DECIS

USER'S GUIDE

Gerd Infanger

A SYSTEM FOR SOLVING LARGE-SCALE
STOCHASTIC PROGRAMS

DECIS[*]USER'S GUIDE (Preliminary)

Gerd Infanger[†]

1997

Abstract

DECIS is a system for solving large-scale stochastic programs. It employs Benders decomposition and Monte Carlo sampling using importance sampling or control variates as variance reduction techniques. DECIS includes a variety of solution strategies and can solve problems with numerous stochastic parameters. For solving master and subproblems, DECIS interfaces with MINOS or CPLEX. This user's guide discusses how to run DECIS, the inputs needed and the outputs obtained, and gives a comprehensive description of the methods used.

[*] Copyright © 1989 – 1997 by Gerd Infanger. All rights reserved. The DECIS User's Guide is copyrighted and all rights are reserved. Information in this document is subject to change without notice and does not represent a commitment on the part of Gerd Infanger. The software described in this document is furnished under a license agreement and may be used only in accordance with the terms of this agreement.

[†] Dr. Gerd Infanger is an Associate Professor at Vienna University of Technology and a Visiting Professor in the Department of Engineering-Economic Systems and Operations Research at Stanford University.

Contents

1. Introduction

DECIS is a system for solving large-scale stochastic programs, programs, which include parameters (coefficients and right-hand sides) that are not known with certainty, but are assumed to be known by their probability distribution. It employs Benders decomposition and allows using advanced Monte Carlo sampling techniques. DECIS includes a variety of solution strategies, such as solving the universe problem, the expected value problem, Monte Carlo sampling within the Benders decomposition algorithm, and Monte Carlo pre-sampling. When using Monte Carlo sampling the user has the option of employing crude Monte Carlo without variance reduction techniques, or using as variance reduction techniques importance sampling or control variates, based on either an additive or a multiplicative approximation function. Pre-sampling is limited to using crude Monte Carlo only.

For solving linear and nonlinear programs (master and subproblems arising from the decomposition) DECIS interfaces with MINOS or CPLEX. MINOS, see Murtagh and Saunders (1983) [13], is a state-of-the-art solver for large-scale linear and nonlinear programs, and CPLEX, see CPLEX Optimization, Inc. (1989–1997) [4], is one of the fastest linear programming solvers available.

This document discusses how to use DECIS, what problems DECIS solves, what solution strategies can be used, how to set up the input files required, how to run the system, and how to interpret the output files. The remainder of this document is organized as follows: A brief discussion of how to run DECIS is given in Section 2; the class of problems DECIS can solve, and an overview of the solution strategies used for solving the problem, are introduced in Section 3; how to input a problem for DECIS, including the detailed format of the input files is described in Section 4; the DECIS output, including the detailed format of the output files and how to interpret the output, is discussed in Section 5. The input and output of DECIS is exemplified using an illustrative example, APL1P, discussed in Appendix A. Finally, the different solution strategies and techniques used for solving the problem are described mathematically and in detail in Section 6. As an example of a practical problem, Appendix B discusses results from a large problem in transportation, obtained from using DECIS.

2. How to run DECIS

You find DECIS installed in a subdirectory on your hard disk. To run DECIS, you type at the command prompt in the subdirectory

```
decis optimizer [mode]
```

where optimizer refers to the optimizer you want to choose for solving master and subproblems, and mode refers to the mode in which you want to run DECIS. If you want to choose MINOS type m as optimizer, if you want to choose CPLEX, type c. In the optimization mode, which is the default and does not need any input for [mode], DECIS solves a stochastic problem. In the evaluation mode, for which type eval for mode, DECIS evaluates a given first-stage solution by computing the corresponding total expected cost. For example,

```
decis m
```

runs DECIS using MINOS in optimization mode, and

```
decis c eval
```

runs DECIS using CPLEX in evaluation mode.

In order to run DECIS you should have a computer with 32 Megabytes of core memory or more. While the DECIS program itself uses little space on your hard disk, problem data and output files may be large, and you are advised to keep enough resources available on your hard disk.

You must have all input files ready in the decis-p subdirectory. The input files include the model specification (files MODEL.COR, MODEL.TIM, MODEL.STO), the parameter input (MODEL.INP), and the file specifying the initial seed (MODEL.ISE). When running DECIS with MINOS as the optimizer, you also need the MINOS specification file (MINOS.SPC).

3. What does DECIS do?

DECIS has two modes:

- the optimization mode for solving stochastic problems, and
- the evaluation mode for evaluating a given solution for the stochastic problem.

3.1. Optimization mode

In its optimization mode, DECIS solves two-stage stochastic linear programs with recourse:

$$\begin{array}{rcrcrcl}
\min\ z & = & cx & + & E\ f^\omega y^\omega & & \\
s/t & & Ax & & & = & b \\
 & & -B^\omega x & + & D^\omega y^\omega & = & d^\omega \\
 & & x, & & y^\omega & \geq & 0, \quad \omega \in \Omega.
\end{array}$$

where x denotes the first-stage, y^ω the second-stage decision variables, c represents the first-stage and f^ω the second-stage objective coefficients, A, b represent the coefficients and right hand sides of the first-stage constraints, and B^ω, D^ω, d^ω represent the parameters of the second-stage constraints, where the transition matrix B^ω couples the two stages. In the literature D^ω is often referred to as the technology matrix or recourse matrix. The first stage parameters are known with certainty. The second stage parameters are random parameters that assume outcomes labeled ω with probability $p(\omega)$, where Ω denotes the set of all possible outcome labels.

At the time the first-stage decision x has to be made, the second-stage parameters are only known by their probability distribution of possible outcomes. Later after x is already determined, an actual outcome of the second-stage parameters will become known, and the second-stage decision y^ω is made based on knowledge of the actual outcome ω. The objective is to find a feasible decision x that minimizes the total expected costs, the sum of first-stage costs and expected second-stage costs.

For discrete distributions of the random parameters, the stochastic linear program can be represented by the corresponding *equivalent deterministic linear program*:

$$\begin{array}{rrcrcrcrcrcl}
\min\ z = & cx & + & p^1 f y^1 & + & p^2 f y^2 & + & \cdots & + & p^W f y^W & & \\
s/t & Ax & & & & & & & & & = & b \\
 & -B^1 x & + & D y^1 & & & & & & & = & d^1 \\
 & -B^2 x & & & + & D y^2 & & & & & = & d^2 \\
 & \vdots & & & & & & \ddots & & & & \vdots \\
 & -B^W x & & & & & & & + & D y^W & = & d^W \\
 & x, & & y^1, & & y^2, & & \ldots, & & y^W & \geq & 0,
\end{array}$$

which contains all possible outcomes $\omega \in \Omega$. Note that for practical problems W is very large, e.g., a typical number could be 10^{20}, and the resulting equivalent deterministic linear problem is too large to be solved directly.

In order to see the two-stage nature of the underlying decision making process the folowing representation is also often used:

$$\begin{array}{rcrcl}
\min\ cx & + & E\ z^\omega(x) & & \\
Ax & & & = & b \\
x & & & \geq & 0
\end{array}$$

where

$$\begin{array}{rcrcl}
z^\omega(x) & = & \min\ f^\omega y^\omega & & \\
 & & D^\omega y^\omega & = & d^\omega + B^\omega x \\
 & & y^\omega & \geq & 0,\ \omega \in \Omega = \{1, 2, \ldots, W\}.
\end{array}$$

DECIS employs different strategies to solve two-stage stochastic linear programs. It computes an exact optimal solution to the problem or approximates the true optimal solution very closely and gives a confidence interval within which the true optimal objective lies with, say, 95% confidence.

3.2. Evaluation mode

In its evaluation mode, DECIS computes the expected optimal cost, corresponding to a given first-stage decision $\hat{x}$, i.e.,

$$c\hat{x} + E\ z^\omega(\hat{x}),$$

where

$$\begin{array}{rcrcl}
z^\omega(\hat{x}) & = & \min\ f^\omega y^\omega & & \\
 & & D^\omega y^\omega & = & d^\omega + B^\omega \hat{x} \\
 & & y^\omega & \geq & 0,\ \omega \in \Omega = \{1, 2, \ldots, W\},
\end{array}$$

and

$$A\hat{x} = b.$$

Note that also in the evaluation model DECIS solves subproblems in order to compute the expected second-stage cost $E\ z^\omega(\hat{x})$, for given $\hat{x}$. All different solution strategies available in the optimization mode can also be employed in the evaluation mode.

3.3. Representing uncertainty

It is favorable to represent the uncertain second-stage parameters in a structure. Using $V = (V_1, \dots, V_h)$ an h-dimensional independent random vector parameter that assumes outcomes $v^\omega = (v_1, \dots, v_h)^\omega$ with probability $p^\omega = p(v^\omega)$, we represent the uncertain second-stage parameters of the problem as functions of the independent random parameter V:

$$f^\omega = f(v^\omega), \quad B^\omega = B(v^\omega), \quad D^\omega = D(v^\omega), \quad d^\omega = d(v^\omega).$$

Each component V_i has outcomes $v_i^{\omega_i}$, $\omega_i \in \Omega_i$, where ω_i labels a possible outcome of component i, and Ω_i represents the set of all possible outcomes of component i. An outcome of the random vector

$$v^\omega = (v_1^{\omega_1}, \dots, v_h^{\omega_h})$$

consists of h independent component outcomes. The set

$$\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_h$$

represents the crossing of sets Ω_i. Assuming each set Ω_i contains W_i possible outcomes, $|\Omega_i| = W_i$, the set Ω contains $W = \prod W_i$ elements, where $|\Omega| = W$ represents the number of all possible outcomes of the random vector V. Based on independence, the joint probability is the product

$$p^\omega = p_1^{\omega_1} p_2^{\omega_2} \cdots p_h^{\omega_h}.$$

Let η denote the vector of all second-stage random parameters, e.g., $\eta = \text{vec}(f, B, D, d)$. The outcomes of η may be represented by the following general linear dependency model:

$$\eta^\omega = \text{vec}(f^\omega, B^\omega, d^\omega, d^\omega) = Hv^\omega, \quad \omega \in \Omega$$

where H is a matrix of suitable dimensions. DECIS can solve problems with such general linear dependency models. Section 4 discusses how to input H and v^ω.

3.4. Solving the universe problem

We refer to the universe problem if we consider all possible outcomes $\omega \in \Omega$ and solve the corresponding problem exactly. This is not always possible, because there may be too many possible realizations $\omega \in \Omega$. For solving the problem DECIS employs Benders decomposition, splitting the problem into a master problem, corresponding to the first-stage decision, and into subproblems, one for each $\omega \in \Omega$, corresponding to the second-stage decision. The details of the algorithm and techniques used for solving the universe problem are discussed in Section 6.1.

Solving the universe problem is referred to as strategy 4 (istrat = 4) in the parameter file (see Section 4.4). Use this strategy only if the number of universe scenarios is reasonably small. There is a maximum number of universe scenarios DECIS can handle, which depends on your particular installation. If you try to solve a model with more than the maximum number of universe scenarios, DECIS will stop with an error message.

3.5. Solving the expected value problem

The expected value problem results from replacing the stochastic parameters by their expectation. It is a linear program that can also easily be solved by employing a solver directly. Solving the expected value problem may be useful by itself (for example as a benchmark to compare the solution obtained from solving the stochastic problem), and it also may yield a good starting solution for solving the stochastic problem. DECIS solves the expected value problem using Benders decomposition. The details of generating the expected value problem and the algorithm used for solving it are discussed in Section 6.2.

Since in the decomposed mode there is only one (expected value) subproblem, solving the expected value problem is the fastest strategy that can be used. Therefore when setting-up a new model, solving the expected value problem is a good way of getting started and of finding possible mistakes in your model files.

To solve the expected value problem choose strategy 1 (istrat = 1) in the parameter input file.

3.6. Using Monte Carlo sampling

As noted above, for many practical problems it is impossible to obtain the universe solution, because the number of possible realizations $|\Omega|$ is way too large. The power of DECIS lies in its ability to compute excellent approximate solutions by employing Monte Carlo sampling techniques. Instead of computing the expected cost and the coefficients and the right-hand sides of the Benders cuts exactly (as it is done when solving the universe problem), DECIS, when using Monte Carlo sampling, estimates the quantities in each iteration using an independent sample drawn from the distribution of the random parameters. In addition to using crude Monte Carlo, DECIS uses importance sampling or control variates as variance reduction techniques.

The details of the algorithm and the different techniques used are described in Section 6.3. You can choose crude Monte Carlo, referred to as strategy 6 (istrat = 6), Monte Carlo importance sampling, referred to as strategy 2 (istrat = 2), or control variates, referred to as strategy 10 (istrat = 11). Both Monte Carlo importance sampling and control variates have been shown for many problems to give a better approximation compared to employing crude Monte Carlo sampling.

When using Monte Carlo sampling DECIS computes a close approximation to the true solution of the problem, and estimates a close approximation of the true optimal objective value. It also computes a confidence interval within which the true optimal objective of the problem lies, say with 95% confidence. The confidence interval is based on rigorous statistical theory. An outline of how the confidence interval is computed is given in Section 6.3.4. The size of the confidence interval depends on the variance of the second-stage cost of the stochastic problem and on the sample size used for the estimation. You can expect the confidence interval to be very small, especially when you employ importance sampling or control variates as a variance reduction technique.

When employing Monte Carlo sampling techniques you have to choose a sample size (set in the parameter file). Clearly, the larger the sample size the better will be the approximate solution DECIS computes, and the smaller will be the confidence interval for the true optimal objective value. The default value for the sample size is 100 (nsamples = 100).

Setting the sample size too small may lead to bias in the estimation of the confidence interval, therefore the sample size should be at least 30. There is a maximum value for the sample size DECIS can handle, which depends on your particular installation. If you try to choose a larger sample size than the maximum value, DECIS will stop with an error message.

3.7. Monte Carlo pre-sampling

We refer to pre-sampling when we first take a random sample from the distribution of the random parameters and then generate the approximate stochastic problem defined by the sample. The obtained approximate problem is then solved exactly using decomposition. This is in contrast to the way we used Monte Carlo sampling in the previous section, where we used Monte Carlo sampling in each iteration of the decomposition.

The details of the techniques used for pre-sampling are discussed in Section 6.4. DECIS computes the exact solution of the sampled problem using decomposition. This solution is an approximate solution of the original stochastic problem. Besides this approximate solution, DECIS computes an estimate of the expected cost corresponding to this approximate solution and a confidence interval within which the true optimal objective of the original stochastic problem lies with, say, 95% confidence. The confidence interval is based on statistical theory, its size depends on the variance of the second-stage cost of the stochastic problem and on the sample size used for generating the approximate problem. In conjunction with pre-sampling no variance reduction techniques are currently implemented.

Using Monte Carlo pre-sampling you have to choose a sample size. Clearly, the larger the sample size you choose, the better will be the solution DECIS computes, and the smaller will be the confidence interval for the true optimal objective value. The default value for the sample size is 100 (nsamples = 100). Again, setting the sample size as too small may lead to a bias in the estimation of the confidence interval, therefore the sample size should be at least 30. There is a maximum value for the sample size DECIS can handle, which depends on your particular installation. If you try to choose a larger sample size than the maximum value, DECIS will stop with an error message.

For using Monte Carlo pre-sampling choose strategy 8 (istrat = 8) in the parameter file.

3.8. Regularized decomposition

When solving practical problems, the number of Benders iterations can be quite large. In order to control the decomposition, with the hope to reduce the iteration count and the solution time, DECIS makes use of regularization. When employing regularization, an additional quadratic term is added to the objective of the master problem, representing the square of the distance between the best solution found so far (the incumbent solution) and the variable x. Using this term, DECIS controls the distance of solutions in different decomposition iterations.

For enabling regularization you have to set the corresponding parameters in the parameter file (ireg = 1). You also have to choose the value of the constant rho in the regularization term. The default is regularization disabled. Details of how DECIS carries out regularization are represented in Section 6.5.2.

Regularization has proven to be especially helpful for problems that need a large number of Benders iteration when solved without regularization. Problems that need only a small number of Benders iterations without regularization are not expected to improve much with regularization, and may need even more iterations with regularization.

4. Inputting the problem

The DECIS software inputs problems in the SMPS input format. SMPS stands for Stochastic Mathematical Programming System. SMPS, see Birge et al. (1989) [2], is an extension to the well known MPS (Mathematical Programming System) format, see International Business Machines (1988) [9], for deterministic problems. The SMPS format uses three files to define a stochastic problem: the core file, the time file, and the stochastic file.

The core file represents a deterministic version of the stochastic problem in MPS format, where the random parameters are represented by an arbitrary outcome. Instead of an arbitrary outcome, you may want to put the expected value of each of the stochastic parameters in the core file. The time file gives pointers as to which variables and constraints belong to the first stage and which variables and constraints belong to the second stage, using the names of rows and columns in the core file. The stochastic file contains all the information about the distribution of the stochastic parameters, where the stochastic parameters are identified by the row and/or column name as defined in the core file. You may want to use a modeling language, e.g., GAMS (Brooke, Kendrik, and Meeraus (1988), [3]) or AMPL (Fourer, Gay and Kernighan (1992) [8]) to formulate the problem and generate the SMPS input files automatically using the smps.pl program of Entriken and Stone (1997) [7].

In addition, parameters for controlling the execution of DECIS are represented in the parameter file, the initial seed for the random number generator is inputted via the initial seed file and, when using MINOS as the optimizer for solving master and subproblems, parameters to control the execution of MINOS as a subroutine of DECIS are inputted via the MINOS specification file. DECIS, in evaluation mode, evaluates a particular first-stage solution. You can input a first-stage solution for evaluation via the free format solution file.

Note the convention for file names: all files specifying a problem and its solution have the filename “MODEL” and are distinguished by their file extension.

4.1. The core file

The name of the core file is “MODEL.COR”. The core file represents a deterministic version of the problem in MPS format. We now give a brief description of how to implement a problem in the MPS format. In the MPS format variable names and constraint names may have up to eight characters of length. The core filecontaines the following cards and sections:

1. The NAME card — It has to be the first record of the file and contains the word “NAME” in the format (A4). After “name” you have 20 characters to write any information you want to identify the file. This information will appear then on the solution output of the solver.

2. The ROWS card — Initializes the rows section of the problem. It has to be the second record and contains the name “ROWS” in the format (A4). It is followed by

3. The Rows section — It contains the constraint names and the relation between the left hand side and the right hand side of each constraint. Use one record for each constraint in the format (1X, A1, 2X, A8), which means keep the first digit free, then use one digit for the relation between left hand and right hand side of the constraint, keep two digits free, and use up to 8 digits for the constraint name. Constraint names can consist of any characters and numbers including the special characters of the ASCII character set. For the relations between the left hand side and right hand side of the constraints use "E" for equal, "G" for greater equal, and "L" for less equal. *The first row in the list has to be the objective function.* The objective function is a free row, denoted by the relation sign "N". It is important that you input all first stage constraint names first and all second stage constraints thereafter. Within the first stage constraints and within the second stage constraints you may enter the rows in any order.

