Math. Prog. Comp. (2015) 7:471–499
DOI 10.1007/s12532-015-0088-0

FULL LENGTH PAPER

Computational study of decomposition algorithms for mean-risk stochastic linear programs

Tanisha G. Cotton[1] · **Lewis Ntaimo**[1]

Received: 3 May 2013 / Accepted: 30 June 2015 / Published online: 17 July 2015

Abstract Mean-risk stochastic programs include a risk measure in the objective to model risk averseness for many problems in science and engineering. This paper reports a computational study of mean-risk two-stage stochastic linear programs with recourse based on absolute semideviation (ASD) and quantile deviation (QDEV). The study was aimed at performing an empirical investigation of decomposition algorithms for stochastic programs with quantile and deviation mean-risk measures; analyzing how the instance solutions vary across different levels of risk; and understanding when it is appropriate to use a given mean-risk measure. Aggregated optimality cut and separate cut subgradient-based algorithms were implemented for each mean-risk model. Both types of algorithms show similar computational performance for ASD whereas the separate cut algorithm outperforms the aggregated cut algorithm for QDEV. The study provides several insights. For example, the results reveal that the risk-neutral approach is still appropriate for most of the standard stochastic programming test instances due to their uniform or normal-like marginal distributions. However, when the distributions are modified, the risk-neutral approach may no longer be appropriate and the risk-averse approach becomes necessary. The results also show that ASD is a more conservative mean-risk measure than QDEV.

Keywords Stochastic programming · Mean-risk objectives · Decomposition · Subgradient optimization

✉ Lewis Ntaimo
ntaimo@tamu.edu

Tanisha G. Cotton
tanisha.g.cotton@gmail.com

[1] Department of Industrial and Systems Engineering, Texas A&M University, 3131 TAMU, College Station, TX 77843, USA

Mathematics Subject Classification 90C15 · 90C25 · 90C06 · 90C90

1 Introduction

A risk-neutral two-stage stochastic program (SP) with recourse can be written as follows:

$$\text{SP1}: \underset{x \in X}{\text{Min}}\ \mathbb{E}[f(x, \tilde{\omega})] \tag{1}$$

where $x \in \mathbb{R}_+^{n_1}$ is a vector of decision variables, $X = \{Ax \geq b, x \geq 0\}$ is the set of feasible solutions, $A \in \mathbb{R}^{m_1 \times n_1}$ and $b \in \mathbb{R}^{m_1}$. The family of real random cost variables $\{f(x, \tilde{\omega})\}_{x \in X} \subseteq \mathcal{F}$ are defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. The symbol $\mathbb{E} : \mathcal{F} \mapsto \mathbb{R}$ denotes the expected value, where $\mathcal{F}$ is the space of all real random cost variables $F : \Omega \mapsto \mathbb{R}$ satisfying $\mathbb{E}[|F(\tilde{\omega})|] < \infty$. For a given $x \in X$ the real random cost variable $f(x, \tilde{\omega})$ is given by

$$f(x, \tilde{\omega}) = c^\top x + Q(x, \tilde{\omega}). \tag{2}$$

For an outcome (scenario) ω of $\tilde{\omega}$ the recourse function $Q(x, \omega)$ is given by

$$\begin{aligned} Q(x, \omega) = \text{Min}\ & q(\omega)^\top y \\ \text{s.t. } & Wy \geq r(\omega) - T(\omega)x \\ & y \geq 0, \end{aligned} \tag{3}$$

where $q(\omega) \in \mathbb{R}^{n_2}$ is the second stage cost vector, $y \in \mathbb{R}_+^{n_2}$ is the recourse decision, $W \in \mathbb{R}^{m_2 \times n_2}$ is the recourse matrix, $T(\omega) \in \mathbb{R}^{m_2 \times n_1}$ is the technology matrix, and $r(\omega) \in \mathbb{R}^{m_2}$ is the right hand side vector. A scenario defines the realization of the stochastic problem data $\{q(\omega), T(\omega), r(\omega)\}$. The recourse function $Q(x, \omega)$ is a value function of a linear program (LP) and is therefore a convex function of x. Since $\mathbb{E}$ is a linear operator, the expected recourse function $\mathbb{E}[Q(x, \omega)]$ is also convex. Consequently, SP1 is a convex program and is amenable to convex optimization methods. The challenge in solving large-scale instances of SP1 lies in evaluating the expectation in multidimensional space.

Modeling problems using only the expectation in the objective makes the formulation risk-neutral. To introduce risk, a risk measure $\mathbb{D} : \mathcal{F} \mapsto \mathbb{R}$ is added to (1) resulting in the following so-called *mean-risk* SP:

$$\text{SP2: } \underset{x \in X}{\text{Min}}\ \mathbb{E}[f(x, \tilde{\omega})] + \lambda \mathbb{D}[f(x, \tilde{\omega})], \tag{4}$$

where $\lambda > 0$ is a suitable weight factor that quantifies the tradeoff between expected cost and risk. The risk measure $\mathbb{D}$ should be convexity preserving to allow for SP2 to remain a convex problem and have access to convex optimization methods. Sufficient conditions for the mean-risk objective function to be convexity-preserving are that $\mathbb{D}$ must be convex, non-decreasing, and positively homogenous [1].

In this paper we consider problem SP2 (4) under the following assumptions:

(A1) The random variable $\tilde{\omega}$ is discrete with finitely many scenarios $\omega \in \Omega$, each with probability of occurrence $p(\omega)$.
(A2) The first-stage feasible set is nonempty, that is, $X \neq \emptyset$.
(A3) For all $x \in X$ and $\omega \in \Omega$, $f(x, \omega) < \infty$.

Assumption (A1) is needed to make the problem tractable while assumptions (A2) and (A3) together guarantee the existence of an optimal solution. Assumption (A3) is the *relatively complete recourse* assumption, and if it does not hold, Benders feasibility cuts should be generated and added to the master problem for every $x \in X$ that leads to infeasibility in the second-stage.

Research on SP2 in the literature include characterization of mean-risk measures for $\mathbb{D}$ and algorithms for each specification of $\mathbb{D}$. However, computational study of the empirical behavior of the algorithms is fairly limited. Most work simply reports some preliminary computational results, usually based on a single application. For a recent survey on mean-risk SP, we refer the interested reader to [10]. We should also point out that algorithms for SP2 with integer restrictions on decision variables is an ongoing active area of research [15,23–25].

Risk measures for SP2 are typically classified into two categories: *deviation* measures and *quantile* measures. The former is based on the deviation of a random variable from the mean or a preselected target, while the latter is formulated using a quantile of the probability distribution. The commonly used mean-risk measures for SP2 include deviation measures, *expected excess* [15] and *absolute semideviation* [19]; and quantile measures, *excess probability* [24], *quantile deviation* [19], and *conditional value-at-risk* (CVaR) [20]. All these risk measures fall within the class of so-called coherent risk measures [2]. A minimax model that is equivalent to a mean-risk model with quantile deviation risk measure is considered in [27]. They use the l_∞-trust-region based decomposition algorithm of [11] for solving the model. The algorithm is implemented in ANSI C with GNU LP Kit and is used to solve three 1000 sampled scenario instances of the standard SP test instances *LandS*, *gbd*, *20term* and *storm*. Their results show increasing the perturbation of the reference distribution in the minimax model is equivalent to increasing the weight term of the mean-risk model which leads to higher expected cost solutions. In addition, it was observed that some dispersion statistics decrease, indicating a reduction of risk. The tested problem instances were found to be very robust with respect to changes in the reference distribution. This was observed by noticing that large changes in the reference distribution did not significantly change the optimal objective value. They also observed large variability in total CPU time for different model parameter combinations. This was attributed to the regularized nature of the mean-risk (or minimax) objective function.

In [9] the deviation measures of central deviation, semideviation, and expected excess are tested on an LP relaxation of a mixed-integer scheduling problem in chemical production. A variant of the L-shaped algorithm with a regularization term in the objective is used and tested on instances with several scenarios. Results for central deviation and semideviation, show that the regularized algorithm reduces number of cuts substantially, but results in similar iterations and computational times when compared with the non-regularized algorithm. The algorithm results in time savings in most runs and a reduction in the number of iterations and cuts in all runs involving expected excess.

Subgradient optimization based decomposition algorithms for SP2 using parametric cutting planes for the absolute semideviation and quantile deviation risk measures are derived in [1]. Using test instances in sizes ranging from 25 to 500, equally likely scenarios were sampled for the standard SP test problems *LandS*, *gbd*, *20term*, *storm* and *ssn* studied in [12]. Results show that the quantile deviation model offers broader flexibility in the mean-risk tradeoff than the absolute semideviation model. Furthermore, in each case, the parametric strategy is substantially more efficient than resolving the problem from scratch for different values of the risk parameter. The results are in the form of graphs showing the mean-risk frontier and no other computational results are reported.

The work in [17] considers an extended and multicut Benders' decomposition method to solve the problem of optimizing a portfolio with finite assets. This is the same problem addressed by [16]. The two-stage mean-risk formulations, of semideviation and mean-weighted deviation from a quantile, are solved with a multicut and extended Benders' decomposition method. For each mean-risk model the algorithms performance is compared with solving the large-scale deterministic equivalent problem (DEP) for the portfolio optimization problem using the simplex method. The computational results for several instances of different sizes show that the simplex method applied to the DEP outperforms both cutting plane methods for both dispersion statistics in terms of computation time. However, the extended Benders' Method requires less memory as problem size increases for the semideviation model. Also, the extended Benders' method uses less memory than the multicut method.

To the best of our knowledge, there is no computational study to benchmark algorithms for SP2. This work aims to do just that and extends preliminary computations reported in [1]. Thus we consider different specifications of $\mathbb{D}$ and corresponding DEP formulations for SP2, and algorithms for solving them. We present detailed algorithms for SP2 and report computational results regarding the algorithms performance on large-scale standard test instances and draw insights and conclusions. The *contribution* of this work to the literature on SP is a computational study benchmarking and characterizing the performance of subgradient optimization based algorithms for SP2 on the standard instances. The study provides several insights into the decisions made using the mean-risk approach versus the risk-neutral approach for different applications. For example, for many standard test instances with marginal uniform distributions, the risk-neutral approach suffices in making good decisions under uncertainty. However, if the marginal distributions are not uniform, the mean-risk-averse approach provides decisions in the face of uncertainty that depend on the decision-maker's level of risk.

The rest of this paper is organized as follows: We give definitions of the mean-risk objectives and then state the DEP formulations of SP2 in Sect. 2. We give formal descriptions of our decomposition algorithms in Sect. 3 and report computational results in Sect. 4. We end the paper with concluding remarks in Sect. 5.

