COMPUTATIONAL STUDY OF MEAN-RISK STOCHASTIC PROGRAMS

A Dissertation

by

TANISHA GREEN COTTON

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Lewis Ntaimo
Committee Members,	Sergiy Butenko
	Georgia-Ann Klutke
	Tiffani Williams
Department Head,	César Malavé

May 2013

Major Subject: Industrial Engineering

ABSTRACT

Mean-risk stochastic programs model uncertainty by including risk measures in the objective function. This allows for modeling risk averseness for many problems in science and engineering. This dissertation addresses gaps in the literature on stochastic programs with mean-risk objectives. This includes a need for a computational study of the few available algorithms for this class of problems. The study was aimed at implementing and performing an empirical investigation of decomposition algorithms for stochastic linear programs with absolute semideviation (ASD) and quantile deviation (QDEV) as mean-risk measures. Specifically, the goals of the study were to analyze for specific instances how algorithms perform across different levels of risk, investigate the effect of using ASD and QDEV as risk measures, and understand when it is appropriate to use the risk-averse approach over the risk-neutral one.

We derive two new subgradient based algorithms for the ASD and QDEV models, respectively. These algorithms are based on decomposing the stochastic program stage-wise and using a single (aggregated) cut in the master program to approximate the mean and deviation terms of the mean-risk objective function. We also consider a variant of each of the algorithms from the literature in which the mean-risk objective function is approximated by separate optimality cuts, one for the mean and one for the deviation term. These algorithms are implemented and applied to standard stochastic programming test instances to study their comparative performance. Both the aggregated cut and separate cut algorithms have comparable computational performance for ASD, while the separate cut algorithm outperforms its aggregate counterpart for QDEV. The computational study also reveals several insights on mean-risk stochastic linear programs. For example, the results show that for most standard test instances the risk-neutral approach is still appropriate. We show that this is the case due to the test instances having random variables with uniform marginal distributions. In contrast, when these distributions are changed to be nonuniform, the risk-averse approach is preferred. The results also show that the QDEV mean-risk measure has broader flexibility than ASD in modeling risk.

DEDICATION

The following verses have guided my life and journey through graduate school: Luke 12:48,“And to whomsoever much is given, of him shall much be required”. Phillipians 4:11-12, “I am not saying this because I am in need, for I have learned to be content whatever the circumstances. I know what it is to be in need, and I know what it is to have plenty. I have learned the secret of being content in any and every situation, whether well fed or hungry, whether living in plenty or in want.” Finally the wise words of Reinhold Niebuhr states,“God grant me the serenity to accept the things I cannot change; courage to change the things I can; and wisdom to know the difference.”

ACKNOWLEDGEMENTS

First, I like would to thank God and my Lord and Savior Jesus Christ for whom all my talents and hard work honor. I would also like to thank my husband, Chris for being supportive of the numerous sacrifices made throughout this process. My Mom Beatrice, Dad Timothy, Sister Crystal, Brother Chris, aunts Claudean and Patricia, Mother-in-law Gladys, Sister-in-law Lorieal, and Uncle Claud have all been fervent cheerleaders and encouragers throughout my years in graduate school.

There are also several people at Texas A&M who have helped to guide and mentor me through this journey. First, I am extremely grateful for my advisor Dr. Ntaimo and mentor Dr. Klutke. Also I want to thank my other committee members Dr. Williams and Dr. Butenko. My fellow collegues Michelle Alvardo, Hiram Moya, Samuel Merriweather, and Eric Beier were classmates who have become like family through this PH. D. process. Last, I like to thank the organizations that provided for my financial support during my time at Texas A&M they are: The Alfred Sloan Foundation, U.S. Department of Education through the Graduate Assistantships in Areas and National Need Fellowship, and the Texas A&M Graduate Diversity Fellowship.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Stochastic programming is a branch of mathematical optimization that deals with optimization problems with uncertainty in the problem data. The uncertainty is mathematically represented by random variables with known probability distributions. Historically, a common representation of a stochastic program's objective function uses the expectation to quantify variability of random variables. Using the expectation implies risk neutrality, however, in certain applications such as financial planning, insurance assessment, energy maintenance and planning, and others, it may be more appropriate to explicitly model risk within its objective. Mean-risk stochastic programs represent risk by adding both a dispersion statistic and expectation in the objective function of stochastic programs, so they can reflect the inherent uncertainty in a problem.

This research addresses some gaps in the literature on stochastic linear programs (SLPs) with mean-risk objectives. This includes a need for a computational study of the few available algorithms for this class of problems. Previous computational studies either focus on a specific application (Kristoffersen (2005), Miller (2008)) or test a wide class of problems but only give results showing the efficient frontier of the selected mean-risk measures (Ahmed (2006)). This work implements and characterizes (benchmark) each algorithm's performance on standard instances involving several applications. This dissertation addresses the following research objectives: 1) derive and implement decomposition algorithms for mean-risk stochastic linear programs; and 2) perform computational experiments based on standard test instances to study the empirical performance of the decomposition algorithms with different risk measures over diverse application areas.

