH4-MT)	Total CH_4 emissions by milestone year (Mt)
N2O-PPB	VAS_CLIBOX(N2O-ATM)	N_2O concentration by milestone year (ppb)
N2O-MT	VAS_CLITOT(N2O-MT)	Total N_2O emissions by milestone year (Mt)
FORC+TOT	VAS_CLITOT(FORCING)	Increase in radiative forcing (W/m^2)
FORC+CO2	VAS_CLIBOX(CO2-ATM)	Increase in radiative forcing (W/m^2) from CO_2
FORC-CH4	VAS_CLIBOX(CH4-ATM)	Increase in radiative forcing (W/m^2) from CH_4
FORC-N2O	VAS_CLIBOX(N2O-ATM)	Increase in radiative forcing (W/m^2) from N_2O
DELTA+ATM	VAS_CLIBOX(DELTA-ATM)	Increase in atmospheric temperature (ºC)
DELTA+LO	VAS_CLIBOX(DELTA-LO)	Increase in deep ocean temperature (ºC)

Concerning the basic results from the Climate Module, the two result parameters include an index (item) for the various result quantities, and this index is translated to the commodity dimension in VEDA-BE. The basic result attributes of the Climate Module are listed in Table 13, together with the standard name of the corresponding variable or result parameter in the module.

Since all these reporting attributes and the related variables are only created by the model when the stochastic extension is used, a special vdd file, called times2veda_stc.vdd has to be applied when transferring results from TIMES to VEDA-BE by using the gdx2veda utility.

6. References

Dantzig, G. B. 1955. Linear programming under uncertainty. *Management Science* 1, 197–206.

Kalvelagen, E. 2003. Two-stage stochastic linear programming with GAMS. GAMS Corporation.

Kanudia, A. & Loulou, R. 1998. “Robust Responses to Climate Change via Stochastic MARKAL: the case of Québec”, *European Journal of Operations Research*, vol. 106, pp. 15–30.

de Kruijk, H. & Goldstein, G. 1999. MARKAL – MUSS User Information Note: Stochastic MARKAL. January 20, 1999.

Loulou, R. & Kanudia, A. 1999. “Minimax Regret Strategies for Greenhouse Gas Abatement: Methodology and Application”, *Operations Research Letters*, 25, 219–230.

Loulou, R., Goldstein, G. & Noble, K. 2004. *Documentation for the MARKAL Family of Models*. October 2004. http://www.etsap.org/documentation.asp

Loulou, R., Remme, U., Kanudia, A., Lehtilä, A. & Goldstein, G. 2005. *Documentation for the TIMES Model*. Energy Technology Systems Ananlysis Programme (ETSAP), April 2005. http://www.etsap.org/documentation.asp

Wets, R.J.-B. 1989. *Stochastic programming.* In G.L. Nemhauser, A.H.G. Rinnoy Kan, and M.J. Todd, editors, Handbook on Operations Research and Management Science, volume 1, pages 573–629. North-Holland, 1989.

APPENDIX A: EXPERIMENTAL VARIANT FOR MODELING RECURRING UNCERTAINTIES

In the multi-stage stochastic mode it is generally assumed that the uncertain parameters are related to some long-term development, such that when the uncertainty is resolved, the information can be utilized when making investment decisions for the future. However, the mathematical formulation should be somewhat different if the uncertainties are recurring at short time frames, thereby effectively preventing the utilization of the information about the resolved uncertainties for any future investment decisions.

Recurring uncertainties that are important for the energy system include for example, hydrological conditions (wet/dry years), wind conditions, and fuel price fluctuations. Such uncertainties are commonly modeled by stochastic programming techniques in short-term operational models, but rarely in long-term models such as TIMES.

When modeling these kinds of recurring uncertainties, all the investment decision variables should only have a single state-of-the-world index in all periods, and only the period-specific flow and activity variables should thus be split into the sets of states implied by the event tree. Because the uncertainties are recurring in each period, the event tree should apparently also branch in every model period into the set of possible states in the following period. An example of such a rapidly expanding event tree is shown in Figure 4, assuming that hydrological conditions are the uncertain parameter to be modeled.