2 Mean-risk objectives and deterministic equivalent formulations

Let us now turn to the specification of $\mathbb{D}$ using coherent risk measures from the literature. The risk measures should be convexity preserving in order for SP2 to remain convex and have access to convex optimization algorithms. In our specifications of $\mathbb{D}$ that follow, $\max(a, b)$ and $\min(a, b)$ denote the maximum and minimum operators, respectively, applied to $a \in \mathbb{R}$ and $b \in \mathbb{R}$. We first state the definitions of the risk measures in the next subsection and then give the deterministic equivalent formulations in the subsequent subsection.

2.1 Risk measures and mean-risk objectives

The deviation measures *expected excess* (EE) and *absolute semideviation* (ASD) can be defined as follows:

(i) Expected Excess: Given a target $\eta \in \mathbb{R}$, expected excess [15] is given as

$$\phi_{EE_\eta}(x) = \mathbb{E}[\max\{f(x, \tilde{\omega}) - \eta, 0\}].$$

It reflects the expected value of the excess over the target $\eta \in \mathbb{R}$. Substituting $\mathbb{D} := \phi_{EE_\eta}$ in (4) we obtain SP2 with expected excess as follows:

$$\underset{x \in X}{\text{Min}} \ \mathbb{E}[f(x, \tilde{\omega})] + \lambda \phi_{EE_\eta}(x). \tag{5}$$

(ii) Absolute semideviation: This is same as expected excess but with the target value replaced by the mean value $\mathbb{E}[f(x, \tilde{\omega})]$ [19] and is given as follows:

$$\phi_{ASD}(x) = \mathbb{E}[\max\{f(x, \tilde{\omega}) - \mathbb{E}[f(x, \tilde{\omega})], 0\}].$$

It reflects the expected value of the excess over the mean value. By setting $\mathbb{D} := \phi_{ASD}$ in (4) we obtain the following SP2 with absolute semideviation:

$$\underset{x \in X}{\text{Min}} \ \mathbb{E}[f(x, \tilde{\omega})] + \lambda \phi_{ASD}(x). \tag{6}$$

The quantile mean-risk measures *excess probability* (EP) and *quantile deviation* (QDEV) can be defined as follows:

(iii) Excess probability: Given a target level $\eta \in \mathbb{R}$, excess probability [24] is the probability of exceeding η and is given by

$$\phi_{EP_\eta}(x) := \mathbb{P}[\omega \in \Omega : \ f(x, \omega) > \eta].$$

Substituting $\mathbb{D} := \phi_{EP_\eta}$ in (4) we obtain the following SP2 with excess probability:

$$\underset{x \in X}{\text{Min}} \ \mathbb{E}[f(x, \tilde{\omega})] + \lambda \phi_{EP_\eta}(x). \tag{7}$$

(iv) Quantile deviation: Given $\alpha \in (0, 1)$, quantile deviation [19] is defined as follows:

$$\phi_{QDEV_\alpha}(x) = \mathbb{E}[(1-\alpha)\max(\kappa_\alpha f(x,\tilde{\omega}) - f(x,\tilde{\omega}), 0) \\ + \alpha \max(f(x,\tilde{\omega}) - \kappa_\alpha f(x,\tilde{\omega}), 0)],$$

where κ_α is the α-quantile of the distribution of $f(x,\tilde{\omega})$. Thus SP2 with $\mathbb{D} := \phi_{QDEV_\alpha}$ can now be given as follows:

$$\underset{x\in X}{\text{Min}}\ \mathbb{E}[f(x,\tilde{\omega})] + \lambda\phi_{QDEV_\alpha}(x). \tag{8}$$

Observe that QDEV is a two-sided weighted mean deviation from a quantile. In [21] ϕ_{QDEV_α} is shown to be equivalent to

$$\phi_{\varepsilon_1,\varepsilon_2}(x) = \underset{\eta\in\mathbb{R}}{\text{Min}}\ \mathbb{E}[\varepsilon_1 \max(\eta - f(x,\tilde{\omega}), 0) + \varepsilon_2 \max(f(x,\tilde{\omega}) - \eta, 0)],$$

where $\alpha = \varepsilon_2/(\varepsilon_1 + \varepsilon_2)$ and $\varepsilon_1, \varepsilon_2 > 0$.

Further details on mathematical structures and properties of mean-risk measures are given in [1].

2.2 Deterministic equivalent formulations

By applying assumption (A1), we can write deterministic equivalent formulations of SP with mean-risk measures defined in the previous subsection. Due to problem size, which grows with the number $|\Omega|$ of scenarios, the problems become too large for direct solvers. This motivates the computational study of decomposition methods to solve these problems. Next we give deterministic equivalent formulations of the ASD deviation measure and QDEV quantile mean-risk measure, which are the focus of this work.

(i) Absolute semideviation

Proposition 2.1 *Given* $\lambda \in [0, 1]$, *then problem* (6) *is equivalent to the following formulation* [1,8,22]*:*

$$\textit{ASD: Min}\quad (1-\lambda)c^\top x + (1-\lambda)\sum_{\omega\in\Omega} p(\omega)q(\omega)^\top y(\omega) + \lambda \sum_{\omega\in\Omega} p(\omega)\nu(\omega) \tag{9a}$$

$$\begin{aligned} \textit{s.t.}\quad & T(\omega)x + Wy(\omega) \geq r(\omega), \quad \forall\omega\in\Omega \\ & -c^\top x - q(\omega)^\top y(\omega) + \nu(\omega) \geq 0, \quad \forall\omega\in\Omega \\ & -c^\top x - \sum_{\omega\in\Omega} p(\omega)q(\omega)^\top y(\omega) + \nu(\omega) \geq 0, \quad \forall\omega\in\Omega \qquad (9b) \\ & x\in X, \quad y(\omega)\in\mathbb{R}_+^{n_2}, \ \nu(\omega)\in\mathbb{R}, \quad \forall\omega\in\Omega. \end{aligned}$$

Unlike all the other forms of SP2, problem ASD does *not* have a block angular structure. This is due to the ‘complicating’ or ‘linking’ constraints (9b). Observe that these constraints link *all* the scenarios and therefore, standard stochastic linear programming methods such as the L-shaped method *cannot* be used to solve ASD. Thus we explore subgradient optimization based algorithms to solve ASD.

(ii) Quantile deviation

Proposition 2.2 *Given* $\lambda \in [0, 1/\varepsilon_1]$, *problem* (8) *is equivalent to the following LP* [1]*:*

$$\begin{aligned} \textit{QDEV: Min} \quad & (1-\lambda\varepsilon_1)c^\top x + \lambda\varepsilon_1\eta + (1-\lambda\varepsilon_1)\sum_{\omega\in\Omega} p(\omega)q(\omega)^\top y(\omega) \\ & + \lambda(\varepsilon_1+\varepsilon_2)\sum_{\omega\in\Omega} p(\omega)\nu(\omega) \qquad (10) \\ \textit{s.t.} \quad & T(\omega)x + Wy(\omega) \geq r(\omega), \quad \forall\omega\in\Omega \\ & -c^\top x - q(\omega)^\top y(\omega) + \eta + \nu(\omega) \geq 0, \quad \forall\omega\in\Omega \\ & x\in X,\ \eta\in\mathbb{R},\ y(\omega)\in\mathbb{R}^{n_2}_+, \quad \nu(\omega)\in\mathbb{R}, \forall\omega\in\Omega. \end{aligned}$$

Problem QDEV is an LP, has a block angular structure and is amenable to standard decomposition methods for two-stage stochastic linear programs. In this case x and η are the first-stage variables, and $y(\omega)$ and $\nu(\omega)$ are the second-stage variables.

3 Decomposition algorithms

The L-shaped algorithm is the method of choice for solving block angular structured stochastic LPs such as EE and QDEV. So we shall focus our attention on solving ASD. Since $f(., \omega)$ is convex for almost every $\omega \in \Omega$ and the mean-risk function $\phi_{ASD}(.)$ in (6) is convexity preserving, problem ASD involves minimizing a convex (often non-smooth) objective function $\mathbb{E}[f(., \omega)]+\lambda\phi_{ASD}(x)$. Therefore, subgradient optimization based methods are suitable for ASD. To solve ASD, we consider the subgradient optimization based method by [1] and a variant of this method. The first method places two separate optimality cuts in the master program at each iteration of the algorithm, one for the expectation term and another for the deviation term. The variant method places only one ‘aggregated’ optimality cut in the master program at each iteration of the algorithm. We refer to the algorithm by [1] as the ‘separate cut’ (SEP) algorithm, and refer to the single cut algorithm as the ‘aggregate cut’ (AGG) algorithm.

Given a feasible solution x, these subgradient optimization based algorithms require computing a subgradient of the objective function. So we adopt a Benders-type decomposition setting and decompose problem ASD stage-wise into a master program and subproblems for each $\omega \in \Omega$. We then derive the subgradients needed in generating the optimality cuts left hand side coefficients and right hand side values. Let us denote the algorithm iteration index by k. Next we give a derivation of the AGG algorithm in Subsect. 3.1 and address the SEP algorithm in Subsect. 3.2. For completeness, we state our implementation of the L-shaped algorithm for solving QDEV in Subsect. 3.3.

3.1 Aggregated cut subgradient-based algorithm

Recall that problem ASD does not have the block angular structure due to the scenario linking constraints (9b). Therefore, we first decompose ASD into a master problem and subproblem, and then derive subgradients to enable us to compute a single (aggregated) optimality cut at each iteration of the algorithm. Let the ASD master problem at iteration k be given as follows:

$$
\begin{aligned}
\ell_k = &\text{Min } (1-\lambda)c^\top x + \gamma \\
&\text{s.t. } Ax \geq b \\
&\quad \beta_t^\top x + \gamma \geq \beta_t^0, \quad t = 1, \ldots, k \\
&\quad x \geq 0.
\end{aligned} \tag{11}
$$

In formulation (11), the free variable γ is the optimality cut variable and it represents a lower bound approximation of $(1-\lambda)\sum_{\omega\in\Omega} p(\omega)q(\omega)^\top y(\omega) + \lambda \sum_{\omega\in\Omega} p(\omega)\nu(\omega)$. The constraints represent the optimality cuts with left hand side coefficients at iteration k denoted by β_k and the corresponding right hand side by β_k^0.