4. The COLUMNS card — Initializes the columns section of the problem. It has to follow the rows section and contains the name "COLU" in the format (A4). You can also write "COLUMNS", but only the first 4 characters will be processed. It is followed by

5. The Columns section — It contains the coefficients of the linear constraints. Only non-zero coefficients have to be considered and you can leave out the zero coefficients of the constraints. You can input the coefficients in the one or two column representation of the MPS format. In the one column representation, each record corresponds to one coefficient and has the format (4X, A8, 2X, A8, 2X, E12.0), which means, leave the first 4 digits blank, use the following 8 digits for the variable name, leave 2 digits blank, use the following 8 digits for the constraint name, leave 2 digits empty, and use up to 12 digits for the decimal representation of the non zero coefficient. In the two column representation, each record corresponds to two coefficients and has the format (4X, A8, 2X, A8, 2X, E12.0, 2X, A8, 2X, E12.0). You can represent another coefficient corresponding to the variable and another row in the same record. Leave after the first row name 2 digits blank, use the following 8 digits for another row name, leave 2 digits blank, and use up to 12 digits for the decimal representation of the non zero coefficient corresponding to the latter row name. You have to input the coefficients grouped by each variable, meaning in the order that all coefficients corresponding to a certain variable come one after the other. You can not switch between variables in the sense that you input some coefficients corresponding to one variable, then input some coefficients of another variable, and then again coefficients of the former variable. It is important that you input all first stage variables first and all second stage variables thereafter. Within the set of first stage variables and within the set of second stage variables you can input variables in any order you wish. The constraint names and corresponding coefficients associated with a variable can be in any order. Corresponding to each variable you do not need to input first stage row names and corresponding coefficients before second stage variables and corresponding coefficients. After the columns section follows

6. The RHS card — It initializes the right hand side section of the constraint. It has to

follow the columns section and contains the name “RHS ” in the format (A4). It is followed by

7. The right hand side section — It contains the values of the right hand sides of the constraints if they are non zero. You can leave out the right hand sides which are zero. There is one record for each non-zero right hand side in the format (4X, A8, 2X, A8, 2X, E12.0), which means leave the first 4 digits empty, then use 8 digits to represent a generic name for all the right hand sides (it has to be the same name for all right hand sides), then use 8 digits for the constraint name, leave 2 digits empty, and use up to 12 digits for the decimal representation of the right hand side value of the corresponding right hand side. You can choose any name for all your right hand sides, e.g., “RHS ” would be a proper name.

8. The RANGES card — It initializes the ranges section, which is used to input ranges on the right-hand side of rows. It has to follow directly the right hand side section and contains the name “RANG” in the format (A4). You can also write ”RANGES”, but only the first 4 characters will be processed. It is followed by

9. The ranges section – It contains the ranges of the right-hand sides of rows. Ranges determine upper and lower bounds on rows, the exact meaning depends on the type of the row and the sign of the right-hand side. The following table gives the resulting lower and upper bound of a row depending on its type and the sign of its right-hand side, where b denotes the value of the right-hand side and r the value of the range inputted in the ranges section.

row type	right-hand side sign	lower bound	upper bound
E	$+$	b	$b+\|r\|$
E	$-$	$b-\|r\|$	b
G	$+$ or $-$	b	$b+\|r\|$
L	$+$ or $-$	$b-\|r\|$	b

The format is (4X, A8, 2X, A8, 2X, E12.0), which means leave the first 4 digits empty, then use 8 digits to represent a generic name for all the ranges (it has to be the same name for all ranges), then use 8 digits for the constraint name, leave 2 digits empty, and use up to 12 digits for the decimal representation of the range value of the corresponding right hand side. You can choose any name for all your ranges, e.g., “RANGES ” would be a proper name.

10. The BOUNDS card — It initializes the bounds section, which is used to input bounds on the value of the variables. It has to follow the right hand side section and contains the name “BOUN” in the format (A4). You can also write ”BOUNDS”, but only the first 4 characters will be processed. It is followed by

11. The bounds section — It contains the values of the bounds on the variables. You can specify upper bounds, lower bounds and fixed bounds on any of the variables defined in the columns section in any order. The format is (1X, A2, 1X, A8, 2X, A8, 2X, E12.0), which means leave the first digit blank, then use 2 digits to input the qualifier

for upper, lower, or fixed bound, leave 1 digit blank, then use up to 8 digits to input a generic name for all the bounds (it has to be the same name for all bounds), leave 2 digits blank, then use up to eight digits for the variable name for which you want to specify the bound, leave 2 digits blank, and use up to 12 digits for the decimal representation of the bound value. The qualifier for upper bounds is "UP", for lower bounds "L", and for fixed bounds "FX".

12. The END card — Reports end of data and contains the name "ENDA" in the format (A4). You can also write "ENDATA", but only the first 4 characters will be processed.

There must be at least one row in the rows section, i.e., the objective function with row type N. Ranges and bounds sections are optional. If both ranges and bounds sections are present, the ranges section must appear first.

In the formulation of the core file, it is important that the objective function row contains at least one variable from the second stage. If you have a cost accounting equation in your model, you must account the first-stage and second-stage cost separately using one accounting equation for the first stage and another accounting equation for the second stage.

Example

As an example we consider the test problem APL1P, discussed in Appendix A. In the core file you can input any arbitrary realization of the stochastic parameters. The model core file in MPS representation reads as follows:

```
NAME          APL1P
ROWS
 N  ZFZF9999
 G  A0000001
 G  A0000002
 L  F0000001
 L  F0000002
 G  D0000001
 G  D0000002
 G  D0000003
COLUMNS
    X0000001  ZFZF9999   4.00000E+00
    X0000001  A0000001   1.00000E+00
    X0000001  F0000001  -1.00000E+00
    X0000002  ZFZF9999   2.50000E+00
    X0000002  A0000002   1.00000E+00
    X0000002  F0000002  -1.00000E+00
    Y0000011  ZFZF9999   4.30000E+00
    Y0000011  F0000001   1.00000E+00
    Y0000011  D0000001   1.00000E+00
    Y0000012  ZFZF9999   2.00000E+00
    Y0000012  F0000001   1.00000E+00
    Y0000012  D0000002   1.00000E+00
    Y0000013  ZFZF9999   0.50000E+00
    Y0000013  F0000001   1.00000E+00
    Y0000013  D0000003   1.00000E+00
    Y0000021  ZFZF9999   8.70000E+00
    Y0000021  F0000002   1.00000E+00
    Y0000021  D0000001   1.00000E+00
    Y0000022  ZFZF9999   4.00000E+00
    Y0000022  F0000002   1.00000E+00
    Y0000022  D0000002   1.00000E+00
    Y0000023  ZFZF9999   1.00000E+00
    Y0000023  F0000002   1.00000E+00
    Y0000023  D0000003   1.00000E+00
    UD000001  ZFZF9999   1.00000E+01
    UD000001  D0000001   1.00000E+00
    UD000002  ZFZF9999   1.00000E+01
    UD000002  D0000002   1.00000E+00
    UD000003  ZFZF9999   1.00000E+01
    UD000003  D0000003   1.00000E+00
RHS
    RHS       A0000001   1.00000E+03
    RHS       A0000002   1.00000E+03
    RHS       F0000001   0.000000+00
    RHS       F0000002   0.000000+00
    RHS       D0000001   1.00000E+03
    RHS       D0000002   1.00000E+03
    RHS       D0000003   1.00000E+03
ENDATA
```

In the example, “ZFZF9999” is the objective function row; “A0000001” and “A0000002” are first stage rows; “F0000001”, F0000002”, ”D0000001”, ”D0000002”, and ”D0000003” are second stage rows; “X0000001” and “X0000002” are first stage variables; “Y0000011”, . . ., “UD000003” are second stage variables. As a generic right hand side name “RHS ” was used.

In the core file the value 1000.0 was inputted for each of the stochastic demands d_i, $i = 1, 2, 3$, and the value 1.0 was inputted for the stochastic availability of the generators β_j, $j = 1, 2$. These values serve only as a place holder for non zero values. They are not further processed and have no impact on the solution. The distribution of the stochastic demands and the distribution of the availability of generators is inputted in the stochastic file described below. However, you can easily read all deterministic data of the problem from the core MPS file.

4.2. The time file

The name of the time file is “MODEL.TIM”. The time file is used to specify which variables and constraints belong to the first stage and which to the second stage. It closely mimics the MPS format for inputting variable names and row names. It consists of the following cards and sections:

1. The TIME card — It has to be the first record of the file and contains the word “TIME” in the format (A4). After “name” you have 20 characters to write any information you want to identify the file. This information is for your records only and will not be further processed.

2. The PERIODS card — It contains the word “PERI” in the format (A4). You can also specify “PERIODS”, but only the first four characters will be processed. The PERIODS card initializes

3. The periods section — It defines the first stage and the second stage variables and constraints through pointers to the first row and first column of each stage. There is one card for each stage in the format (4X, A8, 2X, A8, 16X, A8), which means leave the first 4 digits blank, then use up to 8 digits for the first column name, leave 2 digits blank, the use up to 8 digits for the first row name, leave 16 digits blank, and use up to 8 digits to indicate which stage the pointer refers to. The latter stage indicator is for your information only and is not further processed. The cards must be in order. You have to input the card for the first stage pointers first, followed by the card for the second stage pointers.

4. The END card – Reports end of data and contains the name "ENDA" in the format (A4). You can also write "ENDATA", but only the first four characters will be processed.

Note that the first row of the first stage constraints is the objective function, since the DECIS system treats the objective function as just another constraint. This may be different from other software packages solving stochastic problems, where the objective function is treated separately. These packages then require the first row name of the first stage constraints to be the "real" first row of the constraint matrix.

We have limited the discussion here to specifying two stages only. The SMPS format is able to specify multi-stage problems with more than two stages. For a description of the time file for more than two stages, please see Birge et al. (1989) [2].

Example

For the example of the stochastic electric power planning problem, APL1P, the corresponding time file is as follows:

```
TIME     APL1P
PERIODS   LP
    X0000001  ZFZF9999                PERIOD1
    Y0000011  F0000001                PERIOD2
ENDATA
```

"X000001" is the first of the first stage columns and "ZFZF9999" is the first of the first stage constraints; "Y0000011" is the first of the second stage columns, and "F0000001" is the first of the second stage constraints.

4.3. The stochastic file

The name of the stochastic file is "MODEL.STO". The stochastic file is used to specify the distribution of the stochastic parameters. The stochastic parameters are identified by the corresponding row and column names in the core file. The input format of the stochastic file mimics closely the MPS format for inputting row and column names and the corresponding parameters. In the current implementation of the DECIS system, *we consider only discrete random parameters*. Note that you can closely approximate continuous random parameters by discrete ones using a sufficient number of discrete realizations. However, there are plans to provide also for certain continuous random parameters (normal, uniform, and others) in the near future. DECIS can handle both, independent random parameters and certain types of dependent random parameters. The stochastic file consists of the following cards and sections:

1. The NAME card — It has to be the first record of the file and contains the word "NAME" in the format (A4). After "NAME" you have 20 characters to write any information you want to identify the file. This information is only for your records and will not be further processed.

2. The INDEPENDENT card — It contains the word "INDE" and a qualifier for the distribution in the format (A4, 10X, A8), which means input the word "INDE", leave 10 digits blank, and then use 8 digits for the distribution qualifier. Since we currently only provide for discrete distributions the only valid qualifier is the word "DISCRETE" Instead of "INDE" you can also specify "INDEPENDENT", but only the first 4 characters will be processed. (The first character of the word "DISCRETE" always has to be the 15-th digit of the record.) The independent card initializes

3. The independent section — It is used to specify all independent random parameters of the problem. Use one record for each possible realization of a random parameter and input the different independent random parameters one after the other. You cannot specify some realizations of one random parameter, then continue with inputting realization of another random parameter, and then continue inputting realizations of the first random parameter. The random parameters are identified by 2 identifiers. If the random parameter is a B matrix or a D matrix (including the objective function) coefficient the first identifier is the column name and the second identifier is the row name of that random coefficient. If the random parameter is a right-hand side value the first identifier is the right-hand side name in the core file (e.g., the string "RHS") and the second identifier is the row name of that right-hand side coefficient. If the uncertain parameter is a bound value (lower upper or fixed bound) the first identifier is the string "BND" and the second identifier is the column name of that bound coefficient. For specifying bounds we also need the keywords "LO" (for lower bound), "UP" (for upper bound), or "FX" (for fixed bound), which are in colums 3 and 4 of the record. The format for specifying a possible realization is (1X, A2, 1X, A8, 2X, A8, 2X, E12, 2X, A8, 2X, E12), which means leave the first digit blank, use two digits for the keyword (bounds only), leave one digit blank, then use up to 8 digits to enter the first identifier, leave 2 digits blank, then use up to eight digits to specify the second identifier, leave 2 digits blank, use up to 12 digits for the decimal representation of the value of the realization, leave 2 digits blank, use 8 digits to identify the stage at which the realization will become known, leave 2 digits blank, and use up to 12 digits for the decimal representation of the probability with which the realization occurs. The stage identifier is for your information only, and it is not further processed. Since we are concerned with two stage problems only, all random parameter realizations become known in the second stage. The probabilities of the different possible realizations of each random parameter should sum up to one. If for a particular random parameter the probabilities don't sum to one, DECIS divides the probability of each realization of that random parameter by the sum of the probabilities of all possible realizations

of that random parameter.

4. The BLOCKS card — It contains the word "BLOC" and a qualifier for the distribution in the format (A4, 10X, A8), which means input the word "BLOC", leave 10 digits blank, and then use 8 digits for the distribution qualifier. Since we currently only provide for discrete distributions the only valid qualifier is the word "DISCRETE". Instead of "BLOC" you can also specify "BLOCKS", but only the first 4 characters will be processed. The blocks card initializes a whole section used to specify all dependent random parameters of the problem. *DECIS considers additive dependency.* Dependent random parameters are represented as additive functions of one or several independent random parameters. The independent random parameters are called blocks. For each possible realization of a block, the coefficients for each of the dependent random parameters are specified. You can view these coefficients as the product of the outcome of the independent random parameter times the factor corresponding to the dependent random parameter.

5. The BL card — Initializes a possible realization of a block of random parameters. It has the format (1X, A2, 1X, A8, 2X, A8, 2X, E12), which means leave the first digit blank, use 2 digits to input the block identifier "BL", leave one digit blank, use up to 8 digits to input the block name, leave 2 digits blank, use up to 8 digits to specify the stage identifier, leave 2 digits blank and then use up to 12 digits for the decimal representation of the probability corresponding to the realization of the block. The stage identifier is for your information only, it is not further processed.

6. The BL section — It is used to specify all random parameters realizations of the block named in the "BL" card. Use one record for each dependent random parameter realization and input the different dependent random parameters one after the other. Like in the independent section, the random parameters are identified by 2 identifiers. If the random parameter is a B matrix or a D matrix (including the objective function) coefficient the first identifier is the column name and the second identifier is the row name of that random coefficient. If the random parameter is a right-hand side value the first identifier is the right-hand side name in the core file (e.g., the string "RHS") and the second identifier is the row name of that right-hand side coefficient. If the uncertain parameter is a bound value (lower, upper or fixed bound) the first identifier is the string "BND" and the second identifier is the column name of that bound coefficient. For specifying bounds we also need the keywords "LO" (for lower bound), "UP" (for upper bound), or "FX" (for fixed bound), which are in colums 3 and 4 of the record. The format for specifying a possible realization is (1X, A2, 1X, A8, 2X, A8, 2X, E12, 2X, A8, 2X, E12), which means leave the first digit blank, use two digits for the keyword (bounds only), leave one digit blank, then use up to 8 digits to enter the first identifier, leave 2 digits blank, then use up to eight digits to specify the second identifier, leave 2 digits blank, use up to 12 digits for the decimal representation of the value of the realization.

   After having inputted all dependent realizations associated with a certain outcome of the block, initialize a new realization of the block with corresponding probability using a new "BL" card. To specify a new realization of the same block, keep the same block name, but input the probability corresponding to the new realization of the block. Following the “BL” card input a new “BL section” to specify all dependent random parameters realizations corresponding to the new realization of the block. When specifying the first realization of a block you must input all coefficients corresponding to that realization. For any other realization, you only need to specify the coefficients that change from the previous realization. You do not have to input coefficients that remain the same from the previous realization. The values that don’t change are updated automatically. For all realizations of a block, the probabilities should sum up to one. If for a particular block the probabilities don’t sum to one, DECIS divides the probability of each realization of that block by the sum of the probabilities of all possible realizations of that block.

You can specify more than one block. Different blocks are distinguished by different block names. Each block is treated as an independent random parameter. Using the block section you can specify any additive dependency model: If two or more blocks contain the same dependent random parameter the coefficients of each of the outcomes of the blocks are added. A particular realization of a dependent random parameter is then generated as the the sum of the realizations of independent random parameters (blocks).