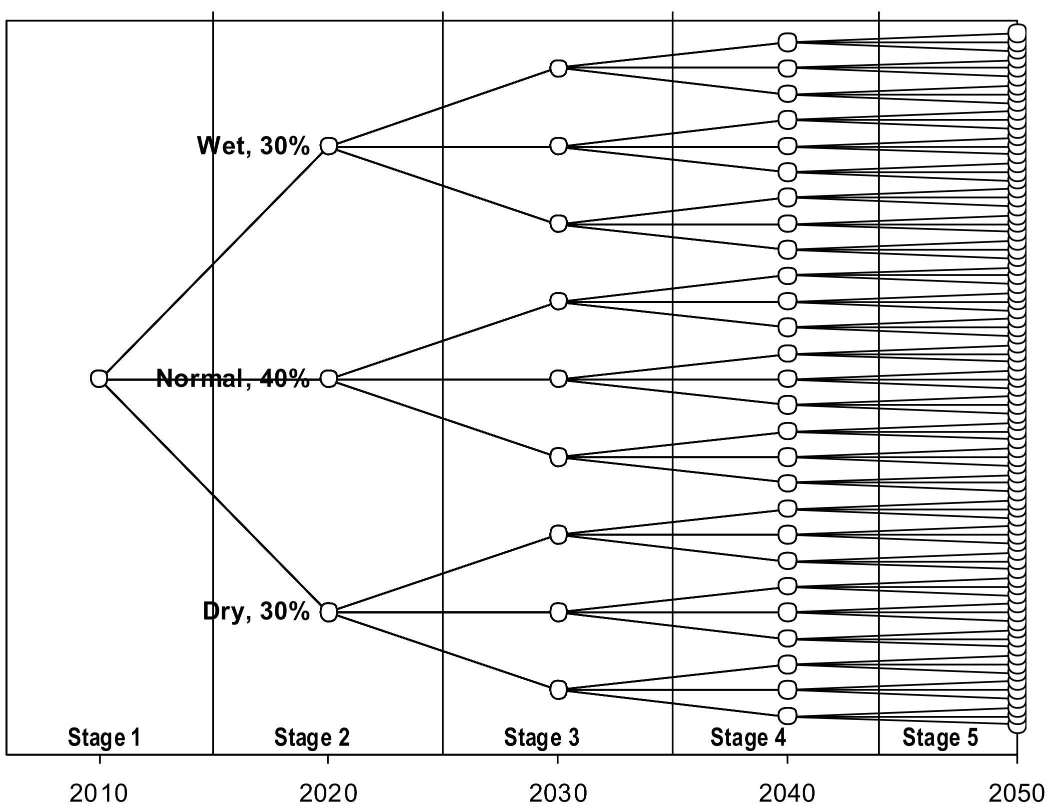

Figure 4. Example event tree for recurring uncertainties in hydrological conditions.

TIMES supports the modeling of such recurring uncertainties by providing an experimental variant of the basic multi-stage stochastic programming mode, where the sets of the state-of-the-world indices are restricted to non-capacity-related variables only. The variant can be activated by the control statement `$SET SPINES YES`.

In addition, TIMES also provides an additional sub-mode for this variant, where the uncertainties of the state variables are assumed independent of each other between successive periods. With this simplifying assumption the event tree can be highly reduced, because the impacts of the uncertainties are no longer conditional on the state-of-the-world in the preceding period. In mathematical terms, all the dynamic and cumulative constraints are simplified to refer only to the expected values of the variables in the preceding period, instead of referring to the state-specific values. For inter-period storage, the initial storage level is thus given as the expected value from the preceding period, but the charging and discharging may be optimized according to the state of the current period. Cumulative and dynamic user constraints are based on the expected values all through. This sub-variant can be activated by the additional setting `$SET SOLVEDA 1`.

Under these simplifying assumptions on the independence of the uncertainties, the event tree of Figure 4 can thus be basically reduced to that illustrated in Figure 5, effectively leading to a two-stage stochastic program. Without a substantial event tree reduction the modeling of recurring uncertainties would easily lead to unrealistically large problem instances of long-term models. However, note that unlike the standard stochastic mode, the SPINES variant can also be used together with the time-stepped mode of TIMES.

Remark: The design and implementation of this additional feature of the TIMES stochastic mode should be considered experimental only.