Given a master problem solution (x^k, γ^k) at iteration k, the subproblem (ignoring the scenario linking constraints) for $\omega \in \Omega$ is given by

$$
\begin{aligned}
Q(x^k, \omega) = &\text{Min } q(\omega)^\top y(\omega) \\
&\text{s.t. } Wy(\omega) \geq r(\omega) - T(\omega)x^k \\
&\quad y(\omega) \geq 0.
\end{aligned} \tag{12}
$$

The set of dual feasible solutions to (12) is given by $\Pi(\omega) = \{\pi(\omega) \in \mathbb{R}_+^{m_2} \mid W^\top \pi(\omega) \leq q(\omega)\}$. Therefore,

$$
\begin{aligned}
f(x,\omega) &= c^\top x + Q(x,\omega), \quad \forall x \in X \\
&= c^\top x + \underset{\pi(\omega)\in\Pi(\omega)}{\text{Max}} \left\{\pi(\omega)^\top [r(\omega) - T(\omega)x]\right\}, \quad \forall x \in X \\
&\geq c^\top x + \pi^k(\omega)^\top [r(\omega) - T(\omega)x], \quad \forall x \in X \\
&= c^\top x + \pi^k(\omega)^\top [r(\omega) - T(\omega)x] + f(x^k,\omega) - c^\top x^k \\
&\quad - \pi^k(\omega)^\top [r(\omega) - T(\omega)x^k], \quad \forall x \in X \\
&= f(x^k,\omega) + [c - T(\omega)^\top \pi^k(\omega)]^\top (x - x^k), \quad \forall x \in X.
\end{aligned}
$$

The fourth equation follows from strong duality and the last equation shows that $c - T(\omega)^\top \pi^k(\omega)$ is a *subgradient* of $f(., \omega)$ at x^k.

Given a solution x^k and corresponding optimal value $Q(x^k, \omega)$ to problem (12) at iteration k for all $\omega \in \Omega$, the absolute semideviation term is given by

$$\nu^k(\omega) = c^\top x^k + \max\left\{Q(x^k, \omega), \sum_{\omega' \in \Omega} p(\omega')Q(x^k, \omega')\right\}. \quad (13)$$

Because of the 'max' term on the right hand side of (13), we have two cases to consider, that is, either $\nu^k(\omega) = c^\top x^k + Q(x^k, \omega)$ or $\nu^k(\omega) = c^\top x^k + \sum_{\omega' \in \Omega} p(\omega')Q(x^k, \omega')$.

Case 1: $\nu^k(\omega) = c^\top x^k + Q(x^k, \omega)$.

For this case we use the subgradient $c - T(\omega)^\top \pi^k(\omega)$ and define $\beta_k(\omega)$ and $\beta_k^0(\omega)$ to be used in the approximation of γ in the master program as follows:

$$\beta_k(\omega) = (1 - \lambda)T(\omega)^\top \pi^k(\omega) + \lambda[T(\omega)^\top \pi^k(\omega) - c]$$

and

$$\beta_k^0(\omega) = (1 - \lambda)\pi^k(\omega)^\top r(\omega) + \lambda \pi^k(\omega)^\top r(\omega).$$

Case 2: $\nu^k(\omega) = c^\top x^k + \sum_{\omega' \in \Omega} p(\omega')Q(x^k, \omega')$.
In this case we use the expected subgradient $c - \sum_{\omega' \in \Omega} p(\omega')T(\omega')^\top \pi^k(\omega')$ in computing β_k and $\beta_k^0(\omega)$ as follows:

$$\beta_k(\omega) = (1 - \lambda)T(\omega)^\top \pi^k(\omega) + \lambda\left[\sum_{\omega' \in \Omega} p(\omega')T(\omega')^\top \pi^k(\omega') - c\right]$$

and

$$\beta_k^0(\omega) = (1 - \lambda)\pi^k(\omega)^\top r(\omega) + \lambda \sum_{\omega' \in \Omega} p(\omega')\pi^k(\omega')^\top r(\omega').$$

Consequently, the optimality cut coefficients β_k and right hand side β_k^0 in the master problem (11) at iteration k can be calculated based on selecting the appropriate $\beta_k(\omega)$ and $\beta_k^0(\omega)$ for each case for a given scenario $\omega \in \Omega$ as follows:

$$\beta_k = \sum_{\omega \in \Omega} p(\omega)\beta_k(\omega)$$

and

$$\beta_k^0 = \sum_{\omega \in \Omega} p(\omega)\beta_k^0(\omega).$$

Hence, the optimality cut to add to the master program is $\beta_k^\top x + \gamma \geq \beta_k^0$. We are now in a position to formally state the AGG algorithm for ASD.

ASD-AGG Algorithm

Step 0. Initialization.
choose $\lambda \in [0, 1]$ and set $k \leftarrow 1$, $\ell_1 \leftarrow -\infty$, $u_0 \leftarrow \infty$, and choose $x^1 \in X$.

Step 1. Solve Subproblem LPs.
initialize $\beta_k \leftarrow \mathbf{0}$, $\beta_k^0 \leftarrow 0$, $\bar{Q}^k \leftarrow 0$ and $\bar{\nu}^k \leftarrow 0$.

for each $\omega \in \Omega$ solve subproblem (12)
  get and store optimal value $Q(x^k, \omega)$
  get dual solution $\pi^k(\omega)$
  compute $\bar{Q}^k \leftarrow \bar{Q}^k + p(\omega) Q(x^k, \omega)$
end for.

for each $\omega \in \Omega$
  if $Q(x^k, \omega) \geq \bar{Q}^k$
    compute $\bar{\nu}^k \leftarrow \bar{\nu}^k + p(\omega) Q(x^k, \omega)$
    compute $\beta_k \leftarrow \beta_k + p(\omega)[(1-\lambda)T(\omega)^\top \pi^k(\omega) + \lambda(T(\omega)^\top \pi^k(\omega) - c^\top)]$
    compute $\beta_k^0 \leftarrow \beta_k^0 + p(\omega)[(1-\lambda)\pi^k(\omega)^\top r(\omega) + \lambda \pi^k(\omega)^\top r(\omega)]$
  else
    compute $\bar{\nu}^k \leftarrow \bar{\nu}^k + p(\omega) \bar{Q}^k$
    $\beta_k \leftarrow \beta_k + p(\omega)[(1-\lambda)T(\omega)^\top \pi^k(\omega) + \lambda(\sum_{\omega' \in \Omega} p(\omega')T(\omega')^\top \pi^k(\omega') - c)]$
    $\beta_k^0 \leftarrow \beta_k^0 + p(\omega)[(1-\lambda)\pi^k(\omega)^\top r(\omega) + \lambda \sum_{\omega' \in \Omega} p(\omega')\pi^k(\omega')^\top r(\omega')]$
  end if.
end for.

compute $u_k \leftarrow (1-\lambda)(c^\top x^k + \bar{Q}^k) + \lambda(c^\top x^k + \bar{\nu}^k)$.

set $u_k \leftarrow \min\{u_k,\ u_{k-1}\}$

if u_k is updated
  set incumbent solution $x_\epsilon^* \leftarrow x^k$
end if

if $u_k - \ell_k \leq \epsilon |u_k|$
  stop and declare x_ϵ^* ϵ-optimal
end if.

Step 2. Update and Solve the Master Problem.
add optimality cut $\beta_k^\top x + \gamma \geq \beta_k^0$ to master problem (11).

solve master problem to get optimal value ℓ_{k+1}
and solution (x^{k+1}, γ^{k+1}).

set $\ell_{k+1} \leftarrow \max\{\ell_{k+1},\ \ell_k\}$.

set $k \leftarrow k + 1$ and repeat from step 1.

3.2 Separate cut subgradient-based algorithm

The derivation of the AGG algorithm shows that the components of the subgradients on the objective function of ASD can be generated separately for the expectation and absolute semideviation terms as in [1]. To accomplish this, we need to redefine the master program. We will now have the optimality cut variable γ, the left hand side coefficients β_k, and the right hand side β_k^0 be used for the expectation term only. The lower bound approximation on the deviation term will be denoted by ζ. The optimality cut left hand side coefficients will be denoted by σ^k and the right hand side by σ_k^0. Now the ASD master program at iteration k can be given as follows:

$$
\begin{aligned}
\ell_k = & \text{Min } (1-\lambda)c^\top x + (1-\lambda)\gamma + \lambda\zeta \\
& \text{s.t. } Ax \geq b \\
& \quad \beta_\tau^\top x + \gamma \geq \beta_\tau^0, \quad \tau = 1, \ldots, t \\
& \quad \sigma_\tau^\top x + \zeta \geq \sigma_\tau^0, \quad \tau = 1, \ldots, t \\
& \quad x \geq 0.
\end{aligned}
\tag{14}
$$

In formulation (14) the second and third constraints represent the (separate) optimality cuts on the expected and deviation terms, respectively. Let (x^k, γ^k, ζ^k) solve problem (14). Then for each $\omega \in \Omega$, the scenario *subproblem* will remain as given in (12). Following the derivation in the previous subsection, let for all $\omega \in \Omega$, $y^k(\omega)$ solve problem (12) and let $\pi^k(\omega)$ be the corresponding optimal dual multipliers associated with the subproblem constraints. The optimality cut for the expectation term can be derived as follows: Recall that $c - T(\omega)^\top \pi^k(\omega)$ is a subgradient of $f(., \omega)$. Therefore,

$$\beta_k(\omega) = T(\omega)^\top \pi^k(\omega)$$

and

$$\beta_k^0(\omega) = \pi^k(\omega)^\top r(\omega).$$

Thus the optimality cut coefficients β_t and right hand side β_k^0 for the master problem (14) can be calculated as follows:

$$\beta_t = \sum_{\omega \in \Omega} p(\omega)\beta_k(\omega)$$

and

$$\beta_k^0 = \sum_{\omega \in \Omega} p(\omega)\beta_k^0(\omega).$$

The optimality cut to add to the master program is then $\beta_k^\top x + \gamma \geq \beta_k^0$.

Let us now turn to the optimality cut for the absolute semideviation term. We shall again have two cases to consider due to equation (13):

Case 1: $\nu^k(\omega) = c^\top x^k + Q(x^k, \omega)$.