Examples

The first example considers the distribution of the model APL1P with 5 independent random parameters. The first and the second random parameter are coefficients in the B matrix, specified by the corresponding column and row names. The first random parameter has four outcomes, namely (-1, -0.9. -0.5, -0.1) with corresponding probabilities (0.2, 0.3, 0.4, 0.1), the second random parameter has five outcomes, namely (-1.0, -0.9, -0.7, -0.1, 0.0) with corresponding probabilities (0.1, 0.2, 0.5, 0.1, 0.1). The third, fourth and fifth random parameter are right hand side values, identified by the identifier “RHS ” and the row name of the random right hand side. All three right hand sides are identically distributed with four different discrete outcomes, namely (900, 1000, 1100, 1200) with corresponding probabilities (0.15, 0.45, 0.25, 0.15). The five independent random parameters give rise to 4*5*4*4*4 = 1280 possible combinations of different realizations. The stochastic file looks as follows:

```
NAME          APL1P
INDEP         DISCRETE
    X0000001  F0000001        -1.0     PERIOD2      0.2
    X0000001  F0000001        -0.9     PERIOD2      0.3
    X0000001  F0000001        -0.5     PERIOD2      0.4
    X0000001  F0000001        -0.1     PERIOD2      0.1
*
    X0000002  F0000002        -1.0     PERIOD2      0.1
    X0000002  F0000002        -0.9     PERIOD2      0.2
    X0000002  F0000002        -0.7     PERIOD2      0.5
    X0000002  F0000002        -0.1     PERIOD2      0.1
    X0000002  F0000002        -0.0     PERIOD2      0.1
*
    RHS       D0000001       900.0     PERIOD2     0.15
    RHS       D0000001      1000.0     PERIOD2     0.45
    RHS       D0000001      1100.0    PERIOD2     0.25
    RHS       D0000001      1200.0    PERIOD2     0.15
*
    RHS       D0000002       900.0    PERIOD2     0.15
    RHS       D0000002      1000.0    PERIOD2     0.45
    RHS       D0000002      1100.0    PERIOD2     0.25
    RHS       D0000002      1200.0    PERIOD2     0.15
*
    RHS       D0000003       900.0    PERIOD2     0.15
    RHS       D0000003      1000.0    PERIOD2     0.45
    RHS       D0000003      1100.0    PERIOD2     0.25
    RHS       D0000003      1200.0    PERIOD2     0.15
ENDATA
```

The second example considers 3 independent random parameters. Tthe first two are the same as above, and the third one is called “BLOCK1” and is a block controlling the outcomes of 3 dependent random parameters. The three dependent random parameters are right hand side values, specified by the identifier “RHS” and the row name of the corresponding random right hand side. The block has two outcomes, each occurring with probability 0.5. For the first outcome the three dependent demands “D0000001”,“D0000002”,“D0000003” take on the values 1100, 1100, 1100, and for the second outcome the values 900, 900, 900, respectively. The two independent random parameters and the independent block generate 4*5*2 = 40 possible combinations of realizations. There are two independent and three dependent random variables specified in the example. The corresponding stochastic file is:

```
NAME          APL1PC
INDEP         DISCRETE
    X0000001  F0000001        -1.0    PERIOD2      0.2
    X0000001  F0000001        -0.9    PERIOD2      0.3
    X0000001  F0000001        -0.5    PERIOD2      0.4
    X0000001  F0000001        -0.1    PERIOD2      0.1
    X0000002  F0000002        -1.0    PERIOD2      0.1
    X0000002  F0000002        -0.9    PERIOD2      0.2
    X0000002  F0000002        -0.7    PERIOD2      0.5
    X0000002  F0000002        -0.1    PERIOD2      0.1
    X0000002  F0000002        -0.0    PERIOD2      0.1
BLOCKS        DISCRETE
 BL BLOCK1    PERIOD2          0.5
    RHS       D0000001      1100.0
    RHS       D0000002      1100.0
    RHS       D0000003      1100.0
 BL BLOCK1    PERIOD2          0.5
    RHS       D0000001       900.0
    RHS       D0000002       900.0
    RHS       D0000003       900.0
ENDATA
```

The third example shows how to specify a general additive dependency model using blocks. Suppose there are two independent random parameters driving the electric power demand, e.g., temperature and economic activity. These independent random parameters are referred to as “BLOCK1” and “BLOCK2”. Each of these blocks has 2 realizations, “BLOCK1” with corresponding probabilities 0.5, and 0.5; “BLOCK2”with corresponding probabilities 0.2, and 0.8. Each of the blocks controls the same dependent random parameters “D0000001”,“D0000002”,“D0000003”. “BLOCK1” in its first realization generates right hand side values of (1100, 900, 500), and in its second realization the values (300, 400, 200); “BLOCK2” in its first realization the values (200, 300, 500) and in its second realization the values (100, 150, 300). When generating realizations, DECIS adds the coefficients of the dependent random parameters regarding different blocks. In the example, there are 2*2 = 4 possible combinations of realizations of the dependent parameters, namely (1300, 1200, 1000), (1000, 1050, 800), (500, 700, 700), and (400, 550, 500), generated as the sum of the coefficients for each of the two blocks. The corresponding stochastic file reads as follows:

```
NAME          APL1PCA
BLOCKS        DISCRETE
 BL BLOCK1   PERIOD2           0.5
    RHS       D0000001      1100.0
    RHS       D0000002       900.0
    RHS       D0000003       500.0
 BL BLOCK1   PERIOD2           0.5
    RHS       D0000001       300.0
    RHS       D0000002       400.0
    RHS       D0000003       200.0
 BL BLOCK2   PERIOD2           0.2
    RHS       D0000001       200.0
    RHS       D0000002       300.0
    RHS       D0000003       500.0
 BL BLOCK2   PERIOD2           0.8
    RHS       D0000001       100.0
    RHS       D0000002       150.0
    RHS       D0000003       300.0
ENDATA
```

4.4. The parameter file

In the parameter input file you can specify parameters regarding the solution algorithm used and control the output of the DECIS program. There is a record for each parameter you want to specify. Each record consists of the value of the parameter you want to specify and the keyword identifying the parameter, separated by a blank character or a comma. You may specify parameters with the following keywords: “istrat”, “nsamples”, “nzrows”, “maxit”, “iwrite”, “ibug”, “iscratch”, “iresamp”, “iappr”, “ireg”, “rho”, “tolben”, and “tolw” *in any order*. Each keyword can be specified in lower case or upper case text in the format (A10). Since DECIS reads the records in free format you don’t have to worry about the format, but some computers require that the text is inputted in quotes. Parameters that are not specified in the parameter file automatically assume their default values.

istrat — Defines the solution strategy used. The default value is istrat = 3.

istrat = 1 Solves the expected value problem. All stochastic parameters are replaced by their expected values and the corresponding deterministic problem is solved using decomposition.

istrat = 2 Solves the stochastic problem using Monte Carlo importance sampling. You have to additionally specify what approximation function you wish to use, and the sample size used for the estimation, see below.

istrat = 3 Refers to istrat = 1 plus istrat = 2. First solves the expected value problem using decomposition, then continues and solves the stochastic problem using importance sampling.

istrat = 4 Solves the stochastic universe problem by enumerating all possible combinations of realizations of the second-stage random parameters. It gives you the exact solution of the stochastic program. This strategy may be impossible, because there may be way too many possible realizations of the random parameters. There is a maximum number of possible universe scenarios DECIS can solve, which is defined at compilation.

istrat = 5 Refers to istrat = 1 plus istrat = 4. First solves the expected value problem using decomposition, then continues and solves the stochastic universe problem by enumerating all possible combinations of realizations of second-stage random parameters.

istrat = 6 Solves the stochastic problem using crude Monte Carlo sampling. No variance reduction technique is applied. This strategy is especially useful if you want to test a solution obtained by using the evaluation mode of DECIS. You have to specify the sample size used for the estimation. There is a maximum sample size DECIS can handle. However, this maximum sample size does not apply when using crude Monte Carlo. Therefore, in this mode you can specify very large sample sizes, which is useful when evaluating a particular solution.

istrat = 7 Refers to istrat = 1 plus istrat = 6. First solves the expected value problem using decomposition, then continues and solves the stochastic problem using crude Monte Carlo sampling.

istrat = 8 Solves the stochastic problem using Monte Carlo pre-sampling. A Monte Carlo sample out of all possible universe scenarios, sampled from the original probability distribution, is taken, and the corresponding "sample problem" is solved using decomposition.

istrat = 9 Refers to istrat = 1 plus istrat = 10. First solves the expected value problem using decomposition, then continues and solves the stochastic problem using Monte Carlo pre-sampling.

istrat = 10 Solves the stochastic problem using control variates. You also have to specify what approximation function and what sample size should be used for the estimation.

istrat = 11 Refers to istrat = 1 plus istrat = 8. First solves the expected value problem using decomposition, then continues and solves the stochastic problem using control variates.

nsamples — Sample size used for the estimation. It should be set greater or equal to 30 in order to fulfill the assumption of large sample size used for the derivation of the probabilistic bounds. The default value is nsamples = 100.

nzrows — Number of rows reserved for cuts in the master problem. It specifies the maximum number of different cuts DECIS maintains during the course of the decomposition algorithm. DECIS adds one cut during each iteration. If the iteration count exceeds nzrows, then each new cut replaces a previously generated cut, where the cut is replaced that has the maximum slack in the solution of the (pseudo) master. If nzrows is specified as too small then DECIS may not be able to compute a solution and stops with an error message. If nzrows is specified as too large the solution time will increase. As an approximate rule set nzrows greater than or equal to the number of first-stage variables of the problem. The default value is nzrows = 100.

maxit — Specifies the maximum number of Benders iterations DECIS uses for solving the problem. After maxit is reached, DECIS stops the decomposition algorithm and reports the best solution found so far. The default value is maxit = 1000.

iwrite — Specifies whether the optimizer invoked for solving subproblems writes output or not. The default value is iwrite = 0.

iwrite = 0 No optimizer output is written.

iwrite = 1 Optimizer output is written to the file "MODEL.MO". The output level of the output can be specified using the optimizer options. It is intended as a debugging device. If you set iwrite = 1, for every master problem and for every subproblem solved the solution output is written. For large problems and large sample sizes the file"MODEL.MO" may become very large, and the performance of DECIS may slow down.

ibug — Specifies the detail of debug output written by DECIS. The output is written to the file "MODEL.SCR", but can also be redirected to the screen by a separate parameter. The higher you set the number of ibug the more output DECIS will write. The parameter is intended to help debugging a problem and should be set to ibug = 0 for normal operation. For large problems and large sample sizes the file "MODEL.SCR" may become very large, and the performance of DECIS may slow down. The default value is ibug = 0.

ibug = 0 This is the setting for which DECIS does not write any debug output.

ibug = 1 In addition to the standard output, DECIS writes the solution of the master problem on each iteration of the Benders decomposition algorithm. Thereby it only writes out variable values which are nonzero. A threshold tolerance parameter for writing solution values can be specified, see below.

ibug = 2 In addition to the output of ibug = 1, DECIS writes the scenario index and the optimal objective value for each subproblem solved. In the case of solving the universe problem, DECIS also writes the probability of the corresponding scenario.

ibug = 3 In addition to the output of ibug = 2, DECIS writes information regarding importance sampling. In the case of using the additive approximation function, it reports the expected value for each i-th component of $\bar{\Gamma}_i$, the individual sample sizes N_i, and results from the estimation process. In the case of using the multiplicative approximation function it writes the expected value of the approximation function $\bar{\Gamma}$ and results from the estimation process.

ibug = 4 In addition to the output of ibug = 3, DECIS writes the optimal dual variables of the cuts on each iteration of the master problem.

ibug = 5 In addition to the output of ibug = 4, DECIS writes the coefficients and the right-hand side of the cuts on each iteration of the decomposition algorithm. In addition it checks if the cut computed is a support to the recourse function (or estimated recourse function) at the solution $\hat{x}^k$ at which it was generated. If it turns out that the cut is not a support, DECIS writes out the value of the (estimated) cut and the value of the (estimated) second stage cost at $\hat{x}^k$.

ibug = 6 In addition to the output of ibug = 5, DECIS writes a dump of the master problem and the subproblem in MPS format after having decomposed the problem specified in the core file. The dump of the master problem is written to the file "MODEL.FST" and the dump of the subproblem is written to the file "MODEL.SND". DECIS also writes a dump of the original problem, as it was read from the core file, to the file "MODEL.ORI".

ibug = 7 In addition to the output of ibug = 6, DECIS writes out information about the right-hand side vector passed to the subproblem, i.e., the values of b^ω, the product of $B^\omega \hat{x}^k$ and the resulting right-hand side $b^\omega + B^\omega \hat{x}^k$.

ibug = 8 In addition to the output of ibug = 7, DECIS echoes the information it read when inputting the stochastic file ("MODEL.STO") into the file "MODEL. SPR", and writes out on each iteration of the decomposition algorithm and for each subproblem solved the index of the realization of each random parameter that made up the scenario of the subproblem.

iscratch — Specifies the internal unit number to which the standard and debug output is written. The default value is iscratch = 17, where the standard and debug output is written to the file "MODEL.SCR". Setting iscratch = 6 redirects the output to the screen. Other internal unit numbers could be used, e.g., the internal unit number of the printer, but this is not recommended. The default value is iscratch = 17.

iresamp — Specifies weather DECIS evaluates the optimal solution using an independent sample or not, in order to compute an independent upper bound on the optimal objective of the problem. The parameter is only relevant when you use a solution strategy where Monte Carlo sampling is involved, i.e., istrat = 2, 3, 6, 7, 8, 9, 10, or 11. The default value is iresamp = 1.

iresamp = 0 The solution found is not evaluated using an independent sample. This setting is for debugging purposes only and should not be used for actual problem solving.

iresamp = 1 After completion of the decomposition algorithm DECIS draws an independent sample using the sampling strategy specified and independently evaluates the solution found to compute a probabilistic upper bound.

iappr — Specifies the type of approximation function used for carrying out importance sampling or control variates. The parameter is only relevant for solution strategies istrat = 2, 3, 10 and 11. The default value is iappr = 1.

iappr = 1 Specifies that the additive marginal cost model is used.

iappr = 2 Specifies that the multiplicative marginal cost factor model is used.

ireg — Specifies whether or not DECIS uses regularized decomposition for solving the problem. This option is considered if MINOS is used as a master and subproblem solver, and is not considered if using CPLEX, since regularized decomposition uses a nonlinear term in the objective. The default value is ireg = 0.

ireg = 1 Specifies that regularized decomposition is used.

ireg = 0 Specifies that no regularization is used.

rho — Specifies the value of the ρ parameter of the regularization term in the objective function. You will have to experiment to find out what value of rho works best for the problem you want to solve. There is no rule of thumb as to what value should be chosen. In many cases it has turned out that regularized decomposition reduces the iteration count if standard decomposition needs a large number of iterations. The default value is rho = 1000.

tolben — Specifies the tolerance for stopping the decomposition algorithm. The parameter is especially important for deterministic solution strategies, i.e., 1, 4, 5, 8, and 9. Choosing a very small value of tolben may result in a significantly increased number of iterations when solving the problem. The default value is 10^{-7}.

tolw — Specifies the nonzero tolerance when writing debug solution output. DECIS writes only variables whose values are nonzero, i.e., whose absolute optimal value is greater than or equal to tolw. The default value is 10^{-9}.

Example

In the following example the parameters istrat = 7, nsamples = 200, nzrows = 200, and maxit = 3000 are specified. All other parameters are set at their default values. DECIS first solves the expected value problem and then the stochastic problem using crude Monte Carlo sampling with a sample size of nsamples = 200. DECIS reserves space for a maximum of nzrows = 50 cuts. The maximum number of Benders iterations is specified as maxit = 3000.

```
7       "ISTRAT"
200     "NSAMPLES"
50      "NZROWS"
3000    "MAXIT"
```

4.5. The initial seed file

In the initial seed file you specify the initial seed for the pseudo random number generator used in DECIS, and the number of times you want to solve the problem. The name of the initial seed file is “MODEL.ISE”. It consists of only two lines, where the first line contains in free format the value of the parameter iseed, and the second line in free format the value of the parameter irep.

iseed — Specifies the initial seed given to the pseudo random parameter. You can specify any integer number greater than 0 and less than 2147483647.

irep — Specifies the number of times you want to solve the problem in one run of DECIS. DECIS has the option of solving a problem repeatedly many times with different seeds, in order to be able to collect statistics about the solutions and the probabilistic bounds obtained from the individual runs. The feature is important when using Monte Carlo sampling based strategies for solving the problem. The outputs of the individual runs are reported in the file “MODEL.REP”.

Example

In the following example an initial seed of iseed = 12783452, and a number of irep = 1 is specified. DECIS only reads the first number in each record; you can use the remainder of each record to write any comments you wish to add. They will not be processed.

```
12783452      ISEED: INITIAL SEED
1             IREP: # OF REPLICATIONS
```

4.6. The MINOS specification file

When you use MINOS as the optimizer for solving the master and the subproblems, you must specify optimization parameters in the MINOS specification file “MINOS.SPC”. Each record of the file corresponds to the specification of one parameter and consists of a keyword and the value of the parameter in free format. Records having a “*” as their first character are considered as comment lines and are not further processed. For a detailed description of these parameters, see the MINOS Users’ Guide (Murtagh and Saunders (1983) [13]. The following parameters should be specified with some consideration:

AIJ TOLERANCE — Specifies the nonzero tolerance for constraint matrix elements of the problem. Matrix elements a_{ij} that have a value for which $|a_{ij}|$ is less than “AIJ TOLERANCE” are considered by MINOS as zero and are automatically eliminated from the problem. It is wise to specify “AIJ TOLERANCE 0.0 ”

SCALE — Specifies MINOS to scale the problem (“SCALE YES”) or not (“SCALE NO”). It is wise to specify “SCALE NO”.

ROWS — Specifies the number of rows in order for MINOS to reserve the appropriate space in its data structures when reading the problem. “ROWS” should be specified as the number of constraints in the core problem or greater. MINOS terminates with an error message if “ROWS” is specified to small to read the problem, or if MINOS is not compiled for a problem the size asked for in the specification file.

COLUMNS — Specifies the number of columns in order for MINOS to reserve the appropriate space in its data structures when reading the problem. “COLUMNS” should be specified as the number of variables in the core problem or greater. MINOS terminates with an error message if “COLUMNS” is specified too small to read the problem, or if MINOS is not compiled for a problem the size asked for in the specification file.

ELEMENTS — Specifies the number of nonzero matrix coefficients in order for MINOS to reserve the appropriate space in its data structures when reading the problem. “ELEMENTS” should be specified as the number of nonzero matrix coefficients in the core problem or greater. MINOS terminates with an error message if “ELEMENTS” is specified to small to read the problem, or if MINOS is not compiled for a problem the size asked for in the specification file.

Example

The following example represents typical specifications for running DECIS with MINOS as the optimizer.

```
BEGIN SPECS
PRINT LEVEL                  1
LOG FREQUENCY               10
SUMMARY FREQUENCY           10
MPS FILE                    12
ROWS                     20000
COLUMNS                  50000
ELEMENTS                100000
ITERATIONS LIMIT         30000
*
FACTORIZATION FREQUENCY  100
AIJ TOLERANCE            0.0
*
SCALE                       NO
END OF SPECS
```

4.7. The free format solution file

In the evaluation mode, DECIS evaluates the solution that you have provided in the free format solution file “MODEL.STA”. Using this file you can specify values for the first-stage variables (i.e., the variables of the master problem) identified by their names in the core file in any order and in free format. You only need to specify nonzero values for variables. Variables for which no value is specified in the free format solution file are automatically assigned the value zero. There is one record per variable. Each record contains the variable name and the value separated by a blank character or a comma. The variable names are read in the format (A8). You do not need to worry about the format, but some computers require that the text is inputted in quotes.

