An Extension of the Multicut L-Shaped Method

INEN 698 - Large-Scale Stochastic Optimization Semester project

Svyatoslav Trukhanov

December 13, 2005
Contents

1 Introduction and Literature Review ........................................ 3
2 Formal Problem Statement .................................................... 5
3 Solution Approach .................................................................. 6
   3.1 L-shaped Method .......................................................... 7
   3.2 Multicut L-shaped Method ............................................. 8
   3.3 Multicut L-shaped method with partial cut aggregation ....... 9
     3.3.1 Partial cut aggregation .......................................... 9
     3.3.2 Aggregation rule .................................................. 10
     3.3.3 Implementation .................................................. 11
4 Computational Experiments .................................................. 13
5 Conclusions and Future Work .............................................. 15
6 References ........................................................................... 16
7 Team Member Critique ......................................................... 17
8 Appendix ............................................................................... 18
   8.1 Iteration, runtime and bounds plot for problem 4node ......... 18
   8.2 More numerical results .................................................. 21
Abstract

The L-shaped method allows to solve large scale stochastic problems that cannot be solved directly. Each major iteration method solves all scenario subproblems and generate a single optimality or feasibility cut to the master problem by aggregating all the cuts obtained from the subproblems. This causes an "information loss" and results in a large number of major iterations. The multicut L-shaped method of Birge and Louveaux (1988) allows to add all cuts at once to the master problem. This potentially reduces the number of major iterations but increase the number of decision variables in the master problem drastically (each cut requires one new variable). Also the number of constraints added each iteration is huge. These facts make multicut method useless for practical application. This project involves making an extension to make partial aggregation of the cuts instead of full aggregation (as in original L-shaped) or no aggregation (as multicut L-shaped does). The main idea is to find a compromise between the master problem size and the number of major iteration that minimizes total runtime.

1 Introduction and Literature Review

The classical two-stage stochastic linear program with fixed recourses was introduced by Dantzig in 1955[4]. Fixed recourse is the most important assumptions here, it yields the convexity property of the objective function. Since objective function is convex nonlinear convex optimization methods may be applied. One of such method was developed in 1960 by Kelley[9]. Method is based on adding cutting planes iteratively that approximate the objective function. So method reduces non-linear convex optimization to the sequence of linear problem, that is well-studied and may be solved effectively.

In 1962 Benders[1] adopt Kelley’s method to solve special case of convex optimization problems by decomposition (partitioning). Benders method also known as outer linearization method, may be applied to solve large scale linear problem with a special structure of constraint matrix. In this case original problem is decomposed into master problem and subproblem, both are smaller size then the original one. Fortunately deterministic equivalent formulation of two-stage stochastic linear program with fixed recourse has the constraint matrix that satisfied Benders requirements, but direct application
of Benders method to such problem produces very huge subproblem matrix and thus is not very helpful.

Further exploiting of deterministic equivalent formulation of two-stage stochastic linear program with fixed recourse constraint matrix structure (L-shaped structure) allows to develop in 1969 [11] the modification of Benders method that works well on problem. New method is called L-shaped method and separates original problem into master problem and family of subproblems. Each major iteration all subproblems must be solved and cut to master problem is generated using results obtained from subproblems. The main difference between L-shaped method and Benders method is that in L-shaped method many small subproblems are solved instead of huge one subproblem. L-shaped method performs effectively on a variety of problems. The most resource consuming part of the method is solving the huge number of subproblems on each major iteration step. Some techniques that allow to decrease the number of subproblems to be solved are described in [3, Chapter 5.4].

Another approach to increase the effectiveness of L-shaped algorithm is to decrease the number of major iteration needed to solve the problem. One possible way to do so was presented by Birge and Louveaux in 1988 and is called Multicut L-shaped method[2]. The main difference from original L-shaped method is that instead of adding one cut to master problem each major iteration using data obtained from all subproblems multicut method adds all obtained from subproblem cuts each iteration. According to authors this helps utilize all information obtained from subproblems and thus approximate master problem objective function tighter then in original L-shaped method. One big disadvantage of multicut L-shaped method is large size of master problem at the beginning and fast growth of its size during each iteration. Also multicut method requires to introduce one additional variable to master problem for each subproblem. Actually for a big number of subproblem (that is equal to the number of possible outcomes in the original problem) multicut cannot perform even one major iteration and thus is useless for solving real problems.