In this case we use the subgradient $c - T(\omega)^\top \pi^k(\omega)$ for $\omega \in \Omega$. Therefore, we can define $\sigma_k(\omega)$ and $\sigma_k^0(\omega)$ as follows:

$$\sigma_k(\omega) = T(\omega)^\top \pi^k(\omega) - c$$

and

$$\sigma_k^0(\omega) = \pi^k(\omega)^\top r(\omega).$$

Case 2: $\nu^k(\omega) = c^\top x^k + \sum_{\omega' \in \Omega} p(\omega') Q(x^k, \omega')$.
In this case we use the expected subgradient $c - \sum_{\omega' \in \Omega} p(\omega') T(\omega')^\top \pi^k(\omega')$ for $\omega \in \Omega$. Therefore define $\sigma_k(\omega)$ and $\sigma_k^0(\omega)$ as follows:

$$\sigma_k(\omega) = \sum_{\omega' \in \Omega} p(\omega') T(\omega')^\top \pi^k(\omega') - c^\top$$

and

$$\sigma_k^0(\omega) = \lambda \sum_{\omega' \in \Omega} p(\omega') \pi^k(\omega')^\top r(\omega').$$

Consequently, the optimality cut coefficients σ_k and right hand side σ_k^0 in (14) at iteration k can be calculated as follows, based on selecting the appropriate case for each scenario $\omega \in \Omega$::

$$\sigma_k = \sum_{\omega \in \Omega} p(\omega) \sigma_k(\omega)$$

and

$$\sigma_k^0 = \sum_{\omega \in \Omega} p(\omega) \sigma_k^0(\omega).$$

The optimality cut to add to the master program is then $\sigma_k^\top x + \zeta \geq \sigma_k^0$. We can now state the SEP algorithm for ASD as follows:

ASD-SEP Algorithm

Step 0. Initialization.

choose $\lambda \in [0, 1]$ and set $k \leftarrow 1$, $\ell_1 \leftarrow -\infty$, $u_0 \leftarrow \infty$, and $x^1 \in X$.

Step 1. Solve Subproblem LPs.

initialize $\beta_k \leftarrow \mathbf{0}$, $\beta_k^0 \leftarrow 0$, $\sigma_k \leftarrow \mathbf{0}$, $\sigma_k^0 \leftarrow 0$, $\bar{Q}^k \leftarrow 0$ and $\bar{\nu}^k \leftarrow 0$.

for each $\omega \in \Omega$ solve subproblem (12)

get and store optimal value $Q(x^k, \omega)$

get dual solution $\pi^k(\omega)$

compute $\bar{Q}^k \leftarrow \bar{Q}^k + p(\omega)Q(x^k, \omega)$

end for.

for each $\omega \in \Omega$

compute $\beta_k \leftarrow \beta_k + p(\omega)[T(\omega)^\top \pi^k(\omega)]$

compute $\beta_k^0 \leftarrow \beta_k^0 + p(\omega)[\pi^k(\omega)^\top r(\omega)]$

if $Q(x^k, \omega) \geq \bar{Q}^k$

compute $\bar{\nu}^k \leftarrow \bar{\nu}^k + p(\omega)Q(x^k, \omega)$

compute $\sigma_k \leftarrow \sigma_k + p(\omega)[T(\omega)^\top \pi^k(\omega) - c]$

compute $\sigma_k^0 \leftarrow \sigma_k^0 + p(\omega)[\pi^k(\omega)^\top r(\omega)]$

else

compute $\bar{\nu}^k \leftarrow \bar{\nu}^k + p(\omega)\bar{Q}^k$

$\sigma_k \leftarrow \sigma_k + p(\omega)[\sum_{\omega' \in \Omega} p(\omega')T(\omega')^\top \pi^k(\omega') - c]$

$\sigma_k^0 \leftarrow \sigma_k^0 + p(\omega)[\sum_{\omega' \in \Omega} p(\omega')\pi^k(\omega')^\top r(\omega')]$

end if.

end for.

compute $u_k \leftarrow (1 - \lambda)(c^\top x^k + \bar{Q}^k) + \lambda(c^\top x^k + \bar{\nu}^k)$.

set $u_k \leftarrow \min\{u_k,\ u_{k-1}\}$.

if u_k is updated set incumbent solution $x_\epsilon^* \leftarrow x^k$.

if $u_k - \ell_k \leq \epsilon|u_k|$, stop and declare x_ϵ^* ϵ-optimal.

Step 2. Update and Solve the Master Problem.

add optimality cut $\beta_k^\top x + \gamma \geq \beta_k^0$ to master problem (14).

add optimality cut $\sigma_k^\top x + \zeta \geq \sigma_k^0$ to master problem (14).

solve master problem to get optimal value ℓ_{k+1}
and solution $(x^{k+1}, \gamma^{k+1}, \zeta^{k+1})$.

set $\ell_{k+1} \leftarrow \max\{\ell_{k+1},\ \ell_k\}$.

set $k \leftarrow k + 1$ and repeat from step 1.

3.3 L-shaped algorithm for QDEV

Let us now turn to solving QDEV using the L-shaped method. As with ASD, we consider two variants of the L-shaped method, a single cut or aggregated (AGG) implementation, and a separate cut (SEP) method. We begin with the AGG L-shaped method. The master problem for QDEV at iteration k can be given as follows:

$$\begin{aligned} \ell^k = &\text{Min } (1-\lambda\varepsilon_1)c^\top x + \lambda\varepsilon_1\eta + \gamma \\ &\text{s.t. } Ax \geq b \\ &\quad \beta_t^\top x + \beta_t^1\eta + \gamma \geq \beta_t^0, \quad t = 1, \ldots, k \\ &\quad x \geq 0, \quad \delta_l \leq \eta \leq \delta_h. \end{aligned} \tag{15}$$

The parameters δ_l and δ_h are lower and upper bounds on η and are needed to speedy up computation. For each master problem solution (x^k, η^k, γ^k) at iteration k, the subproblem for $\omega \in \Omega$ is given by

$$\begin{aligned} Q(x^k, \eta^k, \omega) = &\text{Min } (1-\lambda\varepsilon_1)q(\omega)^\top y \; + \lambda(\varepsilon_1+\varepsilon_2)\nu \\ &\text{s.t. } Wy \; \geq r(\omega) - T(\omega)x^k \\ &\quad - q(\omega)^\top y + \nu \geq -\eta^k + c^\top x^k \\ &\quad y \geq 0. \end{aligned} \tag{16}$$

Let the dual solution to problem (16) corresponding to the two constraints be given by $\pi^k(\omega)$ and $\pi_0^k(\omega)$, respectively. Then

$$\begin{aligned} \beta_k &= \sum_{\omega\in\Omega} p(\omega)[T(\omega)^\top \pi^k(\omega) + \pi_0^k(\omega)c], \\ \beta_k^1 &= -\sum_{\omega\in\Omega} p(\omega)\pi_0^k(\omega) \end{aligned}$$

and

$$\beta_k^0 = \sum_{\omega\in\Omega} p(\omega)\pi^k(\omega)^\top r(\omega).$$

We can now formally state the AGG L-shaped algorithm for QDEV as follows:

QDEV-AGG Algorithm

Step 0. Initialization.

set $k \leftarrow 1$, $\ell_1 \leftarrow -\infty$, $u_0 \leftarrow \infty$, $\epsilon > 0$ and select appropriate values for δ_l and δ_h

choose $\lambda \in [0, \frac{1}{\varepsilon_1}]$, $\alpha \in (0, 1)$, and $\varepsilon_1, \varepsilon_2 > 0$ such that $\alpha = \varepsilon_2/(\varepsilon_1 + \varepsilon_2)$, and $(x^1, \eta^1) \in X$

Step 1. Solve Subproblem LPs.

initialize $\beta_k \leftarrow \mathbf{0}$, $\beta_k^1 \leftarrow 0$ and $\beta_k^0 \leftarrow 0$

for each $\omega \in \Omega$ solve subproblem (16)

get $Q(x^k, \eta^k, \omega)$ and dual solution $(\pi^k(\omega), \pi_0^k(\omega))$

compute $\beta_k \leftarrow \beta_k + p(\omega)[T(\omega)^\top \pi^k(\omega) + \pi_0^k(\omega)c]$

compute $\beta_k^1 \leftarrow \beta_k^1 - p(\omega)\pi_0^k(\omega)$

compute $\beta_k^0 \leftarrow \beta_k^0 + p(\omega)[\pi^k(\omega)^\top r(\omega)]$

set $u_k \leftarrow \min\{(1-\lambda\varepsilon_1)c^\top x^k + \lambda\varepsilon_1\eta^k + \sum_{\omega\in\Omega} p(\omega)Q(x^k, \eta^k, \omega),\ u_{k-1}\}$

if u_k is updated set incumbent solution $(x_\epsilon^*, \eta_\epsilon^*) \leftarrow (x^k, \eta^k)$

if $u_k - \ell_k \leq \epsilon|u_k|$, stop and declare $(x_\epsilon^*, \eta_\epsilon^*)$ ϵ-optimal.

Step 2. Update and Solve the Master Problem.

add optimality cut $\beta_k^\top x + \beta_k^1 \eta + \gamma \geq \beta_k^0$ to master problem (15)

solve master problem to get optimal value ℓ_{k+1}
and solution $(x^{k+1}, \eta^{k+1}, \gamma^{k+1})$

set $\max\{\ell_{k+1},\ \ell_k\}$.

Let us now turn to the SEP L-shaped method for QDEV. In this case the master problem at iteration k takes the following form:

$$
\begin{aligned}
\ell^k = \text{Min}\ & (1 - \lambda\varepsilon_1)c^\top x + \lambda\varepsilon_1\eta + (1 - \lambda\varepsilon_1)\gamma + \lambda(\varepsilon_1 + \varepsilon_2)\zeta \\
\text{s.t.}\ & Ax \geq b \\
& \beta_t^\top x + \gamma \geq \beta_t^0, \quad t = 1, \ldots, k \\
& \sigma_t^\top x + \sigma_t^1 \eta + \zeta \geq \sigma_t^0, \quad t = 1, \ldots, k \\
& x \geq 0, \quad \delta_l \leq \eta \leq \delta_h.
\end{aligned}
\tag{17}
$$

Let the master problem solution at iteration k be $(x^k, \eta^k, \gamma^k, \zeta^k)$. Then the subproblem for $\omega \in \Omega$ is given by

$$\begin{aligned} Q(x^k, \omega) = \text{Min} \ & q(\omega)^\top y \\ \text{s.t.} \ W y \ & \geq r(\omega) - T(\omega) x^k \\ & y \geq 0. \end{aligned} \tag{18}$$