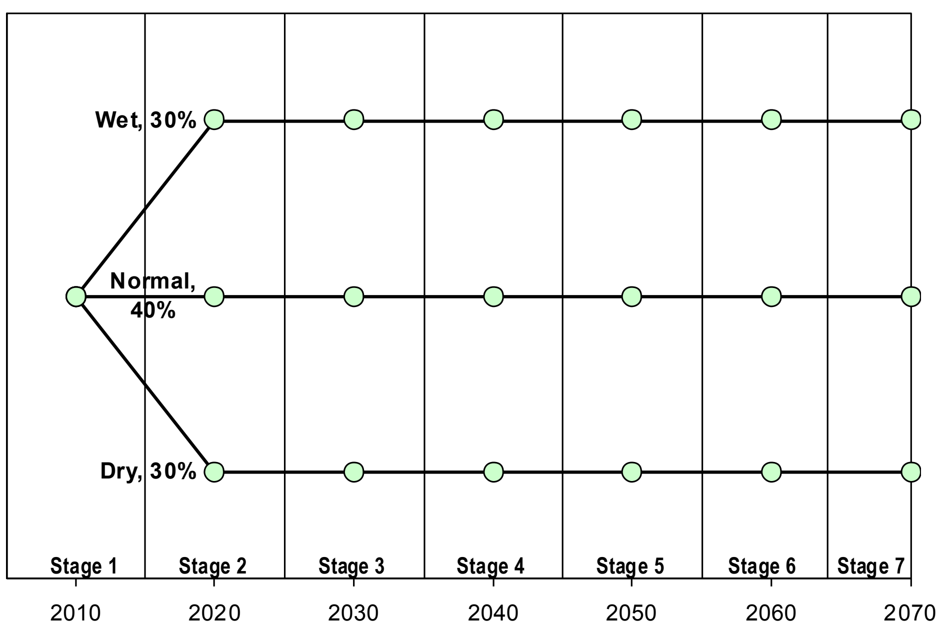

Figure 5. Reduced event tree for simplified recurring hydrological uncertainties.

Appendix B: Control parameters for Stochastic TIMES

Input parameter (Indexes)[1]	**Related parameters**[2]	**Units / Ranges & Default values & Default inter-/extrapolation**[3]	**Instances**[4] (Required / Omit / Special conditions)	**Description**	**Affected equations or variables**[5]
SW_START (j)	SW_SUBS, SW_SPROB, SW_PROB, SW_LAMBDA	• Year • [BOT, EOT]; integer; default value: see doc • Default i/e[6]: n/a	• If less than or equal to start of previous stage, stage is combined with previous	Start year of stochastic stage j	• All
SW _SUBS (j,w)	See above	• Dimensionless • [1, 64]; default value: see Instances	• Can be left unspecified if specified for first state, which is then the default	Number of sub-states of the world for state w at stage j	• All
SW _SPROB (j,w)	See above	• Dimensionless • [0, 1]; default value: none	• Can be left unspecified; see doc for details	Conditional probability of sub-state w at stage j, given its parent's state	• All
SW _PROB (w)	See above	• Dimensionless • [0, 1); default value: none	• Use instead of SW_SPROB; mainly useful for MARKAL style 2-stage programs only	Total probability of stochastic scenario	• All
SW_LAMBDA	See above	• Dimensionless • [0, INF);	• Optional; use only if risk aversion is to be included	Risk aversion coefficient	• EQ_OBJ

[1] The first row contains the parameter name, the second row contains in brackets the index domain over which the parameter is defined.
[2] This column gives references to related input parameters or sets being used in the context of this parameter as well as internal parameters/sets or result parameters being derived from the input parameter.
[3] This column lists the unit of the parameter, the possible range of its numeric value [in square brackets] and the inter-/extrapolation rules that apply.
[4] An indication of circumstances for which the parameter is to be provided or omitted, as well as description of inheritance/aggregation rules applied to parameters having the timeslice (**s**) index.
[5] Equations or variables that are directly affected by the parameter.
[6] Abbreviation i/e = inter-/extrapolation