Specifically, we implement and analyze decomposition algorithms to solve mean-risk stochastic linear programs with absolute semideviation and quantile deviation risk measures. We derive the Aggregated Cut Subgradient-Based Algorithm (ASD-AGG) and implement the L-shaped Algorithm for Quantile Deviation (QDEV-AGG). Based on (Ahmed (2006)), we implement separate optimality cut versions of the algorithms for each risk measure called the Separate Cut Subgradient-Based Algorithm (ASD-SEP) and Separate Cut L-shaped Algorithm for Quantile Deviation (QDEV-SEP). The goals of this computational study are to investigate: (a) computational performance of ASD-AGG, ASD-SEP, QDEV-AGG, and QDEV-SEP algorithms; (b) effect of using of ASD and QDEV; (c) conditions that justify the use of the risk-averse approach over the risk-neutral one. Our contributions are a new algorithm, ASD-AGG; introduction of the application of the L-shaped Method to mean-risk SLP setting with quantile deviation; an empirical study to compare the results of the aggregated and separate optimality cut algorithms across diverse application areas; and related insights on mean-risk SLPs gained from the study.

The dissertation is organized into six chapters. Chapter 2 contains a concise review of literature on Stochastic Programming in general and then specifically mean-risk stochastic programs. Chapter 3 introduces mean-risk objectives and their deterministic equivalent problems . The decomposition algorithms to solve the mean-risk stochastic linear programs and address research objective (1) are described in Chapter 4. Chapter 5 describes the experimental design and results of the computational study to address research objective (2). Finally, conclusions and future research directions are addressed in Chapter 6.

2. LITERATURE REVIEW

In this section past and current literature relevant to this work is discussed. The first section gives a short overview of the literature of stochastic programming. The second section describes the literature of mean-risk stochastic programs and their related algorithms and computational studies. We begin this section by introducing the origins of stochastic programming.

2.1 Stochastic Programming

In 1955, Dantzig introduced the classical two-stage stochastic linear program with fixed Recourse (Dantzig, 1955). Since then numerous advances and solution methods have been proposed for stochastic linear programs. Many of these methods seek to decompose the problem into smaller parts, because as the number of scenarios (representing uncertainty in problem) increases there is an increasing chance for the stochastic program to become computationally intractable.

In the paper Dantzig and Wolfe (1961), the classic Dantzig-Wolfe decomposition method was presented. In it the LP is decomposed into two types of constraints, those with block angular structure and thus without. Then Minkowski's Finite Basis Theorem is used to generate the extreme points of the polyhedron representing the feasible region for problem. The constraints with block structure are placed in subproblems that are priced out to decide which columns (extreme points) should be added to the Restricted Master program. The master program has the rest of the constraints, but only a subset of the extreme points (columns) are in the problem. The procedure continues until no more columns can enter the basis.

Benders (1962) adapted Kelley's classical method (Kelley, 1960) to solve convex optimization problems that had a block diagonal structure. This structure allowed for the problem to be partitioned into a master problem and subproblem. Before a problem could be solved using Benders' Method it had to be decomposed into the deterministic equivalent problem. This method still created a dense subproblem that was still difficult to solve with larger instances.

The L-shaped Method was proposed in 1969 (Slyke and Wets, 1969). It improved upon Benders' Method by separating the large subproblem into many smaller subproblems. At each iteration all the subproblems are solved and the appropriate feasibility or optimality cut are added to the Master based on the subproblem solution. The L-shaped method works efficiently on many problems, but when the subproblems become dense due to a large number of scenarios associated with the second stage subproblems, there are still practical problems that are intractable to solve.

A different approach to improve on the L-shaped method seeks to decrease the number of scenarios enumerated by the second stage recourse function. One such method uses exterior sampling and is called Sample Average Approximation Algorithm (Kleywegt et al. (2001), Ruszczynski and Shapiro (2003)). This method uses Monte Carlo Sampling to generate a subset of independent samples from the total stochastic scenarios associated with the recourse function. One drawback of this method is that the number of replications and the number of independents samples, used to obtain tight lower and upper bounds on the optimal solution, must be obtained through experimentation. But it still is a promising improvement to the L-shaped method to solve large scale convex optimization problems.

Benders,the L-shaped Method, and any of their numerous variants in the literature are decomposition algorithms that decompose the deterministic equivalent formulation stage-wise. Another decomposition approach, scenario decomposition, instead divides the deterministic equivalent problem by its scenario subproblems in order to exploit its block structure. Then copies of the first-stage variable are made for each scenario, and the nonanticipativity constraint is added which enforces the equivalence of the first stage variables in each subproblem. This allows the two-stage linear program's first-stage variables to remain independent of the scenarios generated in the second stage subproblems. Three representations of nonanticipatiivity are listed as scenario representative, cyclic representation, and in expectation (Birge and Louveaux, 1997). One way this problem can be formulated is by taking the Lagrangian Dual with respect to the nonanticipativity constraints. Then this problem can be solved using subgradient, cutting plane (column generation), bundle (trust region), or augmented Lagrangian methods. For more details on Lagrangian Dual see (Caroe, 1998), (Bazaraa et al., 2005), and (Nemhauser and Wolsey, 1999).