Let the dual solution to subproblem (18) be $\pi^k(\omega)$. Then the optimality cut coefficients on the expectation term can be calculated as follows:

$$\beta_k = \sum_{\omega \in \Omega} p(\omega) T(\omega)^\top \pi^k(\omega)$$

and

$$\beta_k^0 = \sum_{\omega \in \Omega} p(\omega) \pi^k(\omega)^\top r(\omega).$$

Recall that given a solution (x^k, η^k) to master problem (17) and solution $y^k(\omega)$ to subproblem (18) for $\omega \in \Omega$, the quantile deviation term takes the value

$$\nu^k(\omega) = \max\{c^\top x^k + Q(x^k, \omega) - \eta^k, 0\}.$$

Therefore, the optimality cut coefficients for the quantile deviation term for $\omega \in \Omega$ can be calculated as follows: Let $\iota(\omega) = 1$ if $c^\top x^k + Q(x^k, \omega) > \eta^k$, and $\iota(\omega) = 0$ otherwise. Then

$$\begin{aligned} \sigma_k &= \sum_{\omega \in \Omega} \iota(\omega) p(\omega) T(\omega)^\top \pi^k(\omega), \\ \sigma_k^1 &= \sum_{\omega \in \Omega} \iota(\omega) p(\omega) \end{aligned}$$

and

$$\sigma_k^0 = \sum_{\omega \in \Omega} \iota(\omega) p(\omega) \pi^k(\omega)^\top r(\omega).$$

We can now formally state the L-shaped algorithm with separate cuts for QDEV as follows:

QDEV-SEP Algorithm

Step 0. Initialization.

set $k \leftarrow 1, \ell_1 \leftarrow -\infty, u_0 \leftarrow \infty, \epsilon > 0$ and select appropriate values for δ_l and δ_h

choose $\lambda \in [0, \frac{1}{\varepsilon_1}], \alpha \in (0, 1)$, and $\varepsilon_1, \varepsilon_2 > 0$ such that $\alpha = \varepsilon_2/(\varepsilon_1+\varepsilon_2)$, and $(x^1, \eta^1) \in X$

Step 1. Solve Subproblem LPs.

Initialize $\beta_k \leftarrow \mathbf{0}$, $\beta_k^0 \leftarrow 0$, $\sigma_k \leftarrow \mathbf{0}$, $\sigma_k^1 \leftarrow 0$ and $\sigma_k^0 \leftarrow 0$, $\bar{\nu}_k \leftarrow 0$.

for each $\omega \in \Omega$ solve subproblem (18)

get $Q(x^k, \omega)$ and dual solution $\pi^k(\omega)$

if $c^\top x^k + Q(x^k, \omega) > \eta^k$

set $\iota(\omega) = 1$

else

set $\iota(\omega) = 0$.

compute $\bar{\nu}_k \leftarrow \bar{\nu}_k + \iota(\omega)p(\omega)[c^\top x^k + Q(x^k, \omega) - \eta^k]$

compute $\beta_k \leftarrow \beta_k + p(\omega)T(\omega)^\top \pi^k(\omega)$

compute $\beta_k^0 \leftarrow \beta_k^0 + p(\omega)\pi^k(\omega)^\top r(\omega)$

compute $\sigma_k \leftarrow \sigma_k + \iota(\omega)p(\omega)T(\omega)^\top \pi^k(\omega)$

compute $\sigma_k^1 \leftarrow \sigma_k^1 + \iota(\omega)p(\omega)$

compute $\sigma_k^0 \leftarrow \sigma_k^0 + \iota(\omega)p(\omega)\pi^k(\omega)^\top r(\omega)$

set $u_k \leftarrow \min\{(1-\lambda\varepsilon_1)c^\top x^k + \lambda\varepsilon_1\eta^k + (1-\lambda\varepsilon_1)\sum_{\omega\in\Omega} p(\omega)Q_\eta(x^k, \omega) + \lambda(\varepsilon_1 + \varepsilon_2)\bar{\nu}_k,\ u_{k-1}\}$

if u_k is updated set incumbent solution $(x_\epsilon^*, \eta_\epsilon^*) \leftarrow (x^k, \eta^k)$

if $u_k - \ell_k \leq \epsilon|u_k|$, stop and declare $(x_\epsilon^*, \eta_\epsilon^*)$ ϵ-optimal.

Step 2. Update and Solve the Master Problem.

add optimality cut $\beta_k^\top x + \gamma \geq \beta_k^0$ to master problem (17)

add optimality cut $\sigma_k^\top x + \sigma_k^1 \eta + \zeta \geq \sigma_k^0$ to master problem (17)

solve master problem to get optimal value ℓ_{k+1}
and solution $(x^{k+1}, \eta^{k+1}, \gamma^{k+1})$

set $\max\{\ell_{k+1},\ \ell_k\}$.

4 Computational results

In this section we describe the test instances we used in our computational experiments and report the results. The experiments were carefully designed to investigate the

Table 1 Standard test problems

Name	Application	Scenarios	First-stage (Cons., Vars.)	Second-stage (Cons., Vars.)
cep1	Capacity expansion planning	216	(9, 8)	(7, 15)
pgp2	Power generation planning	576	(2, 4)	(7, 12)
gbd	Aircraft allocation	6.5×10^5	(4, 17)	(5, 10)
LandS	Electricity planning	10^6	(2, 4)	(7, 12)
20term	Vehicle assignment	1.1×10^{12}	(3, 64)	(4, 17)
ssn	Telecom network design	10^{70}	(1, 89)	(175, 706)
storm	Cargo flight scheduling	6×10^{81}	(185, 121)	(528, 1259)

empirical performance of the algorithms ASD-AGG, ASD-SEP, QDEV-AGG, and QDEV-SEP; understand the effect of using ASD and QDEV for different instances; and investigate when it is appropriate to use the risk-averse approach versus the risk-neutral approach. To evaluate performance of the algorithms, the following measures were used: *computational* (CPU) *time*, number of *algorithm iterations*, and *objective value* at termination.

The algorithms were implemented in C++ using the IBM CPLEX Callable Library version 12.0 [7] in the Microsoft Visual Studio 2010 environment. The object-oriented approach was used in coding the algorithms by creating our own *classes* and *methods* that interact with the CPLEX Callable Library's *functions*. Our code includes five major classes: *LPobjectClass, Reader, Master, Sublp* and *Algorithm*. The class *LPobjectClass* is a superclass and the rest of the classes inherit it except the *Algorithm* class. The *Reader* class reads standard stochastic programming test instances in SMPS (Stochastic Mathematical Programming Society) INDEPENDENT format. The *Master* class handles the master program aspects while *Sublp* deals with the subproblem. Finally, the algorithms are implemented in *Algorithm* class, which contains the C++ *main* program. All the experiments were conducted on a personal computer running Intel Pentium 300 GHz, 3.49 GB RAM processor with Windows XP Professional X64 Edition Version 2003 operating system. Next we describe the test instances.

4.1 Standard test instances

We used well-known standard SP test instances from the literature. These instances are described in [6,12]. They can be accessed at http://pages.cs.wisc.edu/~swright/stochastic/sampling/. The names of the instances are as follows: *cep1* [6], *pgp2* [6], *gbd* [3], *LandS* [13], *20term* [14], *ssn* [26], and *storm* [18]. The characteristics of these test instances are summarized in Table 1. The columns of the table give the instance name, application, number of scenarios, number of first-stage constraints (Cons.) and variables (Vars.), and the number of second-stage constraints and variables. The instances are listed in increasing number of scenarios.

Problem *cep1* is a two-stage machine capacity expansion planning problem. Its first-stage variables represent the number of weekly hours of new capacity assigned to each machine. The second-stage variables define the number of hours a machine is assigned to process a specific part. The weekly demands are treated as identically and identically distributed (*iid*) random variables coming from a known distribution. The objective function minimizes weekly amortized expansion cost (per hour) plus the expected weekly production costs.

Problem *pgp2* is a power generation planning problem and deals with electrical capacity expansion to select the minimum cost strategy for investing in electricity generated from gas-fired, coal-fired, and nuclear generators. The first-stage variables model the annualized amount of power generation (kW) based on the specific type of generator acquired, while the second-stage decisions select a specific operational plan to satisfy the realized regional demand (based on satisfying kW of demand from a specific type of generator). The second-stage power generation costs and regional demands are the stochastic elements of this model.

Problem *gbd* is an aircraft allocation problem whose objective is to maximize profit while allocating different types of aircraft to routes with uncertain demand. There are costs related to bumping passengers (when demand exceeds capacity) and operating the plane. The first-stage variables select the number of aircraft (from the four types) assigned to a particular route, while the constraints bound the availability of the aircrafts. The second-stage variables indicate the number of carried and bumped passengers on each of the five routes, and the constraints balance demand for the routes. The demand, which is the right hand side (RHS) values of second-stage balance constraints, is stochastic.

LandS is a problem from electrical investment planning. The first-stage variables represent capacities of four new technologies and second-stage variables represent the use of the four technologies to produce electricity through three different modes. Constraints from the first-stage program signify minimum total capacity and budget constraints, while second-stage constraints include three random RHS demand constraints. Problem *20term* models a motor freight carrier's operations. In this problem the first-stage variables are the positions of fleet vehicles at the start of the day. The second-stage variables move the fleet through a network to satisfy point-to-point demand for shipments. Unmet demand is penalized and the fleet must finish the day with the same fleet configuration it had at the start of the day.

Problem *ssn* concerns telecommunications network design. This problem is a budget-constrained telephone-switching network expansion problem whose objective is to add capacity (in the form of lines) to a network of existing point-to-point connections to minimize the amount of unmet requests for service. The first-stage decision vector allocates capacity to routes (a sequence of lines or links connecting nodes) before the service requests occur. The second-stage decision variables seek to efficiently route call requests to allow smooth operation of the entire network, while minimizing the number of unserved service requests. The stochastic parameter demand is defined as the number of requests for connections at a given instance of time.

Finally, *storm* is a problem used by the U.S. military to plan the allocation of aircraft routes during the Gulf War of 1991. It is modeled over two periods that plans flights over a set of network routes with uncertain amounts of cargo. Flight routes are scheduled in the first-stage and the objective is to minimize the cost of scheduled flights plus the uncertain penalty and cargo handling costs. The second-stage occurs after demand has been realized, and allocates the cargo delivery routes and seeks to satisfy any unmet demand while minimizing holding and penalty cost.

4.2 Modified test instance distributions

The probability distributions of most of the standard test instances are either uniform, that is, they have equiprobable sample spaces for the marginal distributions, or the are normal-like. Therefore, additional test instances were created by modifying the marginal distributions so that the tail and mid-section probabilities were increased or decreased. This was done for the purpose of studying the impact of the risk-neutral and risk-averse approaches on the optimal solution. Modifying the distributions was done through experimentation by changing the marginal distribution probabilities in the instance's STOCH file. The standard SP test instances are represented using the Stochastic Mathematical programming society (SMPS) format, which has three data files, the CORE file, TIME file and STOCH file. The CORE file has the 'core' problem in MPS format while the TIME file has the timing information for splitting the instance into first- and second-stage problems. The STOCH file stores the stochastic scenario data in INDEPENDENT format. The STOCH files has marginal distributions for each random variable in the SP instance. It is these marginal distributions that we modified to create additional test instances *cep1a, cep1sk, pgp2e, pgp2f, gbd_sk3, 20Tr_lsk1, 20Tr_msk1* and *20Tr_msk2*. These instances are available for the interested reader through the corresponding author's web site.