Appendix C: Input parameters for Stochastic TIMES

Input parameter (Indexes)[1]	Related parameters[2]	Units / Ranges & Default values & Default inter-/extrapolation[3]	Instances[4] (Required / Omit / Special conditions)	Description	Affected equations or variables[5]
S_COM_PROJ (r,datayear,c, j,w)	**com_proj, com_fr**	• Commodity unit • [0, INF); default value: none • Default i/e[6]: standard	• Only applicable to demand commodities	Multiplier for projected annual demand of a commodity.	• Affects the RHS of the commodity balance constraint (EQ(l)_COMBAL
S_COM_CUMNET (r,y1,y2,c,bd, j,w)	**s _com_cumprd, com_cumprd, com_cumnet, rhs_combal**	• Commodity unit • [0, INF); default value: none • Default i/e: NA	• The years y1 and y2 may be any years of the set **allyear**;	Bound on the cumulative net amount of a commodity c between the years y1 and y2, within a region.	• Forces the variable VAR_COMPRD to be included in the balance (EQE_COMBAL) • Generates cumulative commodity constraint (EQ(l)_CUMPRD)
S _COM_CUMPRD (r,y1,y2,c,bd, j,w)	**s _com_cumnet, com_cumprd, com_cumnet, rhs_comprd**	• Commodity unit • [0, INF); default value: none • Default i/e: NA	• The years y1 and y2 may be any years of the set **allyear**;	Bound on the cumulative production of a commodity between the years y1 and y2, within a region.	• Forces the variable VAR_COMNET to be included in the balance (EQE_COMBAL) • Generates cumulative commodity constraint (EQ(l)_CUMNET)

[1] The first row contains the parameter name, the second row contains in brackets the index domain over which the parameter is defined.
[2] This column gives references to related input parameters or sets being used in the context of this parameter as well as internal parameters/sets or result parameters being derived from the input parameter.
[3] This column lists the unit of the parameter, the possible range of its numeric value [in square brackets] and the inter-/extrapolation rules that apply.
[4] An indication of circumstances for which the parameter is to be provided or omitted, as well as description of inheritance/aggregation rules applied to parameters having the timeslice (**s**) index.
[5] Equations or variables that are directly affected by the parameter.
[6] Abbreviation i/e = inter-/extrapolation

Input parameter (Indexes)[1]	Related parameters[2]	Units / Ranges & Default values & Default inter-/extrapolation[3]	Instances[4] (Required / Omit / Special conditions)	Description	Affected equations or variables[5]
S_COM_FR (r,datayear,c,s,j,w)	**com_fr**	• Units: none • Default value: 1 • Default i/e: standard	• Requires that corresponding deterministic parameter is defined	Multiplier for commodity timeslice fraction	• EQ_PTRANS, EQ_ACTEFF, EQ(l)_COMBAL
S _CAP_BND (r,datayear,p,bd,j,w)	**cap_bnd, ncap_bnd, spar_caplo, spar_capup**	• Capacity unit • [0, INF); default value: none • Default i/e: migrate to period year	• Since inter-/extrapolation is by default only migrated, the bound must be specified for each period desired, if no specific option regarding inter-/extrapolation is given.	Bound on total capacity of a process in a period.	• Causes the generation of a capacity transfer equation (EQ_CPT). Imposes a bound on the capacity variable (VAR_CAP).
S_DAM_COST (r,y,com,cur,j,w)	**dam_cost**	• Currency units per commodity units • [0, INF); • Default i/e: standard	• None	Damage cost on commodity net production	• EQ_OBJ VAR_COMNET
S_CM_MAXC (y,item,j,w)	**cm_maxc**	• GtC / °C • [0, INF); default value: none • Default i/e: none	• Since no inter-/extrapolation is done by default, the bound must be specified for each period desired, if no specific option regarding inter-/extrapolation is given.	Maximum allowed atmospheric CO_2 concentration (in GtC) in a given year	• Causes the generation of a concentration bound equation (EQ_CLIMAX)
S_CM_CONST (item, j,w)	**cm_const**	• None	• Item must be either ‘SIGMA1’ or ‘CS’	Climate module constants SIGMA1 or CS	• Temperature equations
S_FLO_CUM (r,p,c,y1,y2,bd,j,w)	**flo_cum, act_cum**	• Flow / activity unit • Default value: none	• None	Bound on cumulative flow or activity	• EQ_CUMFLO
S_FLO_FUNC (r,datayear,p,cg1,cg2, j,w)	**flo_func, flo_sum, act_eff**	• Units: none • Default value: 1 • Default i/e: standard	• Requires that corresponding deterministic parameter is defined	Multiplier for process transformation coeff.	• EQ_PTRANS, EQ_ACTEFF