Example

The following example concerns a free format solution file for the problem APL1P, where the solution $\hat{x}_1 = 1500$, and $\hat{x}_2 = 2000$ is specified. A record that starts with “*” as the first character is assumed to be comment line and is not further processed. Also you can input any text you wish, as long as it is separated by a blank character or a comma from the right of the variable value, and it will not be processed.

```
"X0000001"        1500.0
"X0000002"        2000.0
```

5. DECIS output

While running DECIS outputs important information about the progress of the execution to your computer screen. After successfully having solved a problem, DECIS outputs its optimal solution into the solution output file “MODEL.SOL”. However, there are a number of other files that contain useful information about the problem and its solution. The debug output file “MODEL.SCR” contains important information about the optimization run, the free format solution file “MODEL.STA” contains the best solution found in free format, the optimizer output file “MODEL.MO” contains the solution output from the optimizer called as a subroutine for solving master and subproblems, the problem output files “MODEL.FST” and “MODEL.SND” contain the decomposed first-stage and second-stage problem, respectively, in MPS format, the stochastic output file “MODEL.SPR” mirrors the information read from the stochastic file, and the multiple replications output file “MODEL.REP” contains summary information from repeated runs of one and the same problem. In the following we discuss how to interpret the screen output, the solution report and the information in each of the output files.

5.1. The screen output

The output to the screen allows you to observe the progress in the execution of a DECIS run. After the program logo and the copyright statement, you see four columns of output beeing written to the screen as long as the program proceeds. The first column (from left to right) represents the iteration count, the second column the lower bound (the optimal objective of the master problem), the third column the best upper bound (exact value or estimate of the total expected cost of the best solution found so far), and the fourth column the current upper bound (exact value or estimate of the total expected cost of current solution). After successful completion, DECIS quits with “Normal Exit”, otherwise, if an error has been encountered, the programs stops with the message “Error Exit”.

Example

When solving the illustrative example APL1P using strategy istrat = 5, we obtain the following report on the screen:

```
 T H E   D E C I S   S Y S T E M

 Benders DEComposition and Importance Sampling

 Copyright (c) 1989  -- 1997 by Dr. Gerd Infanger

 All rights reserved.
```

```
iter              lower          best upper       current upper

   0        -0.9935E+06
   1        -0.4626E+06          0.2590E+05          0.2590E+05
   2         0.2111E+05          0.2590E+05          0.5487E+06
   3         0.2170E+05          0.2590E+05          0.2697E+05
   4         0.2368E+05          0.2384E+05          0.2384E+05
   5         0.2370E+05          0.2384E+05          0.2401E+05
   6         0.2370E+05          0.2370E+05          0.2370E+05

iter              lower          best upper       current upper

   6         0.2370E+05
   7         0.2403E+05          0.2470E+05          0.2470E+05
   8         0.2433E+05          0.2470E+05          0.2694E+05
   9         0.2441E+05          0.2470E+05          0.2602E+05
  10         0.2453E+05          0.2470E+05          0.2499E+05
  11         0.2455E+05          0.2470E+05          0.2483E+05
  12         0.2461E+05          0.2467E+05          0.2467E+05
  13         0.2461E+05          0.2467E+05          0.2469E+05
  14         0.2461E+05          0.2465E+05          0.2465E+05
  15         0.2463E+05          0.2465E+05          0.2467E+05
  16         0.2463E+05          0.2465E+05          0.2465E+05
  17         0.2464E+05          0.2465E+05          0.2465E+05
  18         0.2464E+05          0.2464E+05          0.2464E+05
  19         0.2464E+05          0.2464E+05          0.2464E+05
  20         0.2464E+05          0.2464E+05          0.2464E+05
  21         0.2464E+05          0.2464E+05          0.2464E+05
  22         0.2464E+05          0.2464E+05          0.2464E+05

Normal Exit
```

After the optimal objective (initial lower bound) of the relaxed master on iteration 0, you see the lower, best and current upper bounds from solving the expected value problem (strategy istrat = 1), where on iteration 6 the lower bound, the best upper bound and the current upper bound converged to the optimal objective of 0.2370E+05. Then, after showing the optimal objective of the master problem in iteration 6 again (0.2370E+05) you see the development of the lower, best and current upper bounds for solving the universe case (strategy istrat = 4), until the decomposition converges to the optimal value of 0.2464E+05 for the lower bound, the best upper bound and the current upper bound. Note that on each iteration for strategy istrat = 5 (solving the universe case) 1280 subproblems are solved. Normal Exit indicates the successful solution of the problem.

5.2. The solution output file

The solution output file contains the solution report from the DECIS run. Its name is "MODEL.SOL". The file contains the best objective function value found, the corresponding values of the first-stage variables, the corresponding optimal second-stage cost, and a lower and an upper bound on the optimal objective of the problem. In addition, the number of universe scenarios and the settings for the stopping tolerance are reported. In the case of using a deterministic strategy for solving the problem, exact values are reported. When using Monte Carlo sampling, estimated values, their variances, and the sample size used for the estimation are reported. Instead of exact upper and lower bounds, probabilistic upper and lower bounds, and a 95% confidence interval, within which the true optimal solution lies with 95% confidence, are reported.

We continue to discuss the solution output for various strategies using the illustrative example APL1P.

Example

When solving the first the expected value problem and then the universe problem using strategy istrat = 5, we obtain the following solution output report.

```
 EXPECTED VALUE SOLUTION:
 ========================
  ITERATION           :           6
  OPTIMAL SOLUTION :
  OBJECTIVE:        2.37001E+04
        1 X0000001  1.52941E+03
        2 X0000002  1.62500E+03
        3 THETA     1.35200E+04
  ZSUB:             1.35200E+04
  NUMBER OF UNIVERSE SCENARIOS:           1.28000E+03
  TOLERANCE LEVEL:                        1.00000E-07

   EXPECTED TOTAL COST:                   2.46985E+04

-UNIVERSE- SOLUTION:
====================
 ITERATION           :          22
 OPTIMAL SOLUTION :
 OBJECTIVE:        2.46423E+04
       1 X0000001  1.80000E+03
       2 X0000002  1.57143E+03
       3 THETA     1.35137E+04
 ZSUB:             1.35137E+04
 NUMBER OF UNIVERSE SCENARIOS:           1.28000E+03
 TOLERANCE LEVEL:                        1.00000E-07
```

The report first displays the optimal expected value solution, obtained after 6 iterations, having 23700 as the optimal objective value, and the optimal variable values $\hat{x}_1 = 1529.4$, $\hat{x}_2 = 1652.0$, and $\hat{\theta} = 13520$. The optimal second-stage cost corresponding to the solution is labeled "ZSUB" and has the value 13520. DECIS further reports 1280 as the number of universe scenarios and 10^{-7} as the stopping tolerance of the decomposition algorithm.

DECIS then continues to solve the universe problem, starting from the optimal solution of the expected value problem. On the first iteration solving the universe problem, it evaluates the expected value solution by calculating its true expected cost in the stochastic environment. These costs are labeled as "EXPECTED TOTAL COST". This evaluation yields the same result as running DECIS in its evaluation mode with strategy istrat = 4 to evaluate the optimal solution from the expected value problem, inputted in the free format solution file "MODEL.STA".

Then the file displays the optimal universe solution, obtained after 22 iterations (including the 6 iterations from the expected value solution), having 24642.3 as the optimal objective value, and the optimal variable values $\hat{x}_1 = 1800.0$, $\hat{x}_2 = 1514.3$, and $\hat{\theta} = 13513.7$. "ZSUB", the true optimal expected second-stage cost is reported as 13513.7.

Example

When solving the stochastic problem using Monte Carlo importance sampling (istrat = 2) we obtain the following output:

```
STOCHASTIC SOLUTION:
====================
(IMPORTANCE SAMPLING)
 ITERATION           :          10
 OPTIMAL SOLUTION :
 OBJECTIVE:        2.45230E+04
       1 X0000001  2.06327E+03
       2 X0000002  1.36976E+03
       3 THETA     1.29822E+04
 ZSUB:             1.28455E+04
 95% CON. INTERVAL      :  2.44699E+04   2.46621E+04
 95% CON. INTERVAL %    : -2.16525E-01   5.67259E-01

 INITIAL SEED:                              12783452
 SAMPLE SIZE:                                    200
 NUMBER OF UNIVERSE SCENARIOS:           1.28000E+03
 TOLERANCE LEVEL:                        1.00000E-07
```

The report displays the best stochastic solution found after 10 iterations of the decomposition algorithm using importance sampling as having as variable values $\hat{x}_1 = 2063.7$, $\hat{x}_2 = 1397.6$, and $\hat{\theta} = 12982.2$. The estimated optimal expected second-stage cost, labeled "ZSUB", is computed as $\tilde{z}(\hat{x}) = 12845.5$. The optimal solution of the pseudo master problem $\tilde{v} = 24523.0$, and a 95% confidence interval is given as $24469.9 \leq z^* \leq 24662.1$, which translates into a probabilistic lower bound of 0.21% and a probabilistic upper bound of 0.56 % of the optimal pseudo master objective value. DECIS further reports 12783452 as the initial seed given to the pseudo random number generator at the start of the algorithm, 1280 as the number of universe scenarios of the problem, 200 as the sample size used, and 10^{-7} as the stopping tolerance of the decomposition algorithm.

5.3. The debug output file

The debug output file contains the standard output of a run of DECIS containing important information about the problem, its parameters, and its solution. It also contains any error messages that may occur during a run of DECIS. In the case that DECIS does not complete a run successfully, the cause of the trouble can usually be located using the information in the debug output file. If the standard output does not give enough information you can set the debug parameter ibug in the parameter input file to a higher value and obtain additional debug output.

Example

When solving the problem APL1P using strategy istrat = 2 (Monte Carlo importance sampling), corresponding to the second example of above, the file "MODEL.SCR" contains the following output:

```
 T H E   D E C I S   S Y S T E M

 Benders DEComposition and Importance Sampling

 Copyright (c) 1989 -- 1997 by Dr. Gerd Infanger

 All rights reserved.


parameters:

istrat    =           2
nsamples  =         200
nzrows    =         100
maxit     =        3000
iwrite    =           0
ibug      =           0
iscratch  =          17
iresamp   =           1
isinp     =           1
ireg      =           0
rho       =     1000.00000000000
tolben    =    0.100000000000000D-006
tolw      =    0.100000000000000D-008

input core problem

master  rows:         3 cols:           3 nonz:           5
B       rows:         6 cols:           2 nonz:           2
sub     rows:         6 cols:           9 nonz:          24

matrix B adjusted:
B       rows:         6 cols:           3 nonz:           2

START BENDERS W/ IMPORTANCE SAMPLING
SOLVE RELAXED MASTER
RELAXED MASTER DONE, OBJ:   -993500.000000000
OBJM    6500.00000000000
THETA :   -1000000.00000000
SOLVE SUB
OBJS:    19537.1509497404     VAR.:    363.791918213522
SOLVE MASTER, ITERATION:            1
OBJECTIVE:    -453499.793699476
UPPER BND.:    26037.1509497404       VAR.:    363.791918213522
OBJM    546500.206300524
SOLVE SUB
OBJS:    12680.5905253305     VAR.:    1034.00547112916
SOLVE MASTER, ITERATION:            2
OBJECTIVE:     22322.9045540029
```

```
UPPER BND.:    26037.1509497404       VAR.:    363.791918213522
OBJM    16841.2556254250
SOLVE SUB
OBJS:    10327.9172362100     VAR.:    43932.2482665064

                                              ⋮

SOLVE MASTER, ITERATION:           10
OBJECTIVE:     24659.7015921035
UPPER BND.:    24672.5441020544       VAR.:    7993.66881600826
OBJM    11766.7178571284

number of cuts:           10
number of binding cuts:    3
sigma estimated:    7993.66881600826
lbworst:    189.815930692045
RESAMPLE SUBS FOR UPPER BND
UPPER BND.:    24522.9841390933     VAR.:    7107.90709728241
 OPTIMAL SOLUTION :
 OBJ :     24522.9841390933
 X (           1) :     2063.27222540204
 X (           2) :     1369.75786839108
 X (           3) :     12982.1716298570
THETA:    12892.9837349751     ZSUB:    12845.5005665074
95% CONFIDENCE INTERVAL   :    24469.8856614114          24662.0930047817
95% CONFIDENCE INTERVAL % :  -0.216525351811624        0.567259126782395
```

The file begins with the copyright statement. It then reports the values of all DECIS parameters (default values and values set in the parameter file). Then follows a report of the number of variables, constraints, and nonzero elements, respectively for the core problem, the master, the subproblem, and for the transition matrix B after the core problem has been decomposed. Then follows output as the decomposition algorithm proceeds. On each iteration of the decomposition algorithm, it reports "OBJECTIVE", the optimal objective value of the master problem, "OBJM", the first-stage cost, "OBJS" the expected second-stage cost, "UPPER BND" the upper bound as the sum of the first-stage and the corresponding expected second-stage cost of the solution obtained from the master. If the expected second-stage cost is estimated, it reports also "VARIANCE", the variance of the estimate.

After finishing the decomposition, DECIS reports the number of cuts, the number of binding cuts in the optimal (pseudo) master, the estimate for the variance of the optimal second-stage cost, and the value of the 95% point on the worst case error distribution (to be subtracted from the optimal objective of the pseudo master problem to obtain the worst case lower bound on the true optimal objective). The latter quantities are only reported, if a strategy involving Monte Carlo sampling has been chosen. At the end of the file you find a report of the optimal solution found.

5.4. The free format solution file

The free format solution file contains the optimal solution of the problem in free format. It can be used as an input (see Section 4) when using DECIS in the evaluation mode. There is one record per variable. Each record contains the variable name and its optimal value separated by a blank character. The variable names are written in the format (A8), with quotes.

Example

The example shows a solution of problem APL1P found with strategy istrat = 2, (Monte Carlo importance sampling).

```
"X0000001"  2.06327E+03
"X0000002"  1.36976E+03
```

5.5. The optimizer output file

The optimizer output file “MODEL.MO” contains all the output that is outputted by the optimizer called as a subroutine by DECIS. When using MINOS as the optimizer, you can specify what degree of detail should be outputted by setting the appropriate “PRINT LEVEL” in the MINOS specification file.

5.6. The problem output files

The problem output files are “minfez.mps”, “sinfez.mps”, “munbnd.mps”, “sunbnd.mps”, “MODEL.P01”, “MODEL.P02”, and “MODEL.S02”. They contain a dump of the master problem or the subproblem in MPS format.

DECIS writes a dump automatically if the solution of a master or subproblem did not terminate successfully, e.g., infeasibility or unboundedness is detected. The corresponding dump file then can be used to analyze the cause of the unsuccessful termination. The name convention is the following: “minfez.mps” indicates an infeasible master problem, “sinfez.mps” an infeasible subproblem, “munbnd.mps” an unbounded master problem, and “sunbnd.mps” an unbounded subproblem. If you set to ibug $\geq$ 6, DECIS writes a dump of the master problem into the file “MODEL.P01”and a dump of the subproblem into the file “MODEL.P02”, right after the decomposition of the core file has been finished, as well as a dump of the subproblem into the file “MODEL.S02” during the first iteration of the decomposition algorithm. This also may be useful for debugging purposes.

Example “MODEL.P01”

The example contains the master problem corresponding to problem APL1P, outputted right after the core file has been decomposed. In the example, space for nzrows = 10 cuts has been reserved in the master problem. Note that the file is written in two column MPS format.

```
NAME
ROWS
 N  obj
 G  A0000001
 G  A0000002
 G  CUT01000
 G  CUT01001
 G  CUT01002
 G  CUT01003
 G  CUT01004
 G  CUT01005
 G  CUT01006
 G  CUT01007
 G  CUT01008
 G  CUT01009
COLUMNS
    X0000001  obj                  4   A0000001             1
    X0000002  obj                2.5   A0000002             1
    THETA001  obj                  1   CUT01000             0
    THETA001  CUT01001             0   CUT01002             0
    THETA001  CUT01003             0   CUT01004             0
    THETA001  CUT01005             0   CUT01006             0
    THETA001  CUT01007             0   CUT01008             0
    THETA001  CUT01009
RHS
    rhs       A0000001          1000   A0000002          1000
BOUNDS
 LO bnd       THETA001      -1000000
ENDATA
```

Example “MODEL.P02”

The example contains the subproblem corresponding to problem APL1P, outputted right after the core file has been decomposed. Note the two column MPS format.

```
NAME
ROWS
 N  obj
 L  F0000001
 L  F0000002
 G  D0000001
 G  D0000002
 G  D0000003
COLUMNS
    Y0000011  obj                4.3   F0000001             1
    Y0000011  D0000001             1
    Y0000012  obj                  2   F0000001             1
    Y0000012  D0000002             1
    Y0000013  obj                0.5   F0000001             1
    Y0000013  D0000003             1
    Y0000021  obj                8.7   F0000002             1
    Y0000021  D0000001             1
    Y0000022  obj                  4   F0000002             1
    Y0000022  D0000002             1
    Y0000023  obj                  1   F0000002             1
    Y0000023  D0000003             1
    UD000001  obj                 10   D0000001             1
    UD000002  obj                 10   D0000002             1
    UD000003  obj                 10   D0000003             1
RHS
    rhs       D0000001          1000   D0000002          1000
    rhs       D0000003          1000
ENDATA
```

5.7. The stochastic output file

The stochastic output file "MODEL.SPR" contains the information DECIS read from the stochastic file, if the debug parameter has been set to ibug ≥ 8. This is useful for debugging purposes, to see if the stochastic parameters have been specified correctly in the stochastic file "MODEL.STO".

5.8. The replications output file

The replications output file contains a collection of the different solutions when solving a problem several times in one run of DECIS. There is one record per run. Each record contains the number of iterations, the optimal objective, the 95% lower bound, the 95% upper bound, the value of "THETA" in the optimal (pseudo) master problem, and the expected second-stage cost of the optimal solution of the (pseudo) master. The replications output file is intended to be importable into a spreadsheet, and therefore has numbers only, and no text.

Example

The example contains the output of five replications of the solution of APL1P, each solved with strategy istrat = 2 (importance sampling). In the first row, corresponding to the first optimal solution, the columns represent: (10) the number of iterations, (2.45230E+04) the optimal objective of the (pseudo) master, (2.44699E+04) the 95% worst-case lower bound, (2.46621E+04) the 95% upper bound, (1.28930E+04) the value of the variable "THETA" in the optimal (pseudo) master, and (1.28455E+04) the expected second-stage cost of the optimal solution of the (pseudo) master.

```
10  2.45230E+04  2.44699E+04  2.46621E+04  1.28930E+04  1.28455E+04
11  2.47549E+04  2.44546E+04  2.48695E+04  1.39224E+04  1.40524E+04
 8  2.46249E+04  2.43856E+04  2.47383E+04  1.23946E+04  1.37764E+04
10  2.47113E+04  2.45260E+04  2.48573E+04  1.28460E+04  1.26130E+04
 8  2.46974E+04  2.44337E+04  2.48471E+04  1.33049E+04  1.24228E+04
```

6. Solution strategies

DECIS includes a variety of solution strategies. This Section gives a detailed (and mathematical) documentation of the methods used. For more information on the topic of planning under uncertainty, solving large-scale stochastic programs, consult Infanger (1994) [11].