Since the test instances *20term*, *ssn*, and *storm* have a very large number of scenarios to solve, smaller or 'truncated' (Tr) versions of these instances were created by reducing the number of random variables in the STOCH file. These instances are suffixed with 'low' (_l), 'medium' (_m), and 'high' (_h). This allowed us to analyze the impact of problem size on algorithm performance without resorting to sampling approaches. The characteristics of all the test instances are given in Table 2. The original standard test instance names are shown in bold font.

4.3 Results

We first present computational results regarding the empirical performance of the four algorithms: ASD-AGG, ASD-SEP, QDEV-AGG and QDEV-SEP algorithms. We then report the instance optimal values and draw insights from the results regarding risk-neutral versus risk-averse approaches. Each algorithm was run on every test instance for values of risk tradeoff parameter $\lambda \in \{0, 0.1, 0.2, \ldots, 1\}$. For each run we recorded the number of algorithm iterations and computation time (CPU) in seconds (s). We then calculated the average and standard deviation (Stdev) over all the λ values. Solving for different λ values allowed us to trace the efficient frontier for each test instance. Also, we set $\alpha = 0.5$ in the QDEV model by having $\varepsilon_1 = \varepsilon_2 = 1$. Recall that $\alpha = \varepsilon_2/(\varepsilon_1 + \varepsilon_2)$, $\varepsilon_1, \varepsilon_2 > 0$. This was done to provide a fair comparison between the ASD and QDEV approaches. Finally, we set $\epsilon = 10^{-6}$ in all the algorithm runs.

Table 2 Test instance characteristics

Name	Application	Scenarios	First-stage (Cons., Vars.)	Second-stage (Cons., Vars.)
cep1	Capacity expansion	216	(9, 8)	(7, 15)
cep1a	Planning	216	(9, 8)	(7, 15)
cep1sk		216	(9, 8)	(7, 15)
pgp2	Power generation	576	(2, 4)	(7, 12)
pgp2e	Planning	576	(2, 4)	(7, 12)
pgp2f		576	(2, 4)	(7, 12)
gbd	Aircraft allocation	6.5×10^5	(4, 17)	(5, 10)
gbd_sk3		6.5×10^5	(4, 17)	(5, 10)
LandS	Electricity planning	10^6	(2, 4)	(7, 12)
20term	Vehicle assignment	1.1×10^{12}	(3, 64)	(4, 17)
20Tr_l		512	(3, 64)	(4, 17)
20Tr_lsk1		512	(3, 64)	(4, 17)
20Tr_m		1024	(3, 64)	(4,17)
20Tr_msk1		1024	(3, 64)	(4,17)
20Tr_msk2		1024	(3, 64)	(4,17)
20Tr_h		2048	(3, 64)	(4,17)
ssn	Telecom network	10^{70}	(1, 89)	(175, 706)
ssnTr_l	Design	735	(1, 89)	(175, 706)
ssnTr_m		5145	(1, 89)	(175, 706)
ssnTr_h		36015	(1, 89)	(175, 706)
storm	Cargo flight	6×10^{81}	(185, 121)	(528, 1259)
stormTr_l	Scheduling	625	(185, 121)	(528, 1259)
stormTr_m		15625	(185, 121)	(528, 1259)
stormTr_h		390625	(185, 121)	(528, 1259)

Algorithm performance

Table 3 shows the results for ASD-AGG and ASD-SEP algorithms applied to the ASD instances. As can be seen in the table, ASD-SEP takes the same or fewer iterations for all the instances except for *ssnTr_h*. However, both algorithms have comparable performance in terms of average CPU time, with ASD-SEP having slightly better performance overall. The Stdev values for both algorithms are also comparable. Nevertheless, there is variability in CPU time across different values of the risk tradeoff parameter λ. The test results for QDEV-AGG and QDEV-SEP algorithms are reported in Table 4. Like in the ASD case, the separate cut algorithm, QDEV-SEP, takes fewer iterations, but outperforms QDEV-AGG in terms of average CPU time. The algorithms have comparable performance only for *cep1* instances. As with ASD, the Stdev values are comparable but still show variability in CPU time across different λ values. Recall

Table 3 Performance of ASD-AGG and ASD-SEP algorithms

Instance	**ASD-AGG**				**ASD-SEP**			
	Iters		CPU(s)		Iters		CPU(s)	
	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev
cep1	2.00	0.00	0.02	0.01	2.00	0.00	0.02	0.00
cep1a	2.00	0.00	0.03	0.01	2.00	0.00	0.02	0.00
cep1sk	6.00	0.00	0.04	0.03	6.00	0.00	0.05	0.02
pgp2	33.91	2.63	0.57	0.08	31.45	2.30	0.55	0.09
pgp2e	37.91	2.63	0.52	0.07	35.36	2.50	0.51	0.06
pgp2f	37.64	3.29	0.52	0.10	33.09	3.83	0.45	0.07
gbd	33.55	3.72	258.18	30.52	31.55	2.02	242.43	24.18
gbd_sk3	27.09	7.37	186.67	51.26	22.73	5.00	154.02	33.74
LandS	31.00	1.61	389.40	18.61	30.18	1.25	382.81	32.09
20Tr_l	2250.91	332.88	1227.78	217.13	2185.73	572.10	1350.58	507.60
20Tr_lsk1	1738.00	152.01	972.01	108.19	1587.00	166.80	914.05	112.60
20Tr_m	1912.09	316.13	1784.00	318.54	1804.36	233.99	1739.86	244.41
20Tr_msk1	2108.00	217.13	2310.71	287.26	1783.18	246.02	1995.83	382.34
20Tr_msk2	1865.55	238.40	1861.99	297.92	1676.91	238.75	1738.45	321.48
20Tr_h	1757.00	106.73	3088.27	218.56	1505.45	165.88	2631.08	273.72
ssnTr_l	352.91	149.69	90.51	38.57	317.09	151.55	86.45	41.47
ssnTr_m	983.45	1339.86	1647.08	2318.09	890.73	990.84	1491.98	1699.64
ssnTr_h	788.40	300.45	6929.49	2620.53	1013.55	423.16	8876.52	3665.14
stormTr_l	16.00	0.00	2.36	0.09	16.00	0.00	2.42	0.09
stormTr_m	13.00	0.00	47.68	0.35	13.00	0.00	48.25	0.47
stormTr_h	13.00	0.00	1193.81	5.32	13.00	0.00	1195.29	10.24

that in Tables 3 and 4 instances for each problem are listed in order of increasing number of scenarios. Thus we see a general increase in CPU time with the number of scenarios. However, the number of algorithm iterations does not necessarily increase with the number of scenarios.

The fact that the aggregated cut algorithms take more iterations than the separate cut algorithms is expected. This is because more information (in form of disaggregated optimality cuts) is passed on from the subproblems to the master program in the separate cut algorithms. Thus ASD-AGG and QDEV-AGG should have at least as many iterations as ASD-SEP and QDEV-SEP. Now with regards to CPU time, the results show comparable performance of the algorithms for instances with ASD. However, QDEV-SEP performs the best for instances with QDEV in terms of both iterations and CPU time.

We also looked at the optimal objective function values for each tradeoff parameter λ from zero to one for the ASD and QDEV models and observed that QDEV provides a larger (almost twice) range than ASD. This indicates the 'flexibility' of the QDEV model to changes in the risk tradeoff parameter λ. This suggests that the ASD model

Table 4 Performance of QDEV-AGG and QDEV-SEP algorithms

Instance	QDEV-AGG				QDEV-SEP			
	Iters		CPU(s)		Iters		CPU(s)	
	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev
cep1	2.91	0.30	0.03	0.02	2.91	0.30	0.03	0.02
cep1a	6.45	1.51	0.07	0.02	6.45	1.51	0.07	0.04
cep1sk	11.73	2.05	0.11	0.03	9.82	1.78	0.10	0.06
pgp2	50.09	7.76	0.87	0.16	37.00	6.47	0.67	0.14
pgp2e	58.36	8.10	0.84	0.14	43.36	8.83	0.62	0.15
pgp2f	56.55	8.08	0.79	0.18	42.45	6.73	0.59	0.12
gbd	55.09	10.02	519.80	88.33	40.36	9.84	297.45	57.51
gbd_sk3	35.91	5.70	320.28	51.18	28.18	4.90	197.56	30.80
LandS	47.64	6.34	651.46	58.95	34.18	5.67	453.63	92.88
20Tr_l	3112.73	464.63	1895.36	283.70	2289.36	821.20	1338.53	539.05
20Tr_lsk1	2779.45	717.35	1700.16	445.33	1957.27	682.48	1180.37	461.43
20Tr_m	3045.18	779.73	3146.00	748.37	2186.45	753.56	2188.09	806.54
20Tr_msk1	2959.27	806.92	3347.66	874.85	2186.18	563.99	2473.89	621.80
20Tr_msk2	2518.45	702.65	2966.15	785.77	2212.82	675.52	2403.39	808.36
20Tr_h	2645.09	639.44	5066.86	1016.04	2070.55	733.13	3681.41	1298.40
ssnTr_l	1041.18	486.43	288.99	133.31	975.27	534.83	265.77	160.64
ssnTr_m	1694.18	773.04	2853.00	1290.11	1532.55	577.68	2514.10	970.38
ssnTr_h	1915.70	1453.51	17538.67	12339.23	1676.82	474.47	14395.34	3999.08
stormTr_l	38.00	8.11	8.23	2.08	26.45	8.37	3.93	1.14
stormTr_m	39.55	9.08	213.94	53.71	24.64	9.40	89.02	32.93
stormTr_h	38.82	9.09	4592.92	1128.29	24.36	9.37	2175.38	805.88

is relatively more conservative compared to the QDEV model. This was also observed in [1] for a 100 scenario instance of *gbd*.