2.2 Mean-Risk Stochastic Programs

The Markowitz (1952) portfolio optimization model, with the mean-variance objective, was the basis of analysis for many years within financial sector. It is from the analysis of this model that led him to lay the foundations for the mean-risk methodology within Markowitz (1987). But several studies underscore the deficiencies in using variance including: its ineffectiveness in measuring small probability events and indiscriminate penalty of differences below and above a targeted value. Markowitz (1959) suggested the use of semivariance as an alternative measure. Furthermore, the mean-variance criterion has been shown by Ruszczynski and Shapiro (2003) to lead to non-convex formulations. These formulations were demonstrated to be computationally intractable even for simple stochastic programs (Ahmed, 2006).

The foundational works in the study of probability objectives in stochastic programming are Berneau (1964) and Charnes and Cooper (1963). The former author introduced the criterion from the context of stochastic linear programs and the latter author from the context of stochastic linear programs with probabilistic constraints. From early works the excess probability was know as the probability of "ruin" in the context of financial markets, while the SLP itself was called the minimum risk problem or criterion for the objective exclusively. More recent mean-risk formulations with excess probability are used in the context of SIP (Schultz, 2003).

Many attempts to linearize the quadratic objective of the mean-variance model led to the consideration of numerous statistics. One such statistic was the Gini's mean (absolute) difference, which was introduced into the mean-risk setting by Yitzhaki (1982). Konno and Yamazaki (1991) present a model where risk is represented by (mean) absolute deviation. An alternate approach, for representing an equivalent SLP of the mean-risk one, was reflected on by Young (1998) with the minimax approach (the worst case performances) to quantify risk. A large family of mean-risk stochastic programs are included in the aforementioned class of minimax stochastic programs. In addition under the assumption of finite discrete random variables, the LP formulations of the mean-risk models are special cases of the multiple criteria LP models of Ogryczak (2000).

In order for the methodology to progress the fundamental question, which risk measures are most suitable for use in mean-risk stochastic programs, had to be answered. The authors begin to address this question by defining "coherent risk measures" in Artzner et al. (1999). They state the axioms of monotonicity, positive homogeneity, subadditivity, and translation invariance define coherence. These axioms became the foundation for selecting sound risk measures from the financial sector to apply in mean-risk stochastic programs.

Ogryczak and Ruszcynski (1999), Ogryczak and Ruszcynski (2001), and Ogryczak and Ruszcynski (2002) address the aforementioned question from a different perspective. They use stochastic dominance theory to compare the returns from portfolios of several risk measures. These works build on the works of Porter (1974) with fixed-target semivariance and Fishburn (1977) with more general risk measures associated with fixed-target downside risk. The authors prove consistence of these measures with first and second degree stochastic dominance relations (under appropriate conditions for λ for specific mean-risk programs). The measures identified by stochastic dominance as sound, agree with those identified as coherent measures in the mean-risk approach. This approach warrants more attention since mean-risk models fail to capture the entire spectrum of risk-averse preferences.

With the introduction of coherent risk measures and stochastic dominance theory, additional mean-risk SLPs were introduced. Ogryczak (2000) contributes the parametric generalization of the mean-semideviation, mean-gini mean difference, and mean-maximum (downside) risk SLPs. CVaR and VaR were first studied for use as measures in this setting by Rockafellar and Uryasev (2000). Next the mean absolute deviation from a quantile (quantile deviation) are described by Ogryczak and Ruszcynski (2002).

Now we focus on the structural properties of the mean-ASD and mean-QDEV SLPs. Ogryczak and Ruszcynski (1999) and Ogryczak and Ruszcynski (2001) analyze mean-ASD models in terms of stochastic dominance relations. They show that the models yield efficient solutions in terms of first and second order stochastic dominance relations and with construction of the mean-risk efficient frontier corresponding to tradeoff coefficients less than one. Similarly, the unique optimal solutions of the mean-QDEV SLP was shown to agree with second order stochastic dominance relations. Also both mean-risk objectives were shown by Ahmed (2006) to satisfy the sufficient conditions to be convexity-preserving and therefore computationally tractable.

Thus far computational results related to mean-risk SLPs are limited. Relevant papers describing algorithms to solve and the computational results for the mean-ASD,mean-QDEV, other related SLPs are described in the paragraphs to follow.

Shapiro and Ahmed (2004) prove the mean-risk model with quantile deviation is equivalent to the minimax model under specific parameter conditions. The l_∞-trust-region based decomposition algorithm of Linderoth and Wright (2003) is used for solving the LPs. The algorithms are implemented in ANSI C with GNU Linear programming Kit. Three 1000 sampled scenario instances of *LandS*, *gbd*, *20term*, and *storm* are used. Each instance uses a reference distribution P^* with equally weighted sampled scenarios. The solutions for some instances are not guaranteed be consistent with stochastic dominance. The quantile dispersion statistics of mean absolute deviation from the median, mean absolute deviation, standard deviation, absolute deviation, and standard deviation. Results for the objective value, CPU time, and iterations were given for the test instances. The level of risk was observed to decrease, when the value of some dispersion statistics decreased . Higher expected cost solutions occur with increasing risk weight in the mean-risk model. Their computational results showed that even with large changes in the reference distribution the optimal objective values changed in a relatively small manner. Also large variability in total CPU time between different risk weight combinations was explained by regularized nature of the mean-risk (or minimax) objective function. For specific risk weight parameters, faster convergence of the algorithm resulted from the sharpness of the piecewise linear objective function.