Proposed work tries to decrease the information loss that takes place in original L-shaped method, but at the same time does not allow to increase the size of the master problem with the same rate as multicut L-shaped method does.
2 Formal Problem Statement

Two-Stage Stochastic Linear Program with Fixed Recourse is formulated as follows:

\[(SLP - FR)\]
\[
\begin{align*}
\min & \quad c^T x + E[\tilde{f}(x, \tilde{\omega})] \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0,
\end{align*}
\]

where

\[f(x, \omega) = \min_y q^T_\omega y\]
\[\text{s.t.} \quad T_\omega x + W y = r_\omega\]
\[y \geq 0,
\]

where

- \(\Omega\) is the sample space. Important restriction about sample space is that sample space is finite.
- \(\tilde{\omega}\) is the random variable defined on \(\Omega\).
- \(\omega \in \Omega\) is the outcome.
- \(x \in \mathbb{R}^{n_1}\) is the first stage decision variable. This means that decision \(x\) must be made before observing outcome ("here and now").
- \(c \in \mathbb{R}^{n_1}\) is fixed cost associated with decision \(x\).
- \(A \in \mathbb{R}^{m_1 \times n_1}, b \in \mathbb{R}^{m_1}\) are fixed matrix and vector correspondingly that take part in \(x\) restrictions at the first stage.
- \(y \in \mathbb{R}^{n_2}\) is the second stage decision variable. Decision \(y\) is made after event occurs and was observed (\(\omega\) is known and fixed for that time) and no any uncertainty left. Decision \(y\) is restricted depending on \(x\) and \(\omega\), moreover, some additional cost is associated with \(y\).
- \(q_\omega \in \mathbb{R}^{n_2}\) is the cost associated with the second stage decision \(y\), given for \(\forall \omega \in \Omega\).
• $r_\omega \in \mathbb{R}^{m_2}$ is the right-hand side in $y$ restriction constraints, is given $\forall \omega \in \Omega$.

• $T_\omega \in \mathbb{R}^{m_2 \times n_1}$ is called technology matrix, is given $\forall \omega \in \Omega$.

• $W \in \mathbb{R}^{m_2 \times n_2}$ is the recourse matrix. The main assumptions is that despite of being used on the second stage, $W$ does not depend on $\omega$. This fact is the difference between two-stage stochastic linear problem (2-stage SLP) and two-stage stochastic linear problem with fixed recourse (2-stage SLP-FR).

• $E_{\tilde{\omega}}$ denotes the mathematical expectation with respect to $\tilde{\omega}$.

Let $S$ denotes the set of all possible realizations of $\tilde{\omega}$. $S$ is called scenarios. Since sample space is finite, $S$ is finite too. Let $s = 1, 2, \cdots, |S|$ index all possible scenarios and let $p_s$ be their probabilities. Under these assumption it is possible to write 2-stage SLP-FR as the deterministic equivalent program in the extensive form (EF for short). This form is created by associating one set of the second-stage decisions to each realization of $\tilde{\omega}$. Let $y_s$ be the second-stage decision variable, associated to scenario $s$ and $q_s$, $r_s$, $T_s$ be corresponding value of vectors and matrix under scenario $s$, then deterministic equivalent program in EF will be:

$$
(EF) \quad \min c^T x + \sum_{s=1}^{|S|} p_s q_s^T y_s
$$

$$
\text{s.t. } Ax = b
$$

$$
T_s x + W_s y = r_s, \ \forall s = 1, \cdots, |S|
$$

$$
x \geq 0, y_s \geq 0, \ \forall s = 1, \cdots, |S|
$$

3 Solution Approach

Special block (L-shaped) structure of two-stage stochastic linear program with fixed recourse in EF form allows to apply L-shaped algorithm to find a solution[11].
Block structure of the two-stage extensive form.

\[ A \]

\[ T_1 \quad W \]

\[ T_2 \quad W \]

\[ \vdots \]

\[ T_{|S|} \quad \cdots \quad W \]

### 3.1 L-shaped Method

Step 0 Initialization: given \( x^0 \), set \( k = 0 \) - iteration number.