Original versus modified test instance marginal distributions

To provide further details on the differences between the ASD and QDEV models, we show detailed results for using the ASD-AGG and QDEV-AGG algorithms to solve instances with original and modified marginal distributions, respectively. Most of the standard test instances have marginal distributions which are either *uniform* or *normal*-like. Therefore, we modified the marginal distributions by increasing and/or decreasing the tail and mid-section probabilities. Figure 1 shows a plot of the original and modified marginal distributions for cep1 and pgp2. Notice that instance cep1 has uniform distribution for all three random variables, while pgp2 has normal-like distributions. These distributions were modified has shown in the figure through experimentation so that the both the tail and mid-section distributions were either increased or decreased to make them amenable to the risk averse decision-making setting.

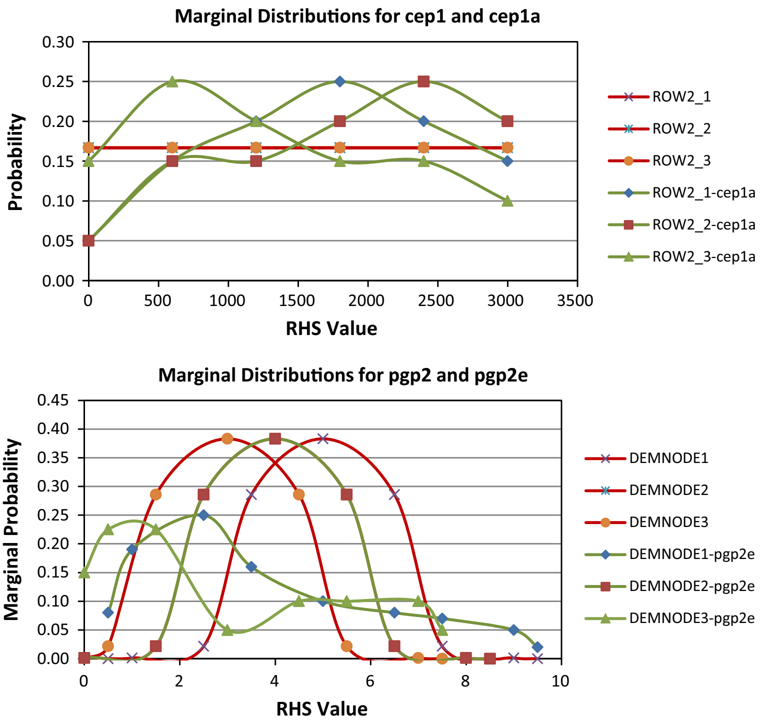

Fig. 1 Example plot of original and modified marginal distributions for cep1 and pgp2

We report detailed results tables for *pgp2* to illustrate our findings but do not include such tables for the rest of the instances due to space considerations. Tables 5 and 6 show the results for the ASD model for *pgp2* (with the normal-like distribution) and for *pgp2e* (modified distribution), respectively. The columns of the table show for each tradeoff parameter λ the overall objective value (Obj.), the mean value $\mathbb{E}[f(x, \tilde{\omega})]$, the ASD value $\mathbb{E}[\nu(\tilde{\omega})]$, and the corresponding mean weighted value $(1-\lambda)\mathbb{E}[f(x, \tilde{\omega})]$ and ASD weighted value $\lambda\mathbb{E}[\nu(\tilde{\omega})]$. These parameters correspond to the objective function of the ASD DEP formulation (9). As can be seen in Table 5, the value $\mathbb{E}[f(x, \tilde{\omega})]$ remains fairly constant across all λ values. There is only a slight change at $\lambda = 0.6$ and $\lambda = 0.9$. This means that the optimal solution is not sensitive to the risk level. Thus the risk-neutral approach provides an optimal solution (at $\lambda = 0$) that is the same or very close to the optimal solution for $\lambda = 0.1$ to $\lambda = 1.0$. On the other hand, in Table 6 we see that the mean value $\mathbb{E}[f(x, \tilde{\omega})]$ varies from $\lambda = 0.4$ to $\lambda = 1.0$, an indication that it is worthwhile to use the mean-ASD model in this case where the distribution is modified. The tradeoff between the mean value and the deviation value is better seen using the plot in Fig. 2. In the figure, 'Mean' is the value of $\mathbb{E}[f(x, \tilde{\omega})]$, 'Wt. Mean' is the value of $(1-\lambda)\mathbb{E}[f(x, \tilde{\omega})]$, and 'Wt. ASD' is the value of $\lambda\mathbb{E}[\nu(\tilde{\omega})]$. Observe how the 'Mean' value is fairly constant for *pgp2* as compared to *pgp2e*. The expected value $\mathbb{E}[f(x, \tilde{\omega})]$ for different λ values changes with corresponding changes in the first-stage optimal solution x, which are reported in Table 7.

Table 5 Results of ASD-AGG for pgp2

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\mathbb{E}[\nu(\tilde{\omega})]$	$(1-\lambda)^*$ $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda\mathbb{E}[\nu(\tilde{\omega})]$
0.0	447.32	447.32	474.00	447.32	0.00
0.1	449.99	447.32	474.00	402.59	47.40
0.2	452.66	447.32	474.00	357.86	94.80
0.3	455.33	447.32	474.00	313.13	142.20
0.4	457.99	447.32	474.00	268.40	189.60
0.5	460.66	447.33	474.00	223.66	237.00
0.6	463.28	447.60	473.74	179.04	284.25
0.7	465.90	447.60	473.74	134.28	331.62
0.8	468.51	447.60	473.74	89.52	378.99
0.9	471.12	447.90	473.70	44.79	426.33
1.0	473.70	447.90	473.70	0.00	473.70

Table 6 Results of ASD-AGG for pgp2e

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\mathbb{E}[\nu(\tilde{\omega})]$	$(1-\lambda)^*$ $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda\mathbb{E}[\nu(\tilde{\omega})]$
0.0	413.94	413.94	464.08	413.94	0.00
0.1	418.95	413.94	464.08	372.54	46.41
0.2	423.96	413.94	464.08	331.15	92.82
0.3	428.98	413.94	464.08	289.76	139.22
0.4	433.97	414.00	463.93	248.40	185.57
0.5	438.97	414.00	463.93	207.00	231.96
0.6	443.91	414.28	463.66	165.71	278.20
0.7	448.85	414.28	463.66	124.29	324.56
0.8	453.73	415.06	463.40	83.01	370.72
0.9	458.48	416.42	463.15	41.64	416.83
1.0	463.15	416.42	463.15	0.00	463.15

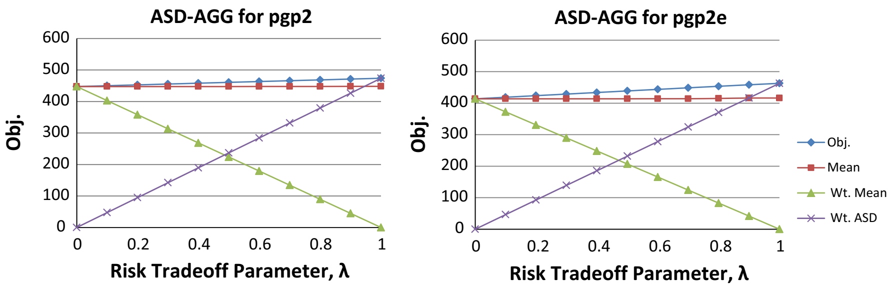

Fig. 2 ASD optimal values versus risk tradeoff parameter λ

Table 7 Optimal first-stage ASD solutions for pgp2 and pgp2e

λ	pgp2				pgp2e			
	x_1	x_2	x_3	x_4	x_1	x_2	x_3	x_4
0.0	1.5	5.5	5.0	5.5	1.5	6.5	3.5	9.0
0.1	1.5	5.5	5.0	5.5	1.5	6.5	3.5	9.0
0.2	1.5	5.5	5.0	5.5	1.5	6.5	3.5	9.0
0.3	1.5	5.5	5.0	5.5	1.5	6.5	3.5	9.0
0.4	1.5	5.5	5.0	5.5	1.5	7.0	3.5	8.5
0.5	1.5	5.5	5.0	5.5	1.5	7.0	3.5	8.5
0.6	0.5	5.5	6.0	5.5	2.5	6.0	3.5	8.5
0.7	0.5	5.5	6.0	5.5	2.5	6.0	3.5	8.5
0.8	0.5	5.5	6.0	5.5	2.0	5.5	4.5	8.5
0.9	0.0	5.5	6.5	5.5	1.5	5.5	5.0	9.0
1.0	0.0	5.5	6.5	5.5	1.5	5.5	5.0	9.0

Recall that problem *pgp2* deals with electrical capacity expansion to select the minimum cost strategy for investing in electricity generated from gas-fired (x_1), coal-fired (x_2), nuclear (x_3) and 'other' (x_4) generators. The first-stage variables model the annualized amount of power generation (kW) based on the specific type of generator acquired at a given annualized capital cost ($/kW), while the second-stage decisions select a specific operational plan to satisfy the realized regional demand. We observe that under original regional demand (pgp2) the optimal solution values for x_2 and x_4 remain fixed for all λ values. The optimal solution values for x_1 and x_3 only change from $\lambda = 0.5$ to 0.6, and then from $\lambda = 0.8$ to 0.9. Observe that as the risk level is increased power generation from gas-fired is reduced while that from nuclear is increased. For the modified demand distribution (pgp2e), however, we see significant changes to the optimal solutions across different lambda values, an indication of the sensitivity of the solution to the mean-risk parameter.