One paper with results related to a specific problem that uses stochastic linear programs is Kristoffersen (2005). In this paper the deviation measures of central deviation, semideviation, and expected excess are tested on a linear relaxation of a mixed-integer, scheduling problem in chemical production. An L-shaped variant algorithm and its regularized version, which adds a regularizing term to the objective in order to penalize divergence from the current solution, are tested on scenarios ranging from 50 to 1000. 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.

Ahmed (2004)and Ahmed (2006) describe decomposition algorithms for stochastic linear programs using parametric cutting planes for the mean-ASD and mean-quantile algorithms. Using test instances in sizes ranging from 25 to 500, equally likely scenarios were sampled for the problems *LandS*, *gbd*, *20term*, *storm* and *ssn* from Linderoth et al. (2006). Results show the mean-QDEV model offers broader flexibility in the mean-risk tradeoff than the mean-ASD-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 efficient frontier and no other computational data is reported.

Miller (2008) proposes an extended and multi-cut Benders' decomposition method to solve the problem of optimizing a portfolio with finite assets. This is the same problem addressed by Markowitz (1952). The two-stage mean-risk formulations, of mean-semideviation and mean-weighted deviation from a quantile, are solved with a multi-cut and extended Benders' decomposition method. For each mean-risk model the algorithms performance is compared with solving the large (extensive or deterministic equivalent formulation) linear programming formulation for the portfolio optimization problem using the simplex method. The results analyze scenario trees in sizes of 10×10, 20×20, and 50×40 (where 50×40 are the number of first stage nodes $\times$ second stage nodes), their total solution time, and memory usage. The run time results show that using the simplex method to solve the large linear program outperforms both cutting plane methods for both dispersion statistics. The extended Benders' Method requires less memory as the scenario tree size increased for the mean-semideviation model. In addition, the extended Benders' method uses less memory than the multi-cut method.

For more information on mean-risk stochastic programs, applications, algorithms, and risk measures refer to a recent survey Krokhmal et al. (2011).

3. MEAN-RISK STOCHASTIC PROGRAMMING

3.1 Introduction

Consider a risk neutral two-stage stochastic linear program (SLP) with recourse that can be written as follows:

$$\begin{aligned} \text{Min} \quad & \mathbb{E}[f(x,\tilde{\omega})] \\ \text{s.t.} \quad & x \in X \end{aligned} \tag{3.1}$$

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 $(\Omega, \mathcal{A}, \mathbb{P})$. $\mathbb{E} : \mathcal{F} \mapsto \mathbb{R}$ 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 $f(x,\tilde{\omega})$ can be represented as

$$f(x,\tilde{\omega}) = c^{\top}x + Q(x,\omega_{\tilde{\omega}}). \tag{3.2}$$

A scenario defines the realization of the stochastic problem data $\{q(\omega), T(\omega), \text{and } r(\omega)\}$. For any realization (scenario) ω of $\tilde{\omega}$ the recourse function $Q(x,\omega)$ is given by

$$\begin{aligned} Q(x,\omega) = & \,\text{Min}\, q(\omega)^{\top}y(\omega) \\ & \text{s.t. } Wy(\omega) \geq r(\omega) - T(\omega)x \\ & y(\omega) \geq 0, \end{aligned} \tag{3.3}$$

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 righthand side vector. 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, (3.1) is a convex program and is amenable to convex optimization methods. In fact, there have been several works on this subject [Birge and Louveaux (1997), Ruszczynski and Shapiro (2003)]. It will become apparent later why this property of the stochastic linear programs will be important to preserve as they are extended to the mean-risk setting. The challenge in solving large-scale instances of (3.1) lies in evaluating the expectation in multidimensional space.

Modeling problems using strictly the expectation in the objective makes the formulation risk-neutral. To introduce risk sensitivity, a dispersion statistic ($\mathbb{D}$) is added to (3.1) resulting in

$$\begin{aligned} \text{Min} \quad & \mathbb{E}[f(x, \tilde{\omega})] + \lambda \mathbb{D}[f(x, \tilde{\omega})] \\ \text{s.t.} \quad & x \in X \\ & \lambda \geq 0, \end{aligned} \tag{3.4}$$

where λ is a weight that quantifies the tradeoff between expected cost and risk. Risk measure $\mathbb{D}$ is chosen so that (3.4) remains a convex optimization problem, allowing it to have access to the readily available convex optimization methods. Now the objective uses a weighted mean-risk criterion showing the risk-averse approach, and (3.4) results in the so called mean-risk stochastic program.

We consider problem (3.4) 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.

This chapter presents results of a computational study of some of the few available algorithms for this class of problems. This includes characterizing (benchmarking) their performance on standard instances involving several applications. Then new decomposition algorithms are presented for SLPs with mean-risk objectives.