6.1. Solving the universe problem

For solving the universe problem, DECIS uses Benders decomposition (see Benders (1962) [1] in its version for stochastic programs, also known as the L-shaped method of Van Slyke and Wets (1969) [15]. Benders decomposition splits the original problem into a master problem and into a series of independent subproblems, one for each $\omega \in \Omega$. The latter are used to generate cuts. The master problem, the subproblems and the cuts are summarized as follows.

The master problem:

$$\begin{array}{rclcccl} z_M^K & = & \min\ cx & + & \theta & & \\ & & Ax & & & = & b \\ & & -G^k x & + & \theta & \geq & g^k,\ k=1,\ldots,K \\ & & x, & & \theta & \geq & 0. \end{array}$$

The sub problems:

$$\begin{array}{rcl} z^\omega(\hat{x}^k) & = & \min\ f^\omega y^\omega \\ \pi^\omega(\hat{x}^k): & & D^\omega y^\omega \ = \ d^\omega + B^\omega \hat{x}^k \\ & & y^\omega \ \geq \ 0,\ \omega \in \Omega = \{1,2,\ldots,W\}, \end{array}$$

where $\pi^\omega(\hat{x}^k)$ is the optimal dual solution of subproblem ω, given $\hat{x}^k$, the first stage decision passed on iteration k to the subproblems.

The expected second-stage costs and parameters of the cuts:

$$\begin{array}{rcl} z(\hat{x}^k) & = & E\ z^\omega(\hat{x}^k) = \sum_{\omega\in\Omega} p^\omega z^\omega(\hat{x}^k) \\ G^{k+1} & = & E\ \pi^\omega(\hat{x}^k)B^\omega = \sum_{\omega\in\Omega} p^\omega \pi^\omega(\hat{x}^k)B^\omega \\ g^{k+1} & = & z(\hat{x}^k) - G^{k+1}\hat{x}^k \end{array}$$

The bounds

$$\begin{array}{rcl} z_{LB}^k & = & z_M^k = c\hat{x}^k + \hat{\theta}^k \\ z_{UB}^k & = & \min\{z_{UB}^{k-1}, c\hat{x}^k + z(\hat{x}^k)\},\quad z_{UB}^0 = \infty. \end{array}$$

The algorithm is started with $\hat{x}^0$ obtained from solving the relaxed master problem $\min cx$ subject to $Ax = b, x \geq 0$. By solving the master problem on iteration k a trial solution $\hat{x}^k$, $\hat{\theta}^k$ is obtained. Given $\hat{x}^k$ we solve W subproblems $\omega \in \Omega$ to compute the optimal expected second-stage costs $z(\hat{x}^k)$ and the coefficients G^{k+1} and the right-hand side g^{k+1} of a cut. If a subproblem ω turns out to be infeasible, with $\pi^0(\hat{x}^k)$ the representative of the dual extreme ray, and $z^0(\hat{x}^k)$ the sum of infeasibilities, we compute a feasibility cut as $-G^{k+1}x \geq g^{k+1}$, where $G^{k+1} = \pi^0(\hat{x}^k)B^\omega$ and $g^{k+1} = z^0(\hat{x}^k) - G^{k+1}\hat{x}^k$, and set the corresponding upper bound as $z_{UB}^k = \infty$. Cuts are sequentially added to the master problem. On each iteration k of the Benders decomposition algorithm a trial solution $\hat{x}^k$, a lower bound z_{LB}^k, and an upper bound z_{UB}^k are obtained. The algorithm proceeds until on iteration $k = K$

$$\frac{|z_{UB}^k - z_{LB}^k|}{|z_{LB}^k|} \leq \text{TOL},$$

the optimality criterion is met. A TOL optimal solution of the problem, denoted as $\hat{x}^\ell$, is obtained as the solution corresponding to the best upper bound,

$$\hat{x}^\ell = \arg\min\ \{c\hat{x}^k + z(\hat{x}^k),\quad k = 1,\ldots,K\}.$$

6.2. Solving the expected value problem

The expected value problem results from replacing the second-stage uncertain parameters by their expectation, $\bar{f} = E\ f^\omega$, $\bar{B} = E\ B^\omega$, $\bar{D} = E\ D^\omega$, $\bar{d} = E\ d^\omega$:

$$\begin{array}{rcrcrcl} \min\ z & = & cx & + & \bar{f}y & & \\ s/t & & Ax & & & = & b \\ & & -\bar{B}x & + & \bar{D}y & = & \bar{d} \\ & & x, & & y & \geq & 0. \end{array}$$

Solving the expected value problem may be useful by itself, its optimal solution $\hat{x}^\ell$ may also provide a good starting point for solving the universe program. It can be solved directly as a linear problem. DECIS, however, solves it using Benders decomposition. In addition to $\hat{x}^\ell$ being a good starting solution, the expected value cuts may be useful to guide the algorithm when solving the stochastic problem. For the expected value problem, the master problem, the subproblems, the cuts, and the bounds are summarized as follows.

The expected value master problem:

$$\begin{array}{rcrcrcl} z_M^K & = & \min\ cx & + & \theta & & \\ & & Ax & & & = & b \\ & & -G^k x & + & \theta & \geq & g^k,\ k = 1, \ldots, K \\ & & x, & & \theta & \geq & 0. \end{array}$$

The expected value sub problem:

$$\begin{array}{rcrcl} z(\hat{x}^k) & = & \min\ \bar{f}y & & \\ \pi(\hat{x}^k): & & \bar{D}y & = & \bar{d} + \bar{B}\hat{x}^k \\ & & y & \geq & 0, \end{array}$$

where $\pi(\hat{x}^k)$ is the optimal dual solution of the expected value subproblem, given $\hat{x}^k$.

The second-stage cost and parameters of the cuts:

$$\begin{array}{rcl} z(\hat{x}^k) & = & z(\hat{x}^k) \\ G^{k+1} & = & \pi(\hat{x}^k)\bar{B} \\ g^{k+1} & = & z(\hat{x}^k) - G^{k+1}\hat{x}^k \end{array}$$

The bounds

$$\begin{array}{rcl} z_{LB}^k & = & z_M^k = c\hat{x}^k + \hat{\theta}^k \\ z_{UB}^k & = & \min\{z_{UB}^{k-1}, c\hat{x}^k + z(\hat{x}^k)\},\quad z_{UB}^0 = \infty. \end{array}$$

The algorithm is started with $\hat{x}^0$ obtained from solving the relaxed master problem $\min cx$ subject to $Ax = b, x \geq 0$. By solving the master problem on iteration k, a trial solution $\hat{x}^k$, $\hat{\theta}^k$ is obtained. Given $\hat{x}^k$ we solve the expected value subproblem to compute the optimal second-stage costs $z(\hat{x}^k)$ and the coefficients G^{k+1} and the right-hand side g^{k+1} of a cut. If a subproblem turns out to be infeasible, with $\pi^0(\hat{x}^k)$ the representative of the dual extreme ray, and $z^0(\hat{x}^k)$ the sum of infeasibilities, we compute a feasibility cut as $-G^{k+1}x \geq g^{k+1}$, where $G^{k+1} = \pi^0(\hat{x}^k)\bar{B}$ and $g^{k+1} = z^0(\hat{x}^k) - G^{k+1}\hat{x}^k$, and set the corresponding upper bound as $z^k_{UB} = \infty$. On each iteration k of the Benders decomposition algorithm a trial solution $\hat{x}^k$, a lower bound z^k_{LB}, and an upper bound z^k_{UB} are obtained. The algorithm proceeds until on iteration $k = K$

$$\frac{|z^k_{UB} - z^k_{LB}|}{|z^k_{LB}|} \leq \text{TOL},$$

the optimality criterion is met. A TOL optimal solution, denoted as $\hat{x}^\ell$, of the problem is obtained as the solution corresponding to the best upper bound,

$$\hat{x}^\ell = \arg\min \{c\hat{x}^k + z(\hat{x}^k), \quad k = 1, \ldots, K\}.$$

6.3. Using Monte Carlo sampling for solving the problem

Monte Carlo sampling works very well for estimating multiple integrals or multiple sums for higher h-dimensional spaces. According to Dantzig and Glynn (1990) [5] and Infanger (1992) [10], instead of computing expectations exactly by enumerating all possible outcomes $\omega \in \Omega$ (which as already noted above for problems with many random parameters is impossible due to the many combinations of possible outcomes), we use Monte Carlo sampling for estimating the expectations $z(\hat{x}^k) = E\ z^\omega(\hat{x}^k)$, and $G^{k+1} = E\ \pi^\omega(\hat{x}^k)$ on each iteration k of the Benders decomposition algorithm.

6.3.1. Crude Monte Carlo sampling

Crude Monte Carlo refers to employing Monte Carlo sampling directly without any variance reduction techniques. On each iteration k of the Benders decomposition algorithm, we draw an independent sample S^k of size $|S^k| = N$ from the distribution of possible outcomes $\omega \in \Omega$ and estimate the expectations $z(\hat{x}^k)$, and G^{k+1} using the unbiased estimators

$$\tilde{z}(\hat{x}^k) = \frac{1}{N} \sum_{\omega \in S^k} z^\omega(\hat{x}^k),$$

and

$$\tilde{G}^{k+1} = \frac{1}{N} \sum_{\omega \in S^k} \pi^\omega(\hat{x}^k) B^\omega$$

The estimator $\tilde{z}(\hat{x}^k)$ has a true variance

$$\sigma^2_{\tilde{z}}(\hat{x}^k) = \frac{1}{N}\sigma^2(\hat{x}^k),$$

where $\sigma^2(\hat{x}^k)$ is the (population) variance of the second stage costs, given $\hat{x}^k$, e.g.,

$$\sigma^2(\hat{x}^k) = \sum_{\omega \in \Omega} (z^\omega(\hat{x}^k) - z(\hat{x}^k))^2 p^\omega.$$

We estimate the variance $\sigma^2(\hat{x}^k)$ using sample S^k as

$$\tilde{\sigma}^2(\hat{x}^k) = \frac{1}{N-1} \sum_{\omega \in S^k} (z^\omega(\hat{x}^k) - z(\hat{x}^k))^2$$

and obtain

$$\tilde{\sigma}^2_{\tilde{z}}(\hat{x}^k) = \frac{1}{N}\tilde{\sigma}^2(\hat{x}^k).$$

6.3.2. Importance sampling

Importance sampling is a powerful variance reduction technique. Compared to crude Monte Carlo you can expect, using the same sample size, a significantly better estimate.

We recall that $z^\omega(\hat{x}^k) = z(v^\omega, \hat{x}^k)$, since the parameters of the second-stage subproblem are all functions of the independent random vector V: $f^\omega = f(v^\omega)$, $B^\omega = B(v^\omega)$, $D^\omega = D(v^\omega)$, $d^\omega = d(v^\omega)$. Suppose there is a function $\Gamma(v^\omega, \hat{x}^k)$ that closely approximates $z(v^\omega, \hat{x}^k)$ and whose expectation $\bar{\Gamma}(\hat{x}^k)$ is known or can be computed easily.

We rewrite $z(\hat{x}^k) = \sum_{\omega\in\Omega} z(v^\omega, \hat{x}^k)p(v^\omega)$ as

$$z(\hat{x}^k) = \sum_{\omega\in\Omega} z(v^\omega, \hat{x}^k)\frac{\bar{\Gamma}(\hat{x}^k)}{\Gamma(v^\omega, \hat{x}^k)}\frac{\Gamma(v^\omega, \hat{x}^k)}{\bar{\Gamma}(\hat{x}^k)}p(v^\omega)$$

and interpret it as

$$z(\hat{x}^k) = \bar{\Gamma}(\hat{x}^k)\ E\ \frac{z(v^\omega, \hat{x}^k)}{\Gamma(v^\omega, \hat{x}^k)}$$

distributed according to the probability mass function

$$q(v^\omega, \hat{x}^k) = \frac{\Gamma(v^\omega, \hat{x}^k)}{\bar{\Gamma}(\hat{x}^k)}p(v^\omega),$$

which depends on the first-stage decision $\hat{x}^k$ on iteration k.

Since $\bar{\Gamma}(\hat{x}^k) = \sum_{\omega\in\Omega} \Gamma(v^\omega, \hat{x}^k)p(v^\omega)$,

$$\sum_{\omega\in\Omega} q(v^\omega, \hat{x}^k) = \frac{\sum \Gamma(v^\omega, \hat{x}^k)p(v^\omega)}{\bar{\Gamma}(\hat{x}^k)} = 1.$$

Since $p(v^\omega) \geq 0$, if all $\Gamma(v^\omega, \hat{x}^k) \geq 0$ also $\bar{\Gamma}(\hat{x}^k) \geq 0$ and it follows that

$$q(v^\omega, \hat{x}^k) = \frac{\Gamma(v^\omega, \hat{x}^k)}{\bar{\Gamma}(\hat{x}^k)}p(v^\omega) \geq 0.$$

Also, if all $\Gamma(v^\omega, \hat{x}^k) \leq 0$ also $\bar{\Gamma}(\hat{x}^k) \leq 0$ and it follows again that

$$q(v^\omega, \hat{x}^k) = \frac{\Gamma(v^\omega, \hat{x}^k)}{\bar{\Gamma}(\hat{x}^k)}p(v^\omega) \geq 0.$$

Either $\Gamma(v^\omega, \hat{x}^k) \geq 0$ for all $\omega \in \Omega$ or $\Gamma(v^\omega, \hat{x}^k) \leq 0$ for all $\omega \in \Omega$ can be ensured by adding an appropriate constant to the approximation function.

We obtain a new estimator of $z(\hat{x}^k)$ as

$$\tilde{z}(\hat{x}^k) = \frac{1}{N}\bar{\Gamma}(\hat{x}^k)\sum_{\omega\in S^k}\frac{z(v^\omega, \hat{x}^k)}{\Gamma(v^\omega, \hat{x}^k)}$$

by sampling a sample S^k of size $|S^k| = N$ from the distribution $q(v^\omega, \hat{x}^k)$. The variance of $\tilde{z}$ is given by

$$\operatorname{var}(\tilde{z}) = \frac{1}{N} \sum_{\omega \in \Omega} \left(\bar{\Gamma}(\hat{x}^k) \frac{z(v^\omega, \hat{x}^k)}{\Gamma(v^\omega, \hat{x}^k)} - z(\hat{x}^k) \right)^2 q(v^\omega).$$

One can see easily that if $\Gamma(v^\omega, \hat{x}^k) = z(v^\omega, \hat{x}^k)$ then $\operatorname{var}(\bar{z}) = 0$, and

$$q^*(v^\omega, \hat{x}^k) = \frac{z(v^\omega, \hat{x}^k)}{z(\hat{x}^k)} p(v^\omega),$$

where q^* reflects the "best" importance distribution. This, of course, is only theoretical, since for computing the "best" importance distribution we need to know $z(\hat{x}^k)$, which we eventually want to compute. Note that the optimal importance density is proportional to the product $z(v^\omega, \hat{x}^k)p(v^\omega)$. If we use a $\Gamma(v^\omega, \hat{x}^k) \approx z(v^\omega, \hat{x}^k)$, we obtain a "good" importance distribution, which is approximately proportional to the product $z(v^\omega, \hat{x}^k)p(v^\omega)$.

DECIS uses additive and multiplicative (in the components of the stochastic vector V) approximation functions for obtaining variance reductions through importance sampling. Since $p(v^\omega) = \prod_{i=1}^h p_i(v^{\omega_i})$, if $\Gamma(v^\omega, \hat{x}^k) = \sum_{i=1}^h \Gamma_i(v^{\omega_i}, \hat{x}^k)$ then

$$\bar{\Gamma}(\hat{x}^k) = \sum_{i=1}^h E\ \Gamma_i(v^{\omega_i}, \hat{x}^k) = \sum_{i=1}^h \sum_{\omega_i \in \Omega_i} \Gamma_i(v^{\omega_i}, \hat{x}^k) p_i(v^{\omega_i})$$

and if $\Gamma(v^\omega, \hat{x}^k) = \prod_{i=1}^h \Gamma_i(v^{\omega_i}, \hat{x}^k)$ then

$$\bar{\Gamma}(\hat{x}^k) = \prod_{i=1}^h E\ \Gamma_i(v^{\omega_i}, \hat{x}^k) = \prod_{i=1}^h \sum_{\omega_i \in \Omega_i} \Gamma_i(v^{\omega_i}, \hat{x}^k) p_i(v^{\omega_i}).$$

Instead of computing one h-dimensional integral or sum, both in the additive and multiplicative case the expected value calculation of the approximation function reduces to computing h 1-dimensional integrals or sums.

Additive approximation function

As additive approximation function DECIS uses a *additive marginal cost* approximation. We introduce $z(\tau, \hat{x}^k)$, the costs of a base case, and compute the approximate cost as the cost of the base case plus the sum of the marginal cost in each dimension of V:

$$\Gamma(v^\omega, \hat{x}^k) = z(\tau, \hat{x}^k) + \sum_{i=1}^h \Gamma_i(v_i^{\omega_i}, \hat{x}^k),$$

$$\Gamma_i(v_i^{\omega_i}, \hat{x}^k) = z(\tau_1, \ldots, \tau_{i-1}, v_i^{\omega_i}, \tau_{i+1}, \ldots, \tau_h, \hat{x}^k) - z(\tau, \hat{x}^k).$$

Using this model, we explore the cost function at the margins, e.g., we vary the random elements V_i to compute the costs for all outcomes $v_i^{\omega_i}$ while we fix the other random elements at the level of the base case τ. The base case can be any arbitrary chosen point of the set of k_i discrete values of V_i, $i = 1, \ldots, h$. If possible, we choose τ_i as that outcome of V_i which leads to the lowest costs, ceteris paribus.

Using the additive marginal cost model, we can express the expected value of the second stage costs in the following form:

$$z(\hat{x}^k) = z(\tau, \hat{x}^k) + \sum_{i=1}^{h} \bar{\Gamma}_i(\hat{x}^k) \sum_{\omega\in\Omega} \left[\frac{z(v^\omega, \hat{x}^k) - z(\tau, \hat{x}^k)}{\sum_{i=1}^{h} \Gamma_i(v_i^{\omega_i}, \hat{x}^k)} \right] \frac{\Gamma_i(v_i^{\omega_i}, \hat{x}^k)}{\bar{\Gamma}_i(\hat{x}^k)} \prod_{j=1}^{h} p_j(v_j^{\omega_i}),$$

where we assume that $\sum_{i=1}^{h} \Gamma_i(v_i^\omega, \hat{x}^k) > 0$, so that at least one $\Gamma_i(v_i^\omega, \hat{x}) > 0$.