Table 8 shows results for the QDEV model for *pgp2* while Table 9 gives the results for *pgp2e*. The columns of the tables show for each λ the overall objective value (Obj.), the 'mean' value $\mathbb{E}[f(x, \tilde{\omega})]$, expected quantile deviation value $\mathbb{E}[\nu(\tilde{\omega})]$, the target value η, weighted mean value $(1-\lambda\varepsilon_1)\mathbb{E}[f(x, \tilde{\omega})]$, weighted quantile deviation value $\lambda(\varepsilon_1+\varepsilon_2)\mathbb{E}[\nu(\tilde{\omega})]$, and the weighted target value $\lambda\varepsilon_1\eta$. These parameters correspond to the objective function of the QDEV DEP formulation (10). In Table 8 we see that there is more variation in the mean value $\mathbb{E}[f(x, \tilde{\omega})]$ as compared to the ASD results in Table 5. In fact now we see a big change from $\lambda = 0.0$ to $\lambda = 1$. Nevertheless, the optimal value from $\lambda = 0$ to $\lambda = 0.7$ does not change significantly due to the original distribution for this instance. In Table 9 we see that for the modified distribution case $\mathbb{E}[f(x, \tilde{\omega})]$ varies from $\lambda = 0.0$ to $\lambda = 1$. This means that the optimal solution is very responsive to the level of risk. Thus it is better to use the risk-averse model instead of the risk-neutral approach in this case. Figure 3 plots the tradeoff between the mean value and the QDEV value. In the plot 'Wt. QDEV' is the value of $(1-\lambda\varepsilon_1)\mathbb{E}[f(x, \tilde{\omega})]$ and 'Wt. η' is the value of $\lambda\varepsilon_1\eta$. Now we can clearly see the relative variation of the 'Mean' value for *pgp2* as compared to *pgp2e*. The variation of the optimal first-stage

Table 8 Results of QDEV-AGG for pgp2

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\mathbb{E}[\nu(\tilde{\omega})]$	η	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\tilde{\omega})]$	$\lambda\varepsilon_1\eta$
0.0	447.324	447.324	447.324	0	447.324	0	0
0.1	452.638	447.324	28.482	443.500	402.592	5.696	44.350
0.2	457.952	447.324	28.484	443.495	357.859	11.394	88.699
0.3	463.234	447.597	27.510	444.700	313.318	16.506	133.410
0.4	468.447	447.597	27.515	444.691	268.558	22.012	177.876
0.5	473.624	447.896	27.025	445.301	223.948	27.025	222.651
0.6	478.770	447.896	27.026	445.300	179.159	32.431	267.180
0.7	483.915	447.896	27.026	445.300	134.369	37.836	311.710
0.8	489.037	448.138	26.981	445.300	89.628	43.169	356.240
0.9	494.149	448.138	26.981	445.300	44.814	48.566	400.770
1.0	499.259	469.253	26.979	445.300	0	53.959	445.300

Table 9 Results of QDEV-AGG for pgp2e

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\mathbb{E}[\nu(\tilde{\omega})]$	η	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\tilde{\omega})]$	$\lambda\varepsilon_1\eta$
0.0	413.935	413.935	413.935	0	413.935	0	0
0.1	423.571	413.936	61.970	386.349	372.543	12.394	38.635
0.2	433.201	414.003	61.747	386.500	331.203	24.699	77.300
0.3	442.733	414.284	61.307	386.500	289.999	36.784	115.950
0.4	452.170	415.062	60.066	387.700	249.037	48.053	155.080
0.5	461.410	415.408	59.556	388.300	207.704	59.556	194.150
0.6	470.428	416.419	57.567	391.300	166.567	69.081	234.780
0.7	479.330	417.841	56.691	392.300	125.352	79.368	274.610
0.8	488.108	419.171	55.021	395.300	83.834	88.033	316.240
0.9	496.702	420.820	54.317	396.500	42.082	97.770	356.850
1.0	505.030	450.983	53.965	397.100	0	107.930	397.100

solutions are given in Table 10. Unlike with the ASD case now we see more variation in the optimal solutions for different λ values. Nevertheless, the modified demand distribution case (pgp2e) has more variation in the optimal solutions compared to the original case (pgp2).

5 Conclusion

Mean-risk stochastic programs include a risk measure in the objective to model risk averseness for optimization problems under uncertainty. In this work we report on a computational study of mean-risk two-stage stochastic linear programs with recourse based on absolute semideviation and quantile deviation. The study was aimed at

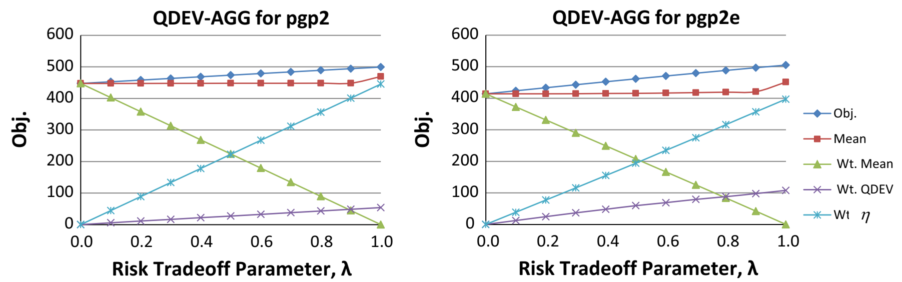

Fig. 3 QDEV optimal values versus risk tradeoff parameter λ

Table 10 Optimal first-stage QDEV solutions for pgp2 and pgp2e

λ	pgp2				pgp2e			
	x_1	x_2	x_3	x_4	x_1	x_2	x_3	x_4
0.0	1.5	5.5	5.0	5.5	1.5	6.5	3.5	9.0
0.1	1.5	5.5	5.0	5.5	1.5	6.5	3.5	9.0
0.2	1.5	5.5	5.0	5.5	1.5	7.0	3.5	8.5
0.3	0.5	5.5	6.0	5.5	2.5	6.0	3.5	8.5
0.4	0.5	5.5	6.0	5.5	2.0	5.5	4.5	8.5
0.5	0.5	5.5	6.5	5.5	1.5	5.5	5.0	8.5
0.6	0.0	5.5	6.5	5.5	1.5	5.5	5.0	9.0
0.7	0.0	5.5	6.5	5.5	2.5	5.5	5.0	8.0
0.8	1.0	4.5	6.5	5.5	2.5	5.5	5.0	8.5
0.9	1.0	4.5	6.5	5.5	1.5	5.5	6.0	8.5
1.0	1.7	3.8	6.5	5.5	1.0	5.5	6.5	8.5

performing an empirical investigation of decomposition algorithms for this class of stochastic programs, analyzing how the instance solutions vary across different levels of risk; and understanding when it is appropriate to use a given mean-risk measure. In particular, we implemented and tested two variants of subgradient-based optimization algorithms for the absolute semideviation (ASD) and quantile deviation (QDEV) stochastic programs, respectively. The computational study provides several insights. In particular, the results show that the risk-neutral approach is still appropriate for most of the standard stochastic programming test instances. This is because most of them have random variables with uniform or normal-like marginal distributions. However, when the distributions are modified by increasing and/or decreasing tail and mid-section probabilities, the risk-neutral approach may no longer be appropriate and the risk-averse approach becomes necessary. The results also show that absolute semideviation is a more conservative mean-risk measure than quantile deviation. Future work along this line of work include implementing multicut versions of the algorithms proposed in this paper, devising similar algorithms for mean-risk stochastic integer programs, and investigating the on-demand accuracy approach [4] for this class of problems which was recently applied to risk-averse two-stage stochastic programs with a conditional value-at-risk constraint [5].

Acknowledgments The authors would like to thank the anonymous referees who provided valuable comments that helped improve this paper. The first author was supported in part by a Graduate Assistantship in Areas of National Need (GAANN) Fellowship.

References

1. Ahmed, S.: Convexity and decompostion of mean-risk stochastic programs. Math Program **106**(3), 433–446 (2006)
2. Artzner, P., Delbean, F., Eber, J., Heath, D.: Coherent measures of risk. Math Finance **9**, 203–228 (1999)
3. Dantzig, G.: Linear programming and extensions. Princeton University Press, Princeton (1963)
4. de Oliveira, W., Sagastizabal, C.: Level bundle methods for oracles with on-demand accuracy. Optim Meth and Soft **29**, 1180–1209 (2014)
5. Fabian, C., Wolf, C., Koberstein, A., Suhl, L.: Risk-averse optimization in two-stage stochastic models: computational aspects and a study. SIAM J Optim **25**, 28–52 (2015)
6. Higle, J., Sen, S.: Stochastic decomposition. Kluwer Academic Publishers, 101 Phillip Drive, Norwell, MA 02061 (1996)
7. IBM: CPLEX 12.1 IBM ILOG CPLEX Callable Library Version 12.1 C API Reference Manual. IBM, USA (2009)
8. Konno, H., Yamazaki, H.: Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market. Manag Sci **37**(5), 519–531 (1991)
9. Kristoffersen, T.: Deviation measures in linear two-stage stochastic programming. Math Methods Oper Res **62**(2), 255–274 (2005)
10. Krokhmal, P., Zabarankin, M., Uryasev, S.: Modeling and optimization of risk. Surv Oper Res Manag Sci **16**, 49–66 (2011)
11. Linderoth, J., Wright, S.: Decomposition algorithms for stochastic programming on a computational grid. Comput Optim Appl **24**(2), 207–250 (2003)
12. Linderoth, J., Shapiro, A., Wright, S.: The empirical behavior of sampling methods for stochastic programming. Ann Oper Res **142**, 215–241 (2006)
13. Louveaux, F., Smeers, Y.: Numerical techniques for stochastic optimization problems. In: Ermoliev, Y., Wets, R. (eds.) Optimal investments for electricity generation: a stochastic model and a test problem, pp. 445–452. Springer-Verlag, Berlin (1998)
14. Mak, W., Morton, D., Wood, R.: Monte carlo bounding techniques for determining solution quality in stochastic programs. Oper Res Lett **24**, 47–56 (1999)
15. Märkert, A., Schultz, R.: On deviation measures in stochastic integer programming. Oper Res Lett **33**(5), 441–449 (2005)
16. Markowitz, H.: Portfolio selection. J Finance **7**(1), 77–91 (1952)
17. Miller, N.: Mean-risk portfolio optimization problems with risk-adjusted measures. Dissertation, The State University of New Jersey (2008)
18. Mulvey, J., Ruszcynski, A.: A new scenario decomposition method for large scale stochasitc optimization. Oper Res **43**, 477–490 (1995)
19. Ogryczak, W., Ruszcynski, A.: Dual stochastic dominance and related mean-risk models. SIAM J Optim **13**, 60–78 (2002)
20. Rockafellar, R.T., Uryasev, S.: Optimization of conditional value-at-risk. J Risk **2**, 21–41 (2000)
21. Ruszcynski, A., Shapiro, A.: Optimization of convex risk functions. Math Oper Res **31**(3), 433–452 (2006)
22. Schultz, R.: Stochastic programming with integer variables. Math Program **97**, 285–309 (2003)
23. Schultz, R., Neise, F.: Algorithms for mean-risk stochastic integer programs in energy. Revista Investig Operacional **28**(1), 4–16 (2007)
24. Schultz, R., Tiedemann, S.: Risk aversion via excess probabilities in stochastic programs with mixed-integer recourse. SIAM J Optim **14**(1), 115–138 (2003)
25. Schultz, R., Tiedemann, S.: Conditional value-at-risk in stochastic programs with mixed-integer recourse. Math Program **105**, 365–386 (2006)
26. Sen, S., Doverspike, R., Cosares, S.: Network planning with random demand. Telecommun Syst **3**(1), 11–30 (1994)
27. Shapiro, A., Ahmed, S.: On a class of minimax stochastic programs. SIAM J Optim **14**(4), 1237–1249 (2004)