As previously stated, most current computational studies either focus on a specific application ((Kristoffersen, 2005), (Miller, 2008)) or test a wide class of problems but only give results showing the efficient frontier of the selected mean-risk measures ((Ahmed, 2004), (Ahmed, 2006)). The papers previously listed use Benders or L-shaped based algorithms or their variants to solve these problems, however there is still a need for new algorithms. An exception to this trend is shown in the work of Shapiro and Ahmed (2004), where four standard instances were formulated as mean-risk models, by establishing equivalence through the minimax models, where an l_∞-trust-region based decomposition algorithm was used to solve the SLPs. Results from solving the mean-risk SLPs that represented risk with the dispersion statistics of expected cost, mean absolute deviation from the median, mean absolute deviation, standard deviation, absolute semideviation, and standard deviation were shown along with computational results showing the total iterations and CPU time to solve each instance. This chapter extends the aforementioned works by studying the impact of using a single cut to approximate both the $\mathbb{E}[f(x,\tilde{\omega})]$ (mean) and risk measure versus using a separate cut for the $\mathbb{E}[f(x,\tilde{\omega})]$ and risk measure within subgradient cutting-plane algorithms. In addition, the computational study reports detailed results concerning the effect of using of the dispersion statistics of absolute semideviation (ASD) and quantile deviation (QDEV) and conditions that govern the use of the risk-averse approach over the risk-neutral one.

The rest of the chapter is organized as follows. Section 3.2 introduces preliminaries on some common convexity-preserving risk-measures that are used as specifications of $\mathbb{D}$. Their deterministic equivalent formulations or problems (DEPs) are presented in Section 3.3. Now we introduce the objective function of some common mean-risk stochastic programs.

3.2 Mean-Risk Objectives

The dispersion statistics ($\mathbb{D}$) of this section are some of the coherent risk measures from the literature. These risk measures are convexity preserving and include both *deviation* and *quantile* measures. 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 $b \in \mathbb{R}$. When selecting an appropriate dispersion statistic for a particular application, one must also consider the practical meaning of risk in that setting. We begin with the specification of ($\mathbb{D}$) for the *deviation* statistics.

(a) Deviation measures

In this work we focus on the two deviation measures, *expected excess* (EE) and *absolute semideviation* (ASD). These can be defined as follows:

(i) Expected Excess (EE): Given a target $\eta \in \mathbb{R}$, expected excess (Markert and Schultz, 2005) 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 (3.4) we obtain the objective with excess probability as follows:

$$\underset{x \in X}{\text{Min}} \ \mathbb{E}[f(x, \tilde{\omega})] + \lambda \phi_{EE_\eta}(x). \tag{3.5}$$

(ii) Absolute Semideviation (ASD): This is same as expected excess but with the target value replaced by the mean value $\mathbb{E}[f(x, \tilde{\omega})]$ (Ogryczak and Ruszczynski, 2001) and is given by

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

It reflects the expected value of the excess over the mean value. By setting $\mathbb{D} := \phi_{ASD}$ in (3.4) we obtain the following objective with absolute semideviation:

$$\underset{x \in X}{\text{Min}} \ \mathbb{E}[f(x, \tilde{\omega})] + \lambda \phi_{ASD}(x). \tag{3.6}$$

(b) Quantile Measures

For quantile mean-risk measures, we consider *excess probability* (EP) and *quan-*

tile deviation (QDEV), and *conditional value-at-risk* (CVaR). These risk measures are defined as follows:

(i) Excess Probability (*EP*): Given a target level $\eta \in \mathbb{R}$, excess probability (Schultz and Tiedemann, 2003) 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 (3.4) we obtain the following objective with excess probability:

$$\operatorname*{Min}_{x \in X} \ \mathbb{E}[f(x,\tilde{\omega})] + \lambda \phi_{EP_\eta}(x). \tag{3.7}$$

(ii) Quantile Deviation (*QDEV*): Given $\psi \in (0,1)$, quantile deviation (Ogryczak and Ruszcynski, 2002) is defined as follows:

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

where κ_ψ is the ψ-quantile of the distribution of $f(x,\tilde{\omega})$. Thus the objective (3.4) with $\mathbb{D} := \phi_{QDEV_\psi}$ can now be given as follows:

$$\operatorname*{Min}_{x \in X} \ \mathbb{E}[f(x,\tilde{\omega})] + \lambda \phi_{QDEV_\psi}(x). \tag{3.8}$$

In Ruszcynski and Shapiro (2006) ϕ_{QDEV_ψ} is shown to be equivalent to

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

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

(iii) Conditional Value-at-Risk (ψ-$CVaR$): Given $\psi \in (0, 1)$, the ψ-$CVaR$ (Rockafellar and Uryasev, 2000) can be expressed by the following formula:

$$\phi_{CVaR_\psi}(x) = \operatorname*{Min}_{\eta \in \mathbb{R}} \{\eta + \frac{1}{1-\psi}\mathbb{E}[\max(f(x, \tilde{\omega}) - \eta, 0)]\}.$$