Step 1 Solve subproblems: for \( s = 1, \ldots, |S| \) solve

\[
\min q_s^T y \\
\text{s.t. } Wy = r_s - T_s x^k \\
y \geq 0
\]

if problem is infeasible for some \( s \), generate a feasibility cut:

- get dual extreme ray \( \mu_s^k \)
- \( \alpha_k = \mu_k^T r_s \)
- \( \beta_k^T = \mu_k^T T_s \)
- go to step 2

else generate optimality cut

- get dual optimal solution \( \pi_{ks} \)
- \( \alpha_k = \sum_{s=1}^{|S|} p_s \pi_{ks}^T r_s \)
- \( \beta_k^T = \sum_{s=1}^{|S|} p_s \pi_{ks}^T T_s \)
Step 2 Solve master problem:

\[
\begin{align*}
\min & \quad c^T x + \eta \\
\text{s.t.} & \quad Ax = b \\
& \quad \beta_t^T x + \eta \geq \alpha_t, \ t \in \Theta_k \\
& \quad \beta_t^T x \geq \alpha_t, \ t \notin \Theta_k \\
& \quad x \geq 0
\end{align*}
\]

where \( \Theta_k \) at each iteration contains the iterations that generate optimality cut.

Step 3 Check stopping criterion, if criterion is not satisfied, then \( k = k + 1 \), go to step 1.

As mentioned in [2] main disadvantage of the presented algorithm is cut aggregation at step 1 that yields “information loss”.

### 3.2 Multicut L-shaped Method

The Multicut L-shaped method presented in [2] is similar to previous one but instead of aggregating cuts is adds all cuts to master problem. There are main differences between L-shaped and multicut L-shaped methods:

1. Master problem introduces additional variables for each subproblem, thus master objective will be:

\[
\min c^T x + \sum_{s=1}^{\mid S \mid} \eta_s
\]

2. After subproblem solving cuts are not aggregated:
   infeasible case: \( \alpha_k = \mu_k^T r_s \), \( \beta_k^T = \mu_k^T T_s \)
   feasible case: \( \alpha_{ks} = \pi_{ks}^T r_s \), \( \beta_{ks} = \pi_{ks}^T T_s \)

3. All cuts are added to master problem:

\[
\min c^T x + \sum_{s=1}^{\mid S \mid} \eta_s
\]
\[ s.t. \ Ax = b \]
\[ \beta^T_t x + \eta_s \geq \alpha_{ts}, \ t \in \Theta_k, \ s = 1, \ldots, |S| \]
\[ \beta^T_t x \geq \alpha_t, \ t \notin \Theta_k \]
\[ x \geq 0 \]

By sending disaggregate cuts, more detailed information is given to the first stage. The number of major iterations is expected then to be less than in the single cut method.\[3\]

### 3.3 Multicut L-shaped method with partial cut aggregation

The number of major iterations is reduced when multicut L-shaped method is used instead of regular one. But the size of first-stage program is increased. Is the number of scenarios is large (in real problem it may be thousands or ever millions) then master problem is extremely large and cannot be solved. This fact makes multicut L-shaped method useless in application to real problems.

The following two goals were aspired when the multicut method extension were built:

- Decrease the ”information loss”. This rule implies that more then one cut must be added each major iteration.

- Do not allow master problem to growth fast. This rule says that the number of cuts added each iteration must be relatively small (compared to the number of scenarios).

The method extension is based on Multicut L-shaped method, but with some changes, caused by rules above.

#### 3.3.1 Partial cut aggregation

Let \( S = S_1 \cup S_2 \cup \cdots \cup S_D \) be a partition of scenario set (so \( S_i \cap S_j = \emptyset \ \forall i \neq j \)). Furthermore, let aggregate cuts that are in the same part of the partition. In this case \( D \) cuts will be added to master problem each iteration. Also
$D$ additional variables are needed for master problem. New master problem will look like:

$$\begin{align*}
\min & \ c^T x + \sum_{d=1}^{D} \eta_d \\
n & \text{s.t. } Ax = b \\
& \beta^T_{td} x + \eta_d \geq \alpha_{td}, \ t \in \Theta_k, \ d = 1, \ldots, D \\
& \beta^T_t x \geq \alpha_t, \ t \notin \Theta_k \\
& x \geq 0
\end{align*}$$

where cut coefficients computed in following way:

infeasible case: $\alpha_k = \mu^T_k r_s$, $\beta^T_k = \mu^T_k T_s$
feasible case: $\alpha_{kd} = \sum_{s \in S_d} p_s \pi^T_{ks} r_s$, $\beta^T_{ks} = \sum_{s \in S_d} p_s \pi^T_{ks} T_s$

Very important note that scenario partitioning must be complete before starting iteration and cannot be changed during the process. If $D = 1$ then multicut L-shaped method with partial cut aggregation is just a regular L-shaped method (full aggregation), if $D = |S|$ then the method is equivalent to multicut L-shaped method (no aggregation).