Input parameter (Indexes)[1]	Related parameters[2]	Units / Ranges & Default values & Default inter-/extrapolation[3]	Instances[4] (Required / Omit / Special conditions)	Description	Affected equations or variables[5]
S_NCAP_AFS (r,datayear,p,s,j,w)	**ncap_af, ncap_afs, ncap_afa**	• Units: none • Default value: none • Default i/e: standard	• Requires that corresponding deterministic parameter is defined	Availability of process in timeslice s	• EC(l)_CAPACT • EQ(l)_CAFLAC
S_NCAP_COST (r,datayear,p,j,w)	**ncap_cost**	• Currency units • Default value: 1 • Default i/e: standard	• None	Multiplier for process investment cost	• EQ_OBJINV
S_UC_RHS (uc_n,lim,j,w)	**uc_rhs**	• None • [open]; default value: none • Default i/e: none	• Used in user constraints	RHS constant with bound type of lim of a user constraint.	• User constraints (EQ(l)_UC)
S_UC_RHSR (r,uc_n,lim,j,w)	**uc_rhsr**	• None • [open]; default value: none • Default i/e: none	• Used in user constraints	RHS constant with bound type of lim of a user constraint.	• User constraints (EQ(l)_UCR)
S_UC_RHST (uc_n,y,lim,j,w)	**uc_rhst**	• None • [open]; default value: none • Default i/e: migrate to period year	• Used in user constraints	RHS constant with bound type of lim of a user constraint.	• User constraints (EQ(l)_UCT)
S_UC_RHSRT (r,uc_n,y,lim,j,w)	**uc_rhsrt**	• None • [open]; default value: none • Default i/e: migrate to period year	• Used in user constraints	RHS constant with bound type of lim of a user constraint.	• User constraints (EQ(l)_UCRT)

Input parameter (Indexes)[1]	Related parameters[2]	Units / Ranges & Default values & Default inter-/extrapolation[3]	Instances[4] (Required / Omit / Special conditions)	Description	Affected equations or variables[5]
S_UC_RHSTS (uc_n,y,s,lim,j,w)	**uc_rhsts**	• None • [open]; default value: none • Default i/e: migrate to period year	• Used in user constraints	RHS constant with bound type of lim of a user constraint.	• User constraints (EQ(l)_UCTS)
S_UC_RHSRTS (r,uc_n,y,s,lim,j,w)	**uc_rhsrts**	• None • [open]; default value: none • Default i/e: migrate to period year	• Used in user constraints	RHS constant with bound type of lim of a user constraint.	• User constraints (EQ(l)_UCRTS)
S_UCOBJ (uc_n,w)	–	• None • [open]; default value: none • Default i/e: n/a	• Used for defining a user-defined objective function in Phase 1 of the tradeoff analysis facility	Weight coefficient of the objective component ***uc_n*** in analysis point ***w***	• Objective function (ES_EXPOBJ, EQ_OBJ)
UC_ACTBET (uc_n,r,y,p)	UC_FLOBET	• Dimensionless • [open]; default value: none • Default i/e: standard	• Used for incorporating risks when using the tradeoff analysis facility • UC_T_SUM can be used for aggregation over periods.	Risk parameter for energy corridor identified by uc_n	• EQG_UCMAX, EQG_UCSUMAX • These equations derive the maximum risk over regions & corridors.
UC_FLOBET (uc_n,r,y,p,cg)	UC_ACTBET	• Dimensionless • [open]; default value: none • Default i/e: standard	• Used for incorporating risks in tradeoff analysis • UC_R_EACH can be used for making the max. risk operator region-specific, UC_T_SUM can be used for aggregation over periods.	Risk parameter for fuel category identified by uc_n and cg	• EQG_UCMAX, EQG_UCSUMAX • These equations derive the maximum risk over processes and regions or over fuels.