ψ-$CVaR$ reflects the expectation of the $(1 - \psi) \times 100\%$ worst outcomes for a given probability level $\psi \in (0, 1)$. The objective from (3.4) with $\mathbb{D} := \phi_{CVaR_\psi}$ can now be given as

$$\operatorname*{Min}_{x \in X} \ \mathbb{E}[f(x, \tilde{\omega})] + \lambda \phi_{CVaR_\psi}(x). \tag{3.9}$$

Alternatively, ψ-$CVaR$ can be expressed in terms of $\phi_{\varepsilon_1,\varepsilon_2}(x)$ as follows (Ruszcynski and Shapiro, 2006):

$$\phi_{CVaR_\psi}(x) = \mathbb{E}[f(x, \tilde{\omega})] + \frac{1}{\varepsilon_1}\phi_{\varepsilon_1,\varepsilon_2}(x).$$

Further details on the mathematical structures and properties of mean-risk measures are found in Ahmed (2006). Specifically, he states the conditions for the mean-risk objective function to be convexity-preserving are that $\mathbb{D}$ must be convex, non-decreasing, and positively homogenous.

3.3 Mean-Risk Deterministic Equivalent Formulations

Henceforth, we assume that $\tilde{\omega}$ is discrete with finitely many scenarios $\omega \in \Omega$ with corresponding probability $p(\omega)$. Therefore, we can write deterministic equivalent formulations or problems (DEPs) of the stochastic programs with mean-risk measures defined in the previous section. Due to problem size, which grows with the number $|\Omega|$ of scenarios, the problems become too big for direct solvers. This motivates the computational study of decomposition methods to solve these problems. The DEPs for all measures are defined in this section. We begin with expected excess.

(i) Expected Excess (EE)

PROPOSITION 3.3.1. *Given $\lambda \geq 0$ and a target level $\eta \in \mathbb{R}$, problem (3.5) is equivalent to the following formulation (Markert and Schultz, 2005) :*

$$
\begin{aligned}
\textit{EE: Min} \quad & c^\top x + \sum_{\omega \in \Omega} p(\omega) q(\omega)^\top y(\omega) \; + \lambda \sum_{\omega \in \Omega} p(\omega) \nu(\omega) \\
\textit{s.t.} \quad & T(\omega) x + W y(\omega) \geq r(\omega), \;\; \forall \omega \in \Omega \\
& - c^\top x - q(\omega)^\top y(\omega) + \nu(\omega) \geq -\eta, \;\; \forall \omega \in \Omega \\
& x \in X, \quad y(\omega) \in \mathbb{R}_+^{n_2}, \;\; \nu(\omega) \in \mathbb{R}_+, \forall \omega \in \Omega.
\end{aligned}
\tag{3.10}
$$

Problem EE is also an LP with a block angular structure and is suitable for standard SLP decomposition methods such as the L-shaped method (Slyke and Wets (1969). The first-stage variables are x (a vector) and the second-stage variables are vectors $y(\omega)$ and $\nu(\omega)$.

(ii) Absolute Semi-Deviation (ASD)

PROPOSITION 3.3.2. *Given* $\lambda \in [0,1]$, *then problem* (3.6) *is equivalent to the following formulation (Schultz, 2003), and (Ahmed, 2006):*

$$\textit{ASD: Min} \quad (1-\lambda)c^\top x + (1-\lambda)\sum_{\omega\in\Omega} p(\omega)q(\omega)^\top y(\omega) \ + \lambda \sum_{\omega\in\Omega} p(\omega)\nu(\omega) \qquad (3.11a)$$

$$\begin{aligned} s.t. \quad & T(\omega)x + Wy(\omega) \geq r(\omega), \ \forall \omega \in \Omega \\ & -c^\top x - q(\omega)^\top y(\omega) + \nu(\omega) \geq 0, \ \forall \omega \in \Omega \\ & -c^\top x - \sum_{\omega\in\Omega} p(\omega)q(\omega)^\top y(\omega) + \nu(\omega) \geq 0, \ \forall \omega \in \Omega \qquad (3.11b) \\ & x \in X, \quad y(\omega) \in \mathbb{R}^{n_2}_+, \ \nu(\omega) \in \mathbb{R}, \forall \omega \in \Omega. \end{aligned}$$

Unlike the rest of the formulations presented above, problem ASD does *not* have a block angular structure due to constraints (3.11b). Observe that these constraints link all the scenarios. Therefore, standard SLP methods such as the L-shaped method cannot be used to solve this problem. However, a subgradient optimization approach (Ahmed, 2006) can be applied.

(iii) Excess Probability (EP)

PROPOSITION 3.3.3. *Let* $\eta \geq 0$. *If* X *is bounded then there exists a constant* $M > 0$ *such that problem* (3.7) *is equivalent to the following problem (Schultz and*

Tiedemann, 2003):

$$
\begin{aligned}
\textit{EP: Min} \quad & c^\top x + \sum_{\omega \in \Omega} p(\omega) q(\omega)^\top y(\omega) \; + \lambda . \sum_{\omega \in \Omega} p(\omega)\theta(\omega) \\
s.t. \quad & T(\omega)x + Wy(\omega) \geq r(\omega), \; \forall \omega \in \Omega \\
& - c^\top x - q(\omega)^\top y(\omega) + \eta + M.\theta(\omega) \geq 0, \; \forall \omega \in \Omega \\
& x \in X, \; \eta \in \mathbb{R}, \; y(\omega) \in \mathbb{R}_+^{n_2}, \; \theta(\omega) \in \{0,1\}, \forall \omega \in \Omega.
\end{aligned} \tag{3.12}
$$

Observe that EP has a block-angular structure. Consider x and η are the first-stage variables and $y(\omega)$ and $\theta(\omega)$ are the second-stage variables. Then second-stage variables for different ω's never occur in the same constraint, but are linked through the first-stage variables only. However, EP is a mixed-binary linear program due to the introduction of the $\theta(\omega)$ variables. Thus decomposition methods for stochastic integer programming are needed to solve EP. Thus we omit the computational study of EP from this paper.