Note that this formulation consists of a constant term and a sum of h expectations. Given a fixed sample size N we partition N into h sub-samples, with sample sizes N_i, $i = 1, \ldots, h$ such that $\Sigma N_i = N$ and $N_i \geq 1$, $i = 1, \ldots, N$, where N_i is approximately proportional to $\bar{\Gamma}_i(\hat{x}^k)$. The h expectations are separately approximated by sampling from marginal densities. The i-th expectation corresponds to the i-th component of V. In generating sample points for the i-th expectation, we use the importance density $(p_i\Gamma_i/\bar{\Gamma}_i)$ for sampling the i-th component of V and the original marginal densities for any other components.

Denoting by

$$\tilde{\mu}_i(\hat{x}) = \frac{1}{N_i} \sum_{\omega=1}^{N_i} \frac{z(v^\omega, \hat{x}^k) - z(\tau, \hat{x}^k)}{\sum_{i=1}^{h} \Gamma_i(v_i^{\omega_i}, \hat{x}^k)}$$

the estimate of the i-th sum, we obtain

$$\tilde{z}(\hat{x}^k) = z(\tau, \hat{x}^k) + \sum_{i=1}^{h} \bar{\Gamma}_i(\hat{x}^k)\tilde{\mu}_i(\hat{x}^k),$$

the estimated expected value of the second-stage costs. Let $\tilde{\sigma}_i^2(\hat{x}^k)$ be the sample variance of the i-th expectation, where we set $\tilde{\sigma}_i^2(\hat{x}^k) = 0$ if $N_i = 1$. The estimated variance of the mean, $\tilde{\sigma}_{\tilde{z}}^2(\hat{x}^k)$, is then given by

$$\tilde{\sigma}_{\tilde{z}}^2(\hat{x}^k) = \sum_{i=1}^{h} \frac{\bar{\Gamma}_i^2(\hat{x}^k)\tilde{\sigma}_i^2(\hat{x}^k)}{N_i}.$$

To derive a cut we use an analogous framework:

$$G(\hat{x}^k) = \pi(\tau, \hat{x}^k)B(\tau) +$$

$$\sum_{i=1}^{h} \bar{\Gamma}_i(\hat{x}^k) \sum_{\omega\in\Omega} \left[\frac{\pi(v^\omega, \hat{x}^k)B(v^\omega) - \pi(\tau, \hat{x}^k)B(\tau)}{\sum_{i=1}^{h} \Gamma_i(v^{\omega_i}, \hat{x}^k)} \right] \frac{\Gamma_i(v^{\omega_i}, \hat{x}^k)}{\bar{\Gamma}_i(\hat{x}^k)} \prod_{j=1}^{h} p_j(v^{\omega_j}).$$

We estimate the coefficients of a cut by sampling using the same scheme and same sample points at hand from the computation of $\tilde{z}(\hat{x}^k)$ to obtain $\tilde{G}(\hat{x}^k)$, and compute

$$\tilde{g}(\hat{x}^k) = \tilde{z}(\hat{x}^k) - \tilde{G}(\hat{x}^k)\hat{x}^k.$$

Multiplicative approximation function

For the multiplicative approximation function we introduce the cost of a base case $z(\tau, \hat{x}^k)$, and compute the approximate costs as the cost of the base case times the relative marginal cost in each dimension of V:

$$\Gamma(v^\omega, \hat{x}^k) = z(\tau, \hat{x}^k) \prod_{i=1}^{h} \Gamma_i(v_i^{\omega_i}, \hat{x}^k),$$

$$\Gamma_i(v_i^{\omega_i}, \hat{x}^k) = \frac{z(\tau_1, \ldots, \tau_{i-1}, v_i^{\omega_i}, \tau_{i+1}, \ldots, \tau_h, \hat{x}^k)}{z(\tau, \hat{x}^k)}.$$

We call this the *multiplicative marginal cost model.* Using this model, we explore the cost function at the margins, e.g. we vary the random elements V_i to compute the costs for all outcomes $v_i^{\omega_i}$ while we fix the other random elements at the level of the base case τ and compute the relative marginal cost with regard to the cost of the base case. The base case can be any arbitrary chosen point of the set of W_i discrete values of V_i, $i = 1, \ldots, h$, since (according to the assumptions) $z(v^\omega, \hat{x}^k) \geq 0$, and therefore also $\Gamma_i(v_i, \hat{x}^k) \geq 0$.

Using the multiplicative marginal cost model, we can express the expected value of the second stage costs in the following form:

$$z(\hat{x}^k) = \bar{\Gamma}(\hat{x}^k) \sum_{\omega \in \Omega} \left[\frac{z(v^\omega, \hat{x}^k)}{\Gamma(v^\omega, \hat{x}^k)} \right] \prod_{i=1}^{h} \frac{\Gamma_i(v_i^{\omega_i}, \hat{x}^k)}{\bar{\Gamma}_i(\hat{x}^k)} p_i(v_i^{\omega_i}).$$

Note that using the multiplicative marginal approximation we need to estimate only one expectation. We estimate the mean by sampling using the marginal densities $p_i \Gamma_i / \bar{\Gamma}_i$ in each dimension i of the multidimensional random vector V:

$$\tilde{z}(\hat{x}^k) = \bar{\Gamma}(\hat{x}^k) \frac{1}{N} \sum_{\omega=1}^{N} \frac{z(v^\omega, \hat{x}^k)}{\Gamma(v^\omega, \hat{x}^k)},$$

and estimate the variance of the mean, $\sigma_{\tilde{z}}^2(\hat{x}^k)$, as

$$\tilde{\sigma}_{\tilde{z}}^2(\hat{x}^k) = \frac{1}{N-1} \sum_{\omega=1}^{N} \left[\bar{\Gamma}(\hat{x}^k) \frac{z(v^\omega, \hat{x}^k)}{\Gamma(v^\omega, \hat{x}^k)} - \tilde{z}(\hat{x}^k) \right]^2.$$

To derive a cut we use an analogous framework:

$$G(\hat{x}^k) = \bar{\Gamma}(\hat{x}^k) \sum_{\omega \in \Omega} \left[\frac{\pi(v^\omega, \hat{x}^k) B(v^\omega)}{\Gamma(v^\omega, \hat{x}^k)} \right] \prod_{i=1}^{h} \frac{\Gamma_i(v_i^{\omega_i}, \hat{x}^k)}{\bar{\Gamma}_i(\hat{x}^k)} p_i(v_i^{\omega_i}).$$

We estimate the coefficients of a cut by sampling using the same scheme and sample points at hand from the computation of $\tilde{z}(\hat{x}^k)$ to obtain

$$\tilde{G}(\hat{x}^k) = \bar{\Gamma}(\hat{x}^k) \frac{1}{N} \sum_{\omega=1}^{N} \frac{\pi(v^\omega, \hat{x}^k) B(v^\omega)}{\Gamma(v^\omega, \hat{x}^k)},$$

and we compute

$$\tilde{g}(\hat{x}^k) = \tilde{z}(\hat{x}^k) - \tilde{G}(\hat{x}^k) \hat{x}^k.$$

6.3.3. Control variates

Control Variates is also a powerful variance reduction technique, which (using the same size of sample) may lead to significantly better estimates. We recall that $z^\omega(\hat{x}^k) = z(v^\omega, \hat{x}^k)$. Suppose there is a function $\Gamma(v^\omega, \hat{x}^k)$ that closely approximates $z(v^\omega, \hat{x}^k)$, (i.e., is positively correlated) and whose expectation $\bar{\Gamma}(\hat{x}^k)$ is known or can be computed easily.

We rewrite $z(\hat{x}^k) = \sum_{\omega\in\Omega} z(v^\omega, \hat{x}^k)p(v^\omega)$ as

$$z(\hat{x}^k) = \sum_{\omega\in\Omega} \left[z(v^\omega, \hat{x}^k) - \alpha\Gamma(v^\omega, \hat{x}^k)\right] p(v^\omega) + \alpha\bar{\Gamma}(\hat{x}^k)$$

and interpret it as

$$z(\hat{x}^k) = E\left[z(v^\omega, \hat{x}^k) - \alpha\Gamma(v^\omega, \hat{x}^k)\right] + \alpha\bar{\Gamma}(\hat{x}^k).$$

We obtain a new estimator of $z(\hat{x}^k)$,

$$\tilde{z}(\hat{x}^k) = \frac{1}{N}\sum_{\omega\in S^k} \left[z(v^\omega, \hat{x}^k) - \alpha\Gamma(v^\omega, \hat{x}^k)\right] + \alpha\bar{\Gamma}(\hat{x}^k)$$

by sampling a sample S^k of size $|S^k| = N$ from the distribution $p(v^\omega)$. The variance of $\tilde{z}$ is given by

$$\mathrm{var}(\tilde{z}) = \frac{1}{N}\mathrm{var}(z(v^\omega, \hat{x}^k) - \alpha\Gamma(v^\omega, \hat{x}^k))$$

$$\mathrm{var}(\tilde{z}) = \frac{1}{N}\left[\mathrm{var}(z(v^\omega, \hat{x}^k)) + \alpha^2\mathrm{var}(\Gamma(v^\omega, \hat{x}^k)) - 2\alpha\mathrm{cov}(z(v^\omega, \hat{x}^k), \Gamma(v^\omega, \hat{x}^k))\right]$$

One can see easily that the larger the covariance between $z(v^\omega, \hat{x}^k)$ and $\Gamma(v^\omega, \hat{x}^k)$ is, the smaller is the variance of the estimate of $z(\hat{x}^k)$. Minimizing the expression above with respect to α, we obtain the optimal value of α,

$$\alpha^* = \frac{\mathrm{cov}(z(v^\omega, \hat{x}^k), \Gamma(v^\omega, \hat{x}^k))}{\mathrm{var}(\Gamma(v^\omega, \hat{x}^k))}$$

the value of alpha that leads to the smallest variance of the estimate. When carrying out Monte Carlo sampling with control variates, we estimate both $\mathrm{cov}(z(v^\omega, \hat{x}^k), \Gamma(v^\omega, \hat{x}^k))$, and $\mathrm{var}(\Gamma(v^\omega, \hat{x}^k))$ using the sample S^k used for estimating the expected value $z(\hat{x}^k)$.

To estimate the coefficients of a cut we use the analogous framework:

$$G(\hat{x}^k) = \sum_{\omega\in\Omega} \left[\pi(v^\omega, \hat{x}^k)B(v^\omega) - \alpha\Gamma(v^\omega, \hat{x}^k)\right] p(v^\omega) + \bar{\Gamma}(\hat{x}^k).$$

We compute

$$\tilde{G}(\hat{x}^k) = \frac{1}{N}\sum_{\omega\in S^k} \left[\pi(v^\omega, \hat{x}^k)B(v^\omega) - \alpha\Gamma(v^\omega, \hat{x}^k)\right] + \alpha\bar{\Gamma}(\hat{x}^k)$$

using the sample points at hand from the computation of $\tilde{z}(\hat{x}^k)$, and compute

$$\tilde{g}(\hat{x}^k) = \tilde{z}(\hat{x}^k) - \tilde{G}(\hat{x}^k)\hat{x}^k.$$

DECIS uses both additive and multiplicative approximation functions as control variates. Both in the additive and multiplicative case the expected value calculation of the approximation function requires computing h 1-dimensional integrals or sums.

Additive approximation function

Using the *additive marginal cost* approximation

$$\Gamma(v^{\omega}, \hat{x}^k) = z(\tau, \hat{x}^k) + \sum_{i=1}^{h} \Gamma_i(v_i^{\omega_i}, \hat{x}^k),$$

where

$$\Gamma_i(v_i^{\omega_i}, \hat{x}^k) = z(\tau_1, \ldots, \tau_{i-1}, v_i^{\omega_i}, \tau_{i+1}, \ldots, \tau_h, \hat{x}^k) - z(\tau, \hat{x}^k),$$

we compute

$$\bar{\Gamma}(\hat{x}^k) = z(\tau, \hat{x}^k) + \sum_{i=1}^{h} \bar{\Gamma}_i(\hat{x}^k),$$

and

$$\bar{\Gamma}_i(\hat{x}^k) = \sum_{\omega_i \in \Omega_i} z(\tau_1, \ldots, \tau_{i-1}, v_i^{\omega_i}, \tau_{i+1}, \ldots, \tau_h, \hat{x}^k) p_i^{\omega_i} - z(\tau, \hat{x}^k).$$

Multiplicative approximation function

Using the *multiplicative marginal cost* approximation

$$\Gamma(v^{\omega}, \hat{x}^k) = z(\tau, \hat{x}^k) \prod_{i=1}^{h} \Gamma_i(v_i^{\omega_i}, \hat{x}^k),$$

where

$$\Gamma_i(v_i^{\omega_i}, \hat{x}^k) = \frac{z(\tau_1, \ldots, \tau_{i-1}, v_i^{\omega_i}, \tau_{i+1}, \ldots, \tau_h, \hat{x}^k)}{z(\tau, \hat{x}^k)},$$

we compute

$$\bar{\Gamma}(\hat{x}^k) = z(\tau, \hat{x}^k) \prod_{i=1}^{h} \bar{\Gamma}_i(\hat{x}^k),$$

and

$$\bar{\Gamma}_i(\hat{x}^k) = \frac{1}{z(\tau, \hat{x}^k)} \sum_{\omega_i \in \Omega_i} z(\tau_1, \ldots, \tau_{i-1}, v_i^{\omega_i}, \tau_{i+1}, \ldots, \tau_h, \hat{x}^k) p_i^{\omega_i}.$$

6.3.4. Summary Monte Carlo sampling

Using Monte Carlo sampling (crude Monte Carlo, importance sampling, or control variates) we estimate on each iteration k of the Benders decomposition algorithm the expected second stage costs $z(\hat{x}^k)$, and the coefficients $G(\hat{x}^k)$, and the right-hand sides $g(\hat{x}^k)$ using an independent sample S^k and compute the estimates $\tilde{z}(\hat{x}^k)$, $\tilde{G}^{k+1} = \tilde{G}(\hat{x}^k)$, and $\tilde{g}^{k+1} = \tilde{g}(\hat{x}^k)$. The master problem (denoted as pseudo master problem), the subproblems and the cuts based on estimates are summarized as follows:

The pseudo master problem:

$$\begin{array}{rrcrcl} \min\ \tilde{v}^K & = & cx & + & \theta & \\ \tilde{\rho}: & & Ax & & & = b \\ \tilde{\lambda}^k: & & -\tilde{G}^k x & + & \theta & \geq \tilde{g}^k,\ k=1,\dots,K \\ & & x, & & \theta & \geq 0, \end{array}$$

where $\tilde{\rho}$ and $\tilde{\lambda}^k$ are optimal dual multipliers.

The sub problems:

$$\begin{array}{rcrcl} z^\omega(\hat{x}^k) & = & \min\ f^\omega y^\omega & & \\ \pi^\omega(\hat{x}^k): & & D^\omega y^\omega & = & d^\omega + B^\omega \hat{x}^k \\ & & y^\omega & \geq & 0,\ \omega \in \Omega, \end{array}$$

where $\pi^\omega(\hat{x}^k)$ is the optimal dual solution of subproblem ω, given $\hat{x}^k$, the first stage decision passed on iteration k to the subproblems.

The expected second-stage costs and parameters of the cuts:

$$\begin{array}{rcl} \tilde{z}(\hat{x}^k) & = & \sum_{\omega \in S^k} z^\omega(\hat{x}^k) \\ \tilde{G}^{k+1} & = & \sum_{\omega \in S^k} \pi^\omega(\hat{x}^k) B^\omega \\ \tilde{g}^{k+1} & = & \tilde{z}(\hat{x}^k) - \tilde{G}\hat{x}^k \end{array}$$

The algorithm is started with $\hat{x}^0$ obtained from solving the relaxed (pseudo) master problem $\min cx$ subject to $Ax = b, x \geq 0$, or obtained as the optimal solution of the expected value problem. By solving the master problem on iteration k a trial solution $\hat{x}^k$, $\hat{\theta}^k$ is obtained. Given $\hat{x}^k$ we solve N subproblems $\omega \in S^k$ to compute the estimates of the optimal expected second-stage costs $\tilde{z}(\hat{x}^k)$ and the coefficients $\tilde{G}^{k+1}$ and the right-hand side $\tilde{g}^{k+1}$ of a cut. The algorithm proceeds until $\min_k\{c\hat{x}^k + \tilde{z}(\hat{x}^k),\ k = 1,\dots,K\}$ and $\tilde{v}^K$ are sufficiently close, which is tested using a Student-t test. We declare the best solution found

$$\hat{x}^\ell \in \arg\min\ \{c\hat{x}^k + \tilde{z}(\hat{x}^k), \quad k = 1,\dots,K\}$$

as the approximately optimal solution of the problem and compute a $\alpha = 0.95$ confidence interval based on probabilistic upper and lower bounds.

6.3.5. The probabilistic lower bound

An extensive discussion of the theory for probabilistic lower bounds is outlined in Dantzig and Infanger (1995) [6]. Basis for the derivation of the lower bounds is the true system of inequalities below:

$$\begin{array}{rcrcrcl} z^* & = & cx^* & + & \theta^* & & \\ & & Ax^* & & & = & b \\ & & -\tilde{G}^k x^* & + & \theta^* & \geq & \tilde{g}^k - \tilde{\epsilon}^k_*,\quad k=1,\dots,K \\ & & x^* & & & \geq & 0, \end{array}$$

where

$$\tilde{\epsilon}_*^k = \sum_{\omega \in S^k} z^\omega(x^*) - z(x^*),$$

is the difference between the estimated optimal expected second stage costs based on sample S^k and its true expected value, and x^* denotes the true optimal solution of the problem.

The optimal solution of the pseudo master problem $\tilde{v}^* = \min \tilde{v}$, and the optimal dual multipliers $\tilde{\rho}$, $\tilde{\lambda}^1, \ldots, \tilde{\lambda}^K$ satisfy

$$\tilde{v}^* = \tilde{\rho} b + \sum_{k=1}^K \tilde{\lambda}^k \tilde{g}^k$$

$$\tilde{\rho} A - \sum_{k=1}^K \tilde{\lambda}^k \tilde{G}^k + \gamma = c, \quad \gamma \geq 0$$

$$\sum_{k=1}^K \tilde{\lambda}^k = 1$$

$$\tilde{\lambda}^k \geq 0, \quad k = 1, \ldots, K.$$

Applying these multipliers to the corresponding relations in the system of inequalities above and subtracting from the first relation, we obtain

$$0 \leq \gamma x^* \leq z^* - (\tilde{\rho} b + \sum_{k=1}^K \tilde{\lambda}^k \tilde{g}^k) + \sum_{k=1}^K \tilde{\lambda}^k \tilde{\epsilon}_*^k.$$

Dropping $\gamma x^* \geq 0$, substituting $\tilde{v}^*$, and rearranging terms, we obtain a lower bound for the optimal objective z^*:

$$\tilde{v}^* - \Delta \leq z^*, \quad \Delta = \sum_{k=1}^K \tilde{\lambda}^k \tilde{\epsilon}_*^k$$

where $\tilde{\lambda}^k$ are optimal dual multipliers of the pseudo master, $\tilde{\lambda}^k \geq 0$, $\sum_{k=1}^K \tilde{\lambda}^k = 1$.