### 3.3.2 Aggregation rule

Performance of the Multicut L-shaped method with partial cut aggregation is dependent on scenario partitioning. Two main questions about partitioning must be considered: how to choose the number of parts $D$ and how to make a partition of scenario space. In general it is a large problem that requires large research. Also partitioning is dependent on problem. One case particular is considered below. Assume that $\tilde{\omega}$ is a multivariate random variable with all its components are independent one dimension random variables. In this case realization can be represented as vector $s = (s^{(1)}, s^{(2)}, \ldots, s^{(t)})$. To create a partition lets choose some number $0 \leq l \leq t$ that will be called aggregation level and let two scenarios $s_1$ and $s_2$ belong to the same part if and only if they have the same first $l$ components, i.e.

$s_1 \in S_i$ and $s_2 \in S_i \iff \forall \ 1 \leq k \leq l \ s_1^{(k)} = s_2^{(k)}$

Using this rule to create a partition the number of parts is not given, but computed as

$$D = \prod_{k=1}^{t} |\tilde{s}_k|$$
where $|\tilde{s}_k|$ is the number of realization of $k$-th component of random variable. If $l = 0$ then regular L-shaped method will be obtained, if $l = t$ then multicut version of L-shaped will be as a result.

### 3.3.3 Implementation

**Input:** As input algorithm requires:

1. $c$, $q_s$, $b$, $A$, $T_s$, $r_s$, $W$ - vectors and matrices as described above, all this data is represented as a .cor file (contains whole matrices data), .tim file (contains name of rows and columns that are used to split data into stages) and .sto file that contains random variable realizations;

2. $\epsilon$ is a positive constant that used in stopping criterion;

3. $l$ is a nonnegative number, cut aggregation level, must be not large then the number of random variables in .cor file.

**Output:** Algorithm outputs:

1. table that contains values of lower, upper bounds and gap on each iteration step;

2. optimal solution and corresponding value of objective function;

3. algorithm runtime, aggregation level and number of cuts added each iteration to the master problem.

**Algorithm steps:**

1. Initialization
   (a) read data from .cor, .tim and .sto files;
   (b) setup master LP problem:
   $$\min c^T x$$
   $$s.t. \ Ax = b$$
   $$x \geq 0$$
   (c) setup subproblem (using data from .cor file and time intervals from .tim file):
   $$\min q^T y$$
   $$s.t. \ Wy = r$$
   $$y \geq 0$$
(d) Solve master problem and put its optimal solution as initial solution $x^0$;

(e) $k = 0$ - iteration number;

(f) $ub = +\infty$, $lb = -\infty$ - current upper and lower bound;

(g) Compute the number $D = \prod_{k=1}^{t} |\tilde{s}_k|$ the number of cuts that will be added each iteration;

(h) Allocate an array for storing $D$ cut coefficients;

(i) Add $D$ columns $\eta_1, \ldots, \eta_D$ to the master problem.

2. Iteration step, subproblem for each scenario:

(a) Generate scenario $s$;

(b) Update right hand side of the subproblem to $r_s - T_s x^k$;

(c) Update subproblem matrix to $T_s$ and objective coefficients to $q_s$;

(d) Solve the subproblem;

(e) If subproblem is feasible then get $f_{k_s}$ - objective value and $\pi_{k_s}$ - dual optimal solution, if subproblem is infeasible get $\mu_k$ - dual extreme ray direction;

(f) If subproblem is feasible go to step 2a.

(g) If all subproblems were feasible update the upper bound: $ub = \min(ub, e^T x^k + \sum_{s=1}^{\vert S \vert} p_s f_{k_s})$, if upper bound was decreased then save current solution $x^k$ as ”incumbent” one;

(h) compute Bender’s cut coefficients $\beta_k$ and RHS $\alpha_k$, for a feasible case $\alpha_{kd} = \sum_{s \in S_d} p_s \pi_{k_s}^T r_s$, $\beta_k^T = \sum_{s \in S_d} p_s \pi_{k_s} T_s$, $\forall d = 1, \ldots, D$ otherwise $\alpha_k = \mu_k T_s$, $\beta_k^T = \mu_k T_s$;

3. Master problem

(a) Add cuts to master problem. If subproblem was feasible then cuts are $\beta_{kd}^T x + \eta_d \geq \alpha_{kd}$, $\forall d = 1, \ldots, D$ - optimality cuts, otherwise $\beta_k^T x \geq \alpha_k$ - feasibility cut;

(b) Solve the master problem, to get $(x^{k+1}, \eta_1^{k+1}, \ldots, \eta_D^{k+1})$;

(c) Update lower bound to optimal objective value of master problem.
4. Check stopping criterion: if $ub - lb > \epsilon \cdot |lb|$ then $k = k + 1$, go to step 2.