(iv) Quantile Deviation (QDEV)

PROPOSITION 3.3.4. *Given* $\lambda \in [0, 1/\varepsilon_1]$*, problem* (3.8) *is equivalent to the* *following LP (Ahmed, 2006):*

QDEV:

$$\text{Min } (1-\lambda\varepsilon_1)c^\top x + \lambda\varepsilon_1\eta + (1-\lambda\varepsilon_1)\sum_{\omega\in\Omega} p(\omega)q(\omega)^\top y(\omega) \ + \lambda(\varepsilon_1+\varepsilon_2)\sum_{\omega\in\Omega} p(\omega)\nu(\omega) \tag{3.13a}$$

$$\begin{aligned} s.t.\ \ & T(\omega)x + Wy(\omega) \geq r(\omega),\ \forall\omega\in\Omega \\ & -c^\top x - q(\omega)^\top y(\omega) + \eta + \nu(\omega) \geq 0,\ \forall\omega\in\Omega \\ & x\in X,\ \eta\in\mathbb{R},\ y(\omega)\in\mathbb{R}_+^{n_2},\ \nu(\omega)\in\mathbb{R}, \forall\omega\in\Omega. \end{aligned} \tag{3.13b}$$

Problem QDEV is an LP, has a block angular structure and is amenable to standard decomposition methods for two-stage SLP such as the L-shaped method (Slyke and Wets, 1969). Now x and η are the first-stage variables, and $y(\omega)$ and $\nu(\omega)$ are the second-stage variables.

(v) Conditional Value-at-Risk (ψ-CVaR)

PROPOSITION 3.3.5. *Given $\lambda \geq 0$, problem* (3.9) *is equivalent to the following LP (Ogryczak and Ruszcynski, 2002) :*

$$\text{CVaR: Min} \quad c^\top x + \sum_{\omega\in\Omega} p(\omega)q(\omega)^\top y(\omega) \ + \lambda\eta + \frac{\lambda}{1-\psi}\sum_{\omega\in\Omega} p(\omega)\nu(\omega) \tag{3.14}$$

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

Problem CVaR is also an LP and has a block angular structure. The first-stage variables are x and η, while $y(\omega)$ and $\nu(\omega)$ are the second-stage variables. Problem CVaR can also be solved using standard SLP decomposition methods and is shown for informational purpose here though not used in the computational study.

4. DECOMPOSITION ALGORITHMS

Due to the increasing complexity of the mean-risk stochastic programs as the number of scenarios increases, algorithms used to solve the problems typically require dividing the deterministic equivalent formulation into smaller problems. The formulations that follow are decomposed using stage-wise decomposition. The problem is decomposed into two stages with the noncomplicating variables in the first-stage and the complicating variables in the second-stage. The noncomplicating variables are associated with the here-and-now decisions that are made before the realization of uncertain problem parameters (represented by random variables). The complicating variables are different for every scenario subproblem and aid in modeling a problem's uncertainty, which is represented by random variables.

4.1 Aggregated Cut Subgradient-Based Algorithm

As previously stated, the mean-ASD formulation does not have the block angular structure due to the scenario linking constraints (3.11b). Thus a subgradient optimization approach is used to solve this problem. The stage-wise decomposition is tested with algorithms that use separate cuts based on (Ahmed (2006)) and a aggregate cut (new extension) to approximate the $\mathbb{E}[f(x, \tilde{\omega})]$ and ASD.

The Aggregated Cut Subgradient-Based Algorithm (ASD-AGG) is described first. Let k denote the current algorithm iteration index and η be the variable approximating the objective value of the master program. Then the master program for the ASD-AGG Algorithm can be represented as follows:

Master Program

$$
\begin{aligned}
l^k := \text{Min} \quad & \eta \\
\text{s.t.} \quad & Ax \geq b \\
& (\hat{\beta}^t)^\top x + \eta \geq \hat{\alpha}^t, \ t \in \mathcal{T}_k && \text{(4.1a)} \\
& (\hat{\beta}^t)^\top x \geq \hat{\alpha}^t, \ t \notin \mathcal{T}_k && \text{(4.1b)} \\
& x \geq 0, \eta \text{ free}
\end{aligned}
$$

where $\mathcal{T}_k$ is the set of iteration indices at which optimality cut (4.1a) is generated. The feasibility cut (4.1b) is generated instead of the optimality cuts when any scenario subproblem is infeasible.