The worst-case lower bound

Since $\tilde{\lambda}^k \geq 0$ and $\sum_{k=1}^K \tilde{\lambda}^k = 1$, the worst-case upper bound for Δ is Δ_W,

$$\Delta_W = \max_k \tilde{\epsilon}_*^k \geq \sum_{k=1}^K \tilde{\lambda}^k \tilde{\epsilon}_*^k = \Delta.$$

Each error term $\tilde{\epsilon}_*^k$ is the difference of the average of N independently drawn observations with replacements *from the same distribution* of $z^\omega(x^*)$ minus its true mean $z(x^*)$. By the central limit theorem, for sufficiently large sample sizes, $\tilde{\epsilon}_*^k$ are approximately normally distributed:

$$\tilde{\epsilon}_*^k \sim N(0, \frac{\sigma_*}{\sqrt{N}}) = \frac{\sigma_*}{\sqrt{N}} N(0,1),$$

where

$$\sigma_*^2 = \sum_{\omega \in \Omega} (z^\omega(x^*) - z(x^*))^2 p^\omega$$

is the (population) variance of the optimum second stage cost. We estimate σ_*^2 by

$$\tilde{\sigma}^2(\hat{x}^\ell) = \frac{1}{N-1} \sum_{\omega \in S^\ell} (z^\omega(\hat{x}^\ell) - z(\hat{x}^\ell))^2,$$

the estimated population variance of the second stage cost of solution $\hat{x}^\ell$, the best solution found.

We determine the point Δ_W^α on the distribution of Δ_W, the distribution of the maximum of K independent normal $(\sigma_*/\sqrt{N})N(0,1)$ variates, such that

$$\text{prob}(\Delta_W \leq \Delta_W^\alpha) \geq \alpha,$$

where

$$\Delta_W^\alpha = \tilde{v}^* - t_K \frac{\sigma_*}{\sqrt{N}},$$

and t_K is defined as

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t_K} e^{-\frac{t^2}{2}} dt = \alpha^{\frac{1}{K}}.$$

We obtain the probabilistic worst-case α lower bound $(\tilde{v}^* - \Delta_W^\alpha)$ for z^*:

$$\text{prob}(\tilde{v}^* - \Delta_W^\alpha \leq z^*) \geq \alpha.$$

6.3.6. The probabilistic upper bound

Having found the approximately optimal solution $\hat{x}^\ell$ we use an independent sample S^ℓ to independently estimate the expected second-stage cost associated with $\hat{x}^\ell$,

$$\tilde{z}(\hat{x}^\ell) = \frac{1}{N} \sum_{\omega \in S^\ell} z^\omega(\hat{x}^\ell),$$

and its variance

$$\tilde{\sigma}_{\tilde{z}}^2(\hat{x}^\ell) = \frac{1}{N(N-1)} \sum_{\omega \in S^\ell} (z^\omega(\hat{x}^\ell) - z(\hat{x}^\ell))^2.$$

An α upper confidence interval for z^* is estimated as

$$\text{prob}(\tilde{z}(\hat{x}^\ell) + t\tilde{\sigma}_{\tilde{z}}(\hat{x}^\ell) \geq z^*) \geq \alpha,$$

where t is defined as

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{t^2}{2}} dt = \alpha.$$

6.4. Using Monte Carlo pre-sampling for solving an approximate problem

We refer to Monte Carlo pre-sampling, when we take a sample $\omega \in S$ of size $|S| = N$ according to the original probability distribution p^ω, and solve the approximate problem based on the obtained sample:

$$\begin{array}{rcrcrcl}
\min\ z & = & cx & + & \frac{1}{N}\sum_{\omega \in S} f^\omega y^\omega & & \\
s/t & & Ax & & & = & b \\
 & & -B^\omega x & + & D^\omega y^\omega & = & d^\omega \\
 & & x, y^\omega & & & \geq & 0, \quad \omega \in S.
\end{array}$$

For solving the approximate problem, DECIS uses Benders decomposition. The master problem, the subproblems and the cuts for the approximate problem are summarized as follows.

The approximate master problem:

$$\begin{array}{rrcrcl} v^K & = & \min\ cx & + & \theta & \\ \rho: & & Ax & & & = \ b \\ \lambda^k: & & -G^k x & + & \theta & \geq \ g^k,\ k=1,\dots,K \\ & & x, & & \theta & \geq \ 0, \end{array}$$

where ρ and λ^k are optimal dual multipliers.

The sub problems:

$$\begin{array}{rcrcl} z^\omega(\hat{x}^k) & = & \min\ f^\omega y^\omega & & \\ \pi^\omega(\hat{x}^k): & & D^\omega y^\omega & = & d^\omega + B^\omega \hat{x}^k \\ & & y^\omega & \geq & 0,\ \omega \in S, \end{array}$$

where $\pi^\omega(\hat{x}^k)$ is the optimal dual solution of subproblem ω, given $\hat{x}^k$, the first stage decision passed on iteration k to the subproblems.

The expected second-stage costs and parameters of the cuts:

$$\begin{array}{rcl} z(\hat{x}^k) & = & \sum_{\omega\in S} z^\omega(\hat{x}^k) \\ G^{k+1} & = & \sum_{\omega\in S} \pi^\omega(\hat{x}^k) B^\omega \\ g^{k+1} & = & z(\hat{x}^k) - G\hat{x}^k \end{array}$$

The bounds for the approximate problem

$$\begin{array}{rcl} z_{LB}^k & = & z_M = c\hat{x}^k + \hat{\theta}^k \\ z_{UB}^k & = & \min\{z_{UB}^{k-1}, c\hat{x}^k + z(\hat{x}^k)\},\quad z_{UB}^0 = \infty. \end{array}$$

The algorithm proceeds in the same way as in the universe case for $\omega \in S$ and $p^\omega = 1/N$ for all $\omega \in S$, and is therefore a deterministic algorithm. The algorithm is started with $\hat{x}^0$ obtained from solving the relaxed master problem $\min cx$ subject to $Ax = b, x \geq 0$, or obtained as the optimal solution of the expected value problem. By solving the master problem on iteration k a trial solution $\hat{x}^k$, $\hat{\theta}^k$ is obtained. Given $\hat{x}^k$ we solve N subproblems $\omega \in S$ to compute the optimal expected second-stage costs $z(\hat{x}^k)$ and the coefficients G^{k+1} and the right-hand side g^{k+1} of a cut. Cuts are sequentially added to the master problem. On each iteration k of the Benders decomposition algorithm a trial solution $\hat{x}^k$, a lower bound z_{LB}^k, and an upper bound z_{UB}^k for the approximate problem is obtained. The algorithm proceeds until on iteration $k = K$

$$\frac{|z_{UB}^k - z_{LB}^k|}{|z_{LB}^k|} \leq \text{TOL},$$

the optimality criterion is met. The TOL optimal solution of the approximate problem is determined as

$$\hat{x}^\ell \in \arg\min\ \{c\hat{x}^k + z(\hat{x}^k),\quad k = 1,\dots,K\},$$

the best solution found. Since $\hat{x}^\ell$ is the optimal solution of the approximate problem, but not optimal for the original problem, we next determine how good this solution is with respect to the original problem and estimate a $\alpha = 0.95$ confidence interval based on probabilistic upper and lower bounds on the optimal objective of the stochastic problem.

6.4.1. The probabilistic lower bound

The theory of probabilistic bounds in the case of pre-sampling is discussed in Infanger (1997) [12]. In order to understand the following discussion, note that the expected second-stage cost $z(\hat{x}^k)$ and the coefficients G^k and the right hand side g^k of each cut k are exact values with regard to the sampled problem, but are estimates based on sample S with regard to the true stochastic problem. For the derivation of the probabilistic bounds we see these as estimates based on sample S with regard to the stochastic problem. Basis for the derivation of the lower bound is the true system of inequalities below:

$$\begin{array}{rcrcrcl} z^* & = & cx^* & + & \theta^* & & \\ & & Ax^* & & & = & b \\ & & -G^k x^* & + & \theta^* & \geq & g^k - \tilde{\epsilon}_*, \quad k = 1, \dots, K \\ & & x^* & & & \geq & 0, \end{array}$$

where

$$\tilde{\epsilon}_* = \sum_{\omega \in S} z^\omega(x^*) - z(x^*),$$

is the difference between the estimated optimal expected second stage costs based on sample S and its true expected value, and x^* denotes the true optimal solution of the original problem. Since we are using on each iteration of the Benders decomposition algorithm the same sample S for computing $z(\hat{x}^k)$, G^{k+1}, and g^{k+1}, the error terms $\tilde{\epsilon}_*$ are all the same and do not depend on k.

The optimal solution of the approximate master problem $v^* = \min v$, and the optimal dual multipliers ρ, $\lambda^1, \dots, \lambda^K$ satisfy

$$v^* = \rho b + \sum_{k=1}^{K} \lambda^k g^k$$

$$\rho A - \sum_{k=1}^{K} \lambda^k G^k + \gamma = c, \quad \gamma \geq 0$$

$$\sum_{k=1}^{K} \lambda^k = 1$$

$$\lambda^k \geq 0, \quad k = 1, \dots, K.$$

Applying these multipliers to the corresponding relations in the system of inequalities above and subtracting from the first relation, we obtain

$$0 \leq \gamma x^* \leq z^* - (\rho b + \sum_{k=1}^{K} \lambda^k g^k) + \sum_{k=1}^{K} \lambda^k \tilde{\epsilon}_*.$$

Dropping $\gamma x^* \geq 0$, substituting v^*, noting that $\sum_{k=1}^K \lambda^k \tilde{\epsilon}_* = \tilde{\epsilon}_*$, since $\sum_{k=1}^K \lambda^k = 1$, and rearranging terms, we obtain a lower bound for the optimal objective z^*:

$$v^* - \tilde{\epsilon}_* \leq z^*$$

The error term $\tilde{\epsilon}_*$ is the difference of the average of N independently drawn observations with replacements *from the same distribution* of the optimal second-stage cost $z^\omega(x^*)$ minus its true mean $z(x^*)$. We do not know its true value. By the central limit theorem, for sufficiently large sample sizes, $\tilde{\epsilon}_*$ is approximately normally distributed:

$$\tilde{\epsilon}_* \sim N(0, \frac{\sigma_*}{\sqrt{N}}) = \frac{\sigma_*}{\sqrt{N}} N(0,1),$$

where

$$\sigma_*^2 = \sum_{\omega \in \Omega} (z^\omega(x^*) - z(x^*))^2 p^\omega$$

is the (population) variance of the optimum second stage cost.

We obtain a probabilistic α lower bound for z^*:

$$\text{prob}(v^* - t\frac{\sigma_*}{\sqrt{N}} \leq z^*) \geq \alpha.$$

where t is defined as

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{t^2}{2}} dt = \alpha.$$

We estimate σ_*^2 by the estimated variance of the second stage cost of solution $\hat{x}^\ell$, the best solution found,

$$\tilde{\sigma}^2(\hat{x}^\ell) = \frac{1}{N-1} \sum_{\omega \in S} (z^\omega(\hat{x}^\ell) - z(\hat{x}^\ell))^2,$$

and determine an estimate of the probabilistic α lower bound as

$$\text{prob}(\tilde{v}^* - t\frac{\tilde{\sigma}(\hat{x}^\ell)}{\sqrt{N}} \leq z^*) \geq \alpha.$$

6.4.2. The probabilistic upper bound

Having found the approximately optimal solution $\hat{x}^\ell$ we use an independent sample S^ℓ to independently estimate the expected second-stage cost associated with $\hat{x}^\ell$,

$$z(\hat{x}^\ell) = \frac{1}{N} \sum_{\omega \in S^\ell} z^\omega(\hat{x}^\ell),$$

and its variance as

$$\tilde{\sigma}^2(\hat{x}^\ell) = \frac{1}{N-1} \sum_{\omega \in S^\ell} (z^\omega(\hat{x}^\ell) - z(\hat{x}^\ell))^2.$$

An α upper confidence interval for z^* is estimated as

$$\text{prob}(c\hat{x}^\ell + \tilde{z}(\hat{x}^\ell) + t\frac{\tilde{\sigma}(\hat{x}^\ell)}{\sqrt{N}} \geq z^*) \geq \alpha,$$

where t is defined as

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{t^2}{2}} dt = \alpha.$$

6.5. Regularized decomposition

Sometimes the number of Benders iterations necessary for solving a problem can be reduced significantly by regularizing (see, e.g., Ruszczynski (1986) [14]) the decomposition through controlling the distance between a new trial solution, and the best solution found as the algorithm proceeds. This is accomplished through adding a nonlinear term in the master problem, calculating the square norm between of the difference between the variable x and the best solution found $\hat{x}_k^\ell$. The way we carry out regularized decomposition depends on if a deterministic (solving the universe, the expected value, or a pre-sampled problem) or a stochastic (Monte Carlo sampling) strategy is used.

6.5.1. Regularized decomposition for deterministic solution strategies

As deterministic solution strategies we consider solving the universe problem, solving the expected value problem, and solving a pre-sampled problem. For these, expected cost, cuts and bounds are all computed exactly with regard to the problem that is solved using decomposition. The regularized master, the bounds, and the stopping criteria are summarized as follows:

The regularized master problem:

$$\begin{array}{lrcll} \min & cx & + & \theta + \frac{1}{2\rho}(x - \hat{x}_K^\ell)^2 & & \\ & Ax & & & = & b \\ & -G^k x & + & \theta & \geq & g^k,\ k = 1, \dots, K \\ & x, & & \theta & \geq & 0. \end{array}$$

The bounds

$$\begin{array}{rcl} z_{LB}^k & = & z_M^k = c\hat{x}^k + \hat{\theta}^k \\ z_{UB}^k & = & \min\{z_{UB}^{k-1}, c\hat{x}^k + z(\hat{x}^k)\}, \quad z_{UB}^0 = \infty, \end{array}$$

where $z(\hat{x}^k) = E\ z^\omega(\hat{x}^k)$, and

$$\begin{array}{rcrcrcl} z_M^K & = & \min\ cx & + & \theta & & \\ & & Ax & & & = & b \\ & & -G^k x & + & \theta & \geq & g^k,\ k = 1, \dots, K \\ & & x, & & \theta & \geq & 0, \end{array}$$

is obtained by additionally solving the master problem without regularization.

On each iteration the best solution found is used as the fixed point for the regularization, i.e.,

$$\hat{x}_K^\ell \in \arg\min\ \{c\hat{x}^k + z(\hat{x}^k), \quad k = 1, \dots, K\},$$

the solution corresponding to the best upper bound. The algorithm proceeds until on iteration $k = K$

$$\frac{1}{2\rho}(x - \hat{x}_K^\ell)^2 \leq \text{TOL},$$

and the solution $\hat{x}_K^\ell$ is a TOL optimal solution of the problem.

6.5.2. Regularized decomposition for stochastic solution strategies

As stochastic solution strategies we consider all different variants of using Monte Carlo sampling within the Benders decomposition algorithm. For these, expected costs, cuts, and bounds are all estimates, and the solutions obtained from the master problem do not necessarily converge to the fixed point of the best solution found, but may vary due to the estimation error of each cut added to the master problem. The regularized master, the bounds, and the stopping criteria are summarized as follows:

The regularized pseudo master problem:

$$\begin{array}{lrcll} \min & cx & + & \theta + \frac{1}{2\rho}(x - \hat{x}_K^\ell)^2 & \\ & Ax & & & = & b \\ & -\tilde{G}^k x & + & \theta & \geq & \tilde{g}^k, \; k = 1, \ldots, K \\ & x, & & \theta & \geq & 0. \end{array}$$

The upper bound

$$z_{UB}^k \;\; = \;\; \min\{z_{UB}^{k-1}, c\hat{x}^k + z(\hat{x}^k)\}, \;\; z_{UB}^0 = \infty,$$

where $z(\hat{x}^k) = E\ z(\hat{x}^k)$.

The pseudo master

$$\begin{array}{rcrcrcl} \tilde{v}^K & = & \min\ cx & + & \theta & & \\ & & Ax & & & = & b \\ & & -G^k x & + & \theta & \geq & g^k, \; k = 1, \ldots, K \\ & & x, & & \theta & \geq & 0, \end{array}$$

On each iteration the best solution found is used as the fixed point for regularization, i.e.,

$$\hat{x}_K^\ell \in \arg\min\ \{c\hat{x}^k + z(\hat{x}^k), \;\; k = 1, \ldots, K\},$$

the solution corresponding to the best upper bound. The algorithm proceeds until z_{UB}^k and $\tilde{v}^k$ are sufficiently close, which is tested using a Student-t test. We declare the best solution found $\hat{x}^\ell = \hat{x}_K^\ell$ as the approximately optimal solution of the problem and compute an $\alpha = 0.95$ confidence interval based on probabilistic upper and lower bounds as discussed in Section 6.3.4 above.

7. Acknowledgments

The software greatly benefitted from tests on a large model for gas contracting decisions conducted by John Stone at Pacific Gas and Electric Company (PG&E). The author wishes to thank John Stone, who wrote interface routines to CPLEX and suggested valuable improvements to a previous version of the software. The author wishes to thank Michael Saunders for his suggestions regarding the interface to MINOS. The author is greatful to Bob Entiken and Bert Hackney for their valuable editorial suggestions regarding a previous version of this user's manual. The author is greatful to Arve Halseth of ECON, Norway, for extensively testing the software on a large model of the Norwegian power market. The author wishes to acknowledge all the interesting discussions with George Dantzig, Peter Glynn, Michael Saunders, and John Stone.

A. Illustrative example

An illustrative example of a stochastic electric power expansion planning problem, APL1P, is discussed in Infanger (1994) [11]. We will use this example to exemplify the stochastic problem, the definition of the uncertain parameters, the input format and the output obtained by DECIS.