5. Output the results. Optimal solution is the ”incumbent” solution, corresponding upper bound is the optimal objective value.

Since the number of scenario $S$ may be huge even for small problem, recursive procedure for scenario enumeration without storing all of them was implemented. To save more memory values of $\pi_{ks}$ and $f_{ks}$ were not stored until all scenario subproblems will be solved, but aggregated to existing cut after each subproblem solved.

Algorithm was implemented using C++ programming language based on classes provided by L. Ntaimo for Benders method implementation[10]. For linear program solving CPLEX callable library was used[8].

4 Computational Experiments

As mentioned in [7, 6] numerical results are very important part of research, especially when developing computational method. First of all let define the goals of numerical experiments. There are two main goals: check whether algorithm converges (because of absence of mathematical proof) and if so, then compare performance of this algorithm with base algorithm (L-shaped and multicut L-shaped). The problems for numerical testing were taken from different sources. The first set of problems is well known and was given for testing another methods implementation during the classtime. These problem instances are CEP1, PGP2, 20, STORM. The second source of stochastic linear problems with fixed recourse is the online stochastic linear problem library[5]. Most of the problems in this library is actually a mixed-integer stochastic problems, so LP-relaxation was used. All input data is presented as 3 files: .cor that contains problem description, .tim that contains time intervals for stages and .sto that contains stochastic data (random variable realizations). Other parameters, such as aggregation level and gap value were passed to the algorithm through the command line. Some problems (20, STORM) could not be solved in reasonable time due to huge number of scenario space.

Each problem were solved several times with different level of aggregation. First, problem was solved with 0 level, that means total aggregation or L-shaped method. The L-shaped method proved to converge, so the solution
obtained by L-shaped is set as base. After, the same problem was solved several times with different non-zero level aggregation. Each time obtained objective value was compared to base value (obtained by L-shaped method). All experiments show that these values are the same and thus show that Multicut L-shaped method with partial cut aggregation converges.

To achieve second goal that is performance comparison between old methods and the new one the number of iterations and the runtime were compared in the same hardware and software settings. Number of iterations is a good characteristic of method performance, but since in different cases iteration takes different time to complete the total runtime were considered too. The ratio between runtimes obtained from different algorithm settings is more important then the absolute value of runtimes. All numerical experiments were held at Intel Pentium 4 2.8GHz processor with 512Mb memory hardware system. Operating system was Microsoft Windows XP, software was compiled using Microsoft Visual C++ 7.1 compiler that is a part of Microsoft Visual .NET studio. CPLEX callable library version 9.0 was used as a dynamic linked library. Parameter $\epsilon$ was equal to $10^{-6}$ for all experiments. Runtime was measured in seconds, using default precision provided by hardware (0.01 sec). Implemented algorithm may be accessed for free in binary executable form (CPLEX library is not included due to licensing reasons) and may be used for reproduce obtained results and gathering new ones.

Table 1 presents number of major iterations and runtime for different cut aggregation level for the same problem. As estimated the number of major iterations is decreased with increasing number of cuts added each iteration. But it is not the case with runtime. Runtime decreases at the beginning but then started increases again. At the beginning (full aggregation) the most time consuming part of the algorithm is subproblem solving, master problem is small and not required much time to solve. But increasing the number of cuts yields increasing the size of master problem and now master problem solving time is not considered as small. That is why total running time is not decreasing all the time. Appendix 8.1 presents plots with number of iterations and runtime depending on aggregation level and upper-lower bounds comparison between regular L-shaped method and multicut L-shaped version with partial cut aggregation. Many other test instances produce similar behavior - iteration number is decreasing monotonously, but not a runtime. This raises a question that is good for future work - what is an optimal value of aggregation level to achieve the minimum running time required for solving a problem. Other appendix presents more results obtained for different
Table 1: Results of experiments, problem 4node, 12 independent random variables, 8192 scenarios