In this model, there are two generators with different investment and operating costs, and the demand is given by a load duration curve with three load levels: base, medium and peak. We index the generators with $j = 1, 2$, and the demands with $i = 1, 2, 3$. The variables x_j, $j = 1, 2$, denote the capacities that can be built and operated to meet demands d_i, $i = 1, 2, 3$. The per-unit cost to build generator j is c_j. The variable y_{ij} denotes the operating level for generator j in load level i, with operating cost f_{ij}. The variable y_{is} defines the unserved demand in load level i that must be purchased with penalty cost $f_{is} > f_{ij}$ for all j. The subscript s is not an index, but denotes only an unserved demand variable. In this way the model is formulated with complete recourse, which means that for any given choice of x, demand is "satisfied" for all outcomes. Building new generators competes with purchasing unserved demand through the cost function, yet there is a minimum capacity b_i that has to be built for each load level. The availabilities of the two generators, β_j, $j = 1, 2$, and the demands in each load level, d_i, $i = 1, 2, 3$, are uncertain. The resulting model APL1P is formulated as follows:

$$
\begin{array}{lccccccll}
\min & \sum_{j=1}^{2} c_j x_j & + & E\,\{\sum_{j=1}^{2}\sum_{i=1}^{3} f_{ij} y_{ij}^{\omega} & + & \sum_{i=1}^{3} f_{is} y_{is}^{\omega}\} & & & \\
 & x_j & & & & & \geq & b_j & j = 1, 2 \\
 & -\beta_j^{\omega} x_j & + & \sum_{i=1}^{3} y_{ij}^{\omega} & & & \leq & 0, & j = 1, 2 \\
 & & & \sum_{j=1}^{2} y_{ij}^{\omega} & + & y_{is}^{\omega} & \geq & d_i^{\omega}, & i = 1, 2, 3 \\
 & x_j, & & y_{ij}^{\omega}, & & y_{is}^{\omega} & \geq & 0, & j = 1, 2, \\
 & & & & & & & & i = 1, 2, 3.
\end{array}
$$

You can easily detect the structure of the two-stage stochastic linear program indroduced above. As the first-stage decision we choose the capacities of the generators. In the second-stage we decide how to operate the power system, i.e., assign the generator output and "buy unserved demand" to meet the different loads, given the capacities of the generators. For making the first-stage decision only the distributions of the availability of the generators and the distributions of the demands are known. For making the second-stage decision the particular outcome of generator availability and demand in the three load segments are known.

The data for the problem are given in the following Table 1. The stochastic parameters are β_1, β_2, d_1, d_2, and d_3, all assumed to be independent random parameters. Since $|\Omega_1| = 4$, $|\Omega_2| = 5$, $|\Omega_3| = 4$, $|\Omega_4| = 4$, and $|\Omega_5| = 4$, we have $4 \times 5 \times 4 \times 4 \times 4 = 1280$ different possible random vector outcomes. In the equivalent deterministic form the problem has 11,522 variables and 6,402 constraints. The optimal solution of the problem is $x_1^* = 1800$ and $x_2^* = 1571.4$. The optimal objective value is $z^* = 24642.3$ summing $cx^* = 11128.6$ capacity cost plus $E\ z^{\omega}(x^*) = 13513.7$ expected operating cost.

Generator Capacity Costs (10^5\$/(MW, a))					
$c_1 = 4.0, \quad c_2 = 2.5$					
Generator Operating Costs (10^5\$/MW, a)					
$f_{11} = 4.3 \quad f_{21} = 8.7$					
$f_{12} = 2.0 \quad f_{22} = 4.0$					
$f_{13} = 0.5 \quad f_{23} = 1.0$					
Unserved Demand Penalties (10^5\$/MW, a)					
$f_{1s} = f_{2s} = f_{3s} = 10.0$					
Minimum Generator Capacities (MW)					
$b_1 = b_2 = 1000$					
Availability generator 1 (β_1)					
ω_1	1	2	3	4	
Outcome	1.0	0.9	0.5	0.1	
Probability	0.2	0.3	0.4	0.1	
Availability generator 2 (β_2)					
ω_2	1	2	3	4	5
Outcome	1.0	0.9	0.7	0.1	0.0
Probability	0.1	0.2	0.5	0.1	0.1
Demands d_1 (MW)					
ω_3	1	2	3	4	
Outcome	900	1000	1100	1200	
Probability	0.15	0.45	0.25	0.15	
Demands d_2 (MW)					
ω_4	1	2	3	4	
Outcome	900	1000	1100	1200	
Probability	0.15	0.45	0.25	0.15	
Demands d_3 (MW)					
ω_5	1	2	3	4	
Outcome	900	1000	1100	1200	
Probability	0.15	0.45	0.25	0.15	

Table 1: Example APL1P, problem data

B. A Large-scale problem in transportation

An example of a practical large-scale stochastic model is problem TR20. It is a test-problem from the area of LTL (less than truck load) vehicle allocation and scheduling under uncertain demand. The problem considers 20 cities around a consolidation center. At any given day, there are demands for shipments from the consolidation center to each of the 20 cities, and demands from each of the 20 cities to the consolidation center. The model considers the optimal allocation of tractors and trailers to the consolidation center and the cities, the scheduling to meet the demand, and empty movements. The model is formulated as a two-stage stochastic linear program, where the first-stage decision concerns the allocation of tractors and trailers to the different locations without knowledge of any realizations of the next day's demand distribution, and in the second-stage, the best scheduling of tractors and trailers, given their position, for known demand realizations is computed. The objective is to minimize total expected cost (operating cost and penalty cost for unserved demand). There are 40 uncertain demands. Each demand is modeled as an independent random variable with 3 outcomes each; therefore the total number of possible demand realizations is $3^{40} = 1.2 \cdot 10^{19}$. The following Table 2 represents the results from solving the model using DECIS:

# iter stoch. (exp. val.)		1407 (853)
sample size		250
exp. val. solution obj		269.2
exp. val. solution, exp. cost		309.7
stochastic solution, obj		284.5
est.	conf. left	0.04 (0.14 %)
	conf. right	0.25 (0.09 %)
Problem Size		
Master	rows	3
	columns	64
Sub	rows	125
	columns	764
# stochastic parameters		40
# universe scenarios		$1.2 \cdot 10^{19}$

Table 2: Computational results problem TR20

The results were obtained using strategy 3, solving the expected value problem first, and then using Monte Carlo importance sampling for solving the stochastic problem. The sample size used was 250 out of $1.2 \cdot 10^{19}$ universe scenarios (in each Benders iteration). Solving the expected value problem resulted in an objective value of 269.2. The true expected cost corresponding to the optimal solution of the expected value problem was estimated as 309.7. Solving the stochastic problem led to an optimal solution with an expected cost of 284.5. Compared to the expected cost of the optimal solution of the expected value problem, the stochastic solution resulted in approximately 10% smaller cost. The 95% confidence interval (within which z^*, the true optimal solution of the problem, lies with 95% confidence) was estimated as $284.1 \leq z^* \leq 284.75$. Note how small this confidence interval is.

C. Error messages

1. ERROR: need icm columns for master
   The number of columns of the master, icm, exceeds the dimension. The problem you want to solve is too big for the dimension of your version of DECIS. Obtain a new version of DECIS with increased column dimensions.

2. ERROR: need irm+nzrows rows for master
   The number of rows of the master problem exceeds the dimension. The number of rows includes the original rows of the master problem (irm) plus the number rows for place-holders for cuts (nzrows). The parameter nzrows is set in the input file. Reducing nzrows to a smaller number may help. Otherwise, obtain a new version

3. ERROR: need imast nonzeros for master
   The number of nonzeros of the master problem exceeds the dimension. The problem you want to solve is too big for the dimension of your version of DECIS. Obtain a new version of DECIS with increased dimensions.

4. ERROR: need ics columns for sub
   The number of columns of the subproblem, ics, exceeds the dimension. The problem you want to solve is too big for the dimension of your version of DECIS. Obtain a new version of DECIS with increased dimensions.

5. ERROR: need irs rows for sub
   The number of rows of the subproblem, irs, exceeds the dimension. The problem you want to solve is too big for the dimension of your version of DECIS. Obtain a new version of DECIS with increased dimension.

6. ERROR: need isub nonzeros for sub
   The number of nonzeros of the subproblem exceeds the dimension. The problem you want to solve is too big for the dimension of your version of DECIS. Obtain a new version of DECIS with increased dimensions.

7. ERROR: need icm columns for B matrix
   The number of columns of the B-matrix, icm, exceeds the dimension. The problem you want to solve is too big for the dimension of your version of DECIS. Obtain a new version of DECIS with increased dimensions.

8. ERROR: need ielb nonzeros for B-matrix
   The number of nonzeros in the B-matrix exceeds the dimension. The problem you want to solve is too big for the dimension of your version of DECIS. Obtain a new version of DECIS with increased dimensions.

9. ERROR: need nprep universe size
   The number of preparatory scenarios (nprep) exceeds the dimension. Preparatory scenarios are necessary for computing the marginal approximation function when using importance sampling or control variates. Obtain a version of DECIS with a larger maximum universe size.

10. ERROR: need nsamples universe size
    The number of universe scenarios exceeds the dimension. There is a maximum number of universe scenarios for which your implementation of DECIS has been dimensioned. Use a sampling based solution strategy. Obtain a version of DECIS with a larger maximum universe size.

11. ERROR: need msv stochastic parameters
    The number of stochastic parameters exceed the dimension. Obtain a version of DECIS with a larger maximum number of stochastic parameters.

12. ERROR: need nsoc(i) outcomes, parameter i
    The number of outcomes of random parameter i exceeds the dimension. Obtain a version of DECIS with a larger maximum number outcomes for each stochastic parameter.

13. ERROR: need itele random elements
    The total number of random elememt outcomes that need to be stored for generating the distribution exceeds the dimension. Obtain a version of DECIS with a larger maximum number of random element outcomes.

14. ERROR in MODEL.STO: kwd, word1, word2 was not matched in first realization of block
    The specification of the stochastic parameters is incorrect. The stochastic parameter has not been specified in the specification of the first outcome of the block. When specifying the first outcome of a block always include all stochastic parameters corresponding to the block.

15. Option word1 word2 not supported
    You specified an input distribution in the stochastic file that is not supported. Check the DECIS manual for supported distributions.

16. Error in time file
    The time file is not correct. Check the file MODEL.TIM. Check the DECIS manual for the form of the time file.

17. ERROR in MODEL.STO: stochastic RHS for objective, row name2
    The specification in the stochastic file is incorrect. You attempted to specify a stochastic right-hand side for the objective row (row name2). Check file MODEL.STO.

18. ERROR in MODEL.STO: stochastic RHS in master, row name2
    The specification in the stochastic file is incorrect. You attempted to specify a stochastic right-hand side for the master problem (row name2). Check file MODEL.STO.

19. ERROR in MODEL.STO: col not found, name1
    The specification in the stochastic file is incorrect. The entry in the stochastic file, name1, is not found in the core file. Check file MODEL.STO.

20. ERROR in MODEL.STO: invalid col/row combination, (name1/name2)
    The stochastic file (MODEL.STO) contains an incorrect specification.

21. ERROR in MODEL.STO: no nonzero found (in B or D matrix) for col/row (name1, name2)
    There is no nonzero entry for the combination of name1 (col) and name2(row) in the B-matrix or in the D-matrix. Check the corresponding entry in the stochastic file (MODEL.STO). You may want to include a nonzero coefficient for (col/row) in the core file (MODEL.COR).

22. ERROR in MODEL.STO: col not found, name2
    The column name you specified in the stochastic file (MODEL.STO) does not exist in the core file (MODEL.COR). Check the file MODEL.STO.

23. ERROR in MODEL.STO: stochastic bound in master, col name2
    You specified a stochastic bound on first-stage variable name2. Check file MODEL.STO.

24. ERROR in MODEL.STO: invalid bound type (kwd) for col name2
    The bound type, kwd, you specified is invalid. Check file MODEL.STO.

25. ERROR in MODEL.STO: row not found, name2
    The specification in the stochastic file is incorrect. The row name, name2, does not exist in the core file. Check file MODEL.STO.

26. ERROR: problem infeasible
    The problem solved (master- or subproblem) turned out to be infeasible. If a subproblem is infeasible, you did not specify the problem as having the property of "complete recourse". Complete recourse means that whatever first-stage decision is passed to a subproblem, the subproblem will have a feasible solution. It is the best way to specify a problem, especially if you use a sampling based solution strategy. If DECIS encounters a feasible subproblem, it adds a feasibility cut and continues the execution. If DECIS encounters an infeasible master problem, the problem you specified is infeasible, and DECIS terminates. Check the problem formulation.

27. ERROR: problem unbounded
    The problem solved (master- or subproblem) turned out to be unbounded. Check the problem formulation.

28. ERROR: error code: inform
    The solver returned with an error code from solving the problem (master- or subproblem). Consult the users' manual of the solver (MINOS or CPLEX) for the meaning of the error code, inform. Check the problem formulation.

29. ERROR: while reading SPECS file
    The MINOS specification file (MINOS.SPC) containes an error. Check the specification file. Consult the MINOS user's manual.

30. ERROR: reading mps file, mpsfile
    The core file mpsfile (i.e., MODEL.COR) is incorect. Consult the DECIS manual for instructions regarding the MPS format.

31. ERROR: row 1 of problem ip is not a free row
    The first row of the problem is not a free row (i.e., is not the objective row). In order to make the fist row a free row, set the row type to be 'N'. Consult the DECIS manual for the MPS specification of the problem.

32. ERROR: name not found = nam1, nam2
    There is an error in the core file (MODEL.COR). The problem cannot be decomposed correctly. Check the core file and check the model formulation.

33. ERROR: matrix not in staircase form
    The constraint matrix of the problem as specified in core file (MODEL.COR) is not in staircase form. The first-stage rows and columns and the second-stage rows and columns are mixed within each other. Check the DECIS manual as to how to specify the core file. Check the core file and change the order of rows and columns.

References

[1] Benders, J.F. (1962): Partitioning Procedures for Solving Mixed-Variable Programming Problems, *Numerische Mathematic 4, 238-252*.

[2] Birge, J.R., Dempster, M.H.A., Gassmann, H.I., Gunn, E.A., King, A.J., and Wallace, S.W. (1989): A Standard Input Format for Multiperiod Stochastic Linear Programs. *Mathematical Programming Society COAL Newsletter*, 17, pp. 1–19.

[3] Brooke, A., Kendrik, D. and Meeraus, A. (1988): *GAMS, A Users Guide*, The Scientific Press, South San Francisco, California.

[4] CPLEX Optimization, Inc. (1989–1997): *Using the CPLEX Callable Library*, 930 Tahoe Blvd. Bldg. 802, Suite 279, Incline Village, NV 89451, USA.

[5] Dantzig, G.B. and Glynn, P.W. (1990): Parallel Processors for Planning Under Uncertainty, *Annals of Operations Research* 22, 1-21.

[6] Dantzig, G.B. and Infanger, G.(1995): A Probabilistic Lower Bound for Two-Stage Stochastic Programs, Technical Report SOL 95-6, Department of Operations Research, Stanford University, Stanford, CA.

[7] Entriken, R. and Stone, J. (1997): Smps.pl User's Guide, webpage http://www-leland.stanford.edu/ entriken/smps/, Department of Engineering-Economic Systems and Operations Research, Stanford University, Stanford CA.

[8] Fourer, R., Gay, D.M., and Kernighan, B.W. (1992): *AMPL: A Modeling Language for Mathematical Programming*, The Scientific Press, South San Francisco, California.

[9] International Business Machines, Inc. (1988): *IBM Mathematical Programming System Extended/370 Version 2 (MPSX/370/V2) General Information Manual*.

[10] Infanger, G. (1992): Monte Carlo (Importance) Sampling within a Benders Decomposition Algorithm for Stochastic Linear Programs, *Annals of Operations Research* 39, 69–95.

[11] Infanger, G. (1994): *Planning Under Uncertainty – Solving Large-Scale Stochastic Linear Programs*, The Scientific Press Series, Boyd and Fraser.

[12] Infanger, G. (1997): Sampling versus Pre-sampling for Stochastic Linear Programs, Technical Report SOL 97-2, Department of Engineering-Economic Systems and Operations Research, Stanford University, Stanford, CA.

[13] Murtagh, B.A. and Saunders, M.A. (1983): MINOS User's Guide, SOL 83-20, Department of Operations Research, Stanford University, Stanford CA 94305.

[14] Ruszczynski, A. (1986): A Regularized Decomposition Method for Minimizing a Sum of Polyhedral Functions, *Mathematical Programming* 35, 309–333.

[15] Van Slyke and Wets, R.J. (1969): L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming, *SIAM Journal of Applied Mathematics*, Vol 17, 638-663.

License and Warranty

The software, which accompanies this license (the “Software”) is the property of Gerd Infanger and is protected by copyright law. While Gerd Infanger continues to own the Software, you will have certain rights to use the Software after your acceptance of this license. Except as may be modified by a license addendum, which accompanies this license, your rights and obligations with respect to the use of this Software are as follows:

- You may

  1. Use one copy of the Software on a single computer,
  2. Make one copy of the Software for archival purposes, or copy the software onto the hard disk of your computer and retain the original for archival purposes,
  3. Use the Software on a network, provided that you have a licensed copy of the Software for each computer that can access the Software over that network,
  4. After a written notice to Gerd Infanger, transfer the Software on a permanent basis to another person or entity, provided that you retain no copies of the Software and the transferee agrees to the terms of this agreement.

- You may not

  1. Copy the documentation, which accompanies the Software,
  2. Sublicense, rent or lease any portion of the Software,
  3. Reverse engineer, de-compile, disassemble, modify, translate, make any attempt to discover the source code of the Software, or create derivative works from the Software.

Limited Warranty:

Gerd Infanger warrants that the media on which the Software is distributed will he free from defects for a period of thirty (30) days from the date of delivery of the Software to you. Your sole remedy in the event of a breach of the warranty will be that Gerd Infanger will, at his option, replace any defective media returned to Gerd Infanger within the warranty period or refund the money you paid for the Software. Gerd Infanger does not warrant that the Software will meet your requirements or that operation of the Software will be uninterrupted or that the Software will be error-free.

THE ABOVE WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

Disclaimer of Damages:

REGARDLESS OF WHETHER ANY REMEDY SET FORTH HEREIN FAILS OF ITS ESSENTIAL PURPOSE, IN NO EVENT WILL GERD INFANGER BE LIABLE TO YOU FOR ANY SPECIAL, CONSEQUENTIAL, INDIRECT OR SIMILAR DAMAGES, INCLUDING ANY LOST PROFITS OR LOST DATA ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF GERD INFANGER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IN NO CASE SHALL GERD INFANGER’S LIABILITY EXCEED THE PURCHASE PRICE FOR THE SOFTWARE. The disclaimers and limitations set forth above will apply regardless of whether you accept the Software.

General:

This Agreement will be governed by the laws of the State of California. This Agreement may only be modified by a license addendum, which accompanies this license or by a written document, which has been signed by both you and Gerd Infanger. Should you have any questions concerning this Agreement, or if you desire to contact Gerd Infanger for any reason, please write:

Gerd Infanger, 1590 Escondido Way, Belmont, CA 94002, USA.