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>243</td>
<td>145.22</td>
<td>446.856</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>114</td>
<td>66.53</td>
<td>446.856</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>76</td>
<td>43.39</td>
<td>446.856</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>66</td>
<td>39.39</td>
<td>446.856</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>60</td>
<td>40.94</td>
<td>446.856</td>
</tr>
<tr>
<td>5</td>
<td>64</td>
<td>51</td>
<td>42.11</td>
<td>446.856</td>
</tr>
<tr>
<td>6</td>
<td>128</td>
<td>44</td>
<td>45.78</td>
<td>446.856</td>
</tr>
</tbody>
</table>

testing instances.

5 Conclusions and Future Work

Summary, multicut L-shaped algorithm with partial cut aggregation was developed. Implemented software actually is a general framework that allow to create different modifications of algorithm with different rules of aggregation. Choosing total aggregation regular L-shaped method will be obtained and choosing no aggregation results Multicut L-shaped method. One of the aggregation criteria described above was tried to perform algorithm. Numerous numerical experiments were made and their results were studied.

As a future work some directions may be considered. First of all it is a good idea to prove rigorously that described method converges. Numerical results show that method converges, but mathematical proof is required. Second, other aggregation criteria may be used to find better performing particular cases. Finally, adaptive (or dynamic) aggregation that allow to change scenario space partitioning during the iterative process may be studied.
6 References

[1] J. F. Benders. Partitioning procedures for solving mixed-variables pro-

linear programs. European Journal of Operational Research, 34:384–392,


tional experiments with mathematical software. ACM Transactions on


[8] ILOG. ILOG CPLEX: High-performance software for mathematical pro-


602/Projects/STUDENT BENDERS.zip.

tions to optimal control and stochastic programming. SIAM Journal of
7 Team Member Critique

Despite of worked on his own project ”Approximation Algorithms for 2-Stage Stochastic-Weighted Matching Problem” that is not closely related to current project, team member Balabhaskar Balasundaram made some useful comments and proposed some interest idea about this project.
8 Appendix

8.1 Iteration, runtime and bounds plot for problem 4node
8.2 More numerical results

Table 2: Results of experiments, problem phone, 15 independent random variables, 32768 scenarios

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>5</td>
<td>3.05</td>
<td>36.9</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2.33</td>
<td>36.9</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2.39</td>
<td>36.9</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>1</td>
<td>0.91</td>
<td>36.9</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>1</td>
<td>0.94</td>
<td>36.9</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>1</td>
<td>0.94</td>
<td>36.9</td>
</tr>
</tbody>
</table>

Table 3: Results of experiments, problem storm (truncated), 8 independent random variables, 390625 scenarios

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>23</td>
<td>1760.49</td>
<td>1.20304e+007</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>23</td>
<td>1720.98</td>
<td>1.20304e+007</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>23</td>
<td>1701.68</td>
<td>1.20304e+007</td>
</tr>
<tr>
<td>3</td>
<td>125</td>
<td>18</td>
<td>1355.51</td>
<td>1.20304e+007</td>
</tr>
<tr>
<td>4</td>
<td>625</td>
<td>18</td>
<td>1389.75</td>
<td>1.20304e+007</td>
</tr>
<tr>
<td>5</td>
<td>3125</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Out of memory on aggregation level 5 and more.
Table 4: Results of experiments, problem PGP2, 3 independent random variables, 576 scenarios

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>28</td>
<td>0.28</td>
<td>447.324</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>15</td>
<td>0.17</td>
<td>447.324</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
<td>12</td>
<td>0.22</td>
<td>447.324</td>
</tr>
<tr>
<td>3</td>
<td>576</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Out of memory on aggregation level 3.

Table 5: Results of experiments, problem 4node, 12 independent random variables, 4096 scenarios

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>198</td>
<td>59.87</td>
<td>446.856</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>142</td>
<td>40.64</td>
<td>446.856</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>80</td>
<td>20.30</td>
<td>446.856</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>60</td>
<td>15.06</td>
<td>446.856</td>
</tr>
</tbody>
</table>

Table 6: Results of experiments, problem 4node, 12 independent random variables, 32768 scenarios

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>236</td>
<td>436.86</td>
<td>446.856</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>131</td>
<td>223.50</td>
<td>446.856</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>86</td>
<td>123.47</td>
<td>446.856</td>
</tr>
</tbody>
</table>