Let the optimal solution to the master program at iteration k be x^k and η^k, and the corresponding optimal value be l^k. Then the second-stage subproblem is given as follows:

Second-Stage Subproblem

$$
\begin{aligned}
Q(x^k, \omega) := c^\top x^k + \underset{y(\omega)}{\text{Min}} \quad & q(\omega)^\top y(\omega) && \text{(4.2)} \\
\text{s.t.} \quad & W(\omega)y(\omega) \geq r(\omega) - T(\omega)x^k \quad \leftarrow \pi^k(\omega) \\
& y(\omega) \geq 0.
\end{aligned}
$$

where $\pi^k(\omega)$ is the vector of dual solutions for the corresponding set of constraints. Now the algorithm can be stated.

Algorithm 1: ASD-AGG

1. Initialization
   (a) Select relative tolerance $\epsilon \geq 0$.
   (b) Set iteration counter $k \leftarrow 0$, $LB \leftarrow -\infty$, and $UB \leftarrow \infty, \mathcal{T}_0 \leftarrow \emptyset$.
   (c) $\lambda \leftarrow$ specific value in [0,1]
   (d) Obtain $x^0 = argmin\{(1-\lambda)c^\top x | Ax = b, x \geq 0\}$
   (e) Setup master program (4.1)
2. For $\omega \in \Omega$ solve the second-stage subproblem (4.2)
   (a) If any subproblem is infeasible, generate the feasibility cut:
      i. Get the dual extreme ray, $\gamma^k(\omega)$
      ii. Compute $\hat{\alpha}^k = (\gamma^k(\omega))^\top r(\omega)$ and $\hat{\beta}^k = (\gamma^k(\omega))^\top T(\omega)$ and continue to Step 6.
   (b) Otherwise, the subproblems are feasible $\forall \omega$, generate the optimality cut:
      i. Get the dual solution, $\pi^k(\omega)$.
      ii. Compute $\mu(x^k) = \mathbb{E}[f(x^k, \tilde{\omega})] = c^\top x^k + \sum_{\omega \in \Omega} p(\omega)[q(\omega)^\top y(\omega)^k]$, and a subgradient for the cost function $\zeta(\omega)^k = \partial f(x^k, \omega) = c^\top - (\pi^k(\omega))^\top T(\omega)$, and $u^k = \mathbb{E}[\zeta(\omega)^k] = c^\top - \sum_{\omega \in \Omega} p(\omega)(\pi^k(\omega))^\top T(\omega)$.
      iii. Compute $\nu(x^k) = \mathbb{E}[\max\{f(x^k, \omega), \mu(x^k)\}]$ and the subgradient $v^k = \sum_{\omega \in \Omega} p(\omega)[\iota(\omega)\zeta(\omega)^k + (1 - \iota(\omega))u^k]$, where $\iota(\omega) = 1$ if $f(x^k, \omega) > \mu(x^k)$ and 0 otherwise.
      iv. Compute $\alpha^k = \sum_{\omega \in \Omega} p(\omega)\pi(\omega)^\top r(\omega)$, $\beta^k = -u^k, \sigma^k = -v^k$, and $\hat{\theta}^k = \sum_{\omega \in \Omega} p(\omega)[\iota(\omega)\pi(\omega)^\top r(\omega) + (1 - \iota(\omega))\alpha^k]$.

v. Calculate coefficients for the aggregate cut: $\hat{\beta}^k = (1-\lambda)\beta^k + \lambda\sigma^k$ and $\hat{\alpha}^k = (1-\lambda)\alpha^k + \lambda\hat{\theta}^k$.

3. Compute upperbound UB $= \min\{(1-\lambda)\mu(x^k) + \lambda\nu(x^k), UB\}$.
4. If the UB is updated, set the incumbent solution x^* to x^k
5. Check termination criteria
   (a) If UB-LB $\leq \epsilon$|UB|, STOP and report optimal solution x^*.
   (b) Otherwise, continue.
6. Update and solve the master program.
   (a) If a subproblem was infeasible, add $(\hat{\beta}^k)^\top x^k \geq \hat{\alpha}^k$ to master program (4.1).
   (b) Otherwise, add $(\hat{\beta}^k)^\top x^k + \eta \geq \hat{\alpha}^k$ and k to $\mathcal{T}_k$ to master program (4.1).
   (c) Solve the master program to obtain its optimal value l^{k+1} and solution x^{k+1}.
7. Compute the lowerbound LB $= \max\{l^{k+1}, LB\}$
8. Set $k \leftarrow k+1$ and return to Step 2.

4.2 Separate Cut Subgradient-Based Algorithm

Next, we describe the Separate Cut Subgradient-Based Algorithm (ASD-SEP). Let k denote the current algorithm iteration index, θ be the variable approximating the $\mathbb{E}[f(x,\tilde{\omega})]$, and η be the variable approximating ASD. Then the ASD-SEP master program is:

Master Program

$$
\begin{aligned}
l^k := \text{Min} \quad & (1-\lambda)\theta + \lambda\eta \\
\text{s.t.} \quad & Ax \geq b \\
& (\beta^t)^\top x + \eta \geq \alpha^t, \; t \in \mathcal{T}_k && \text{(4.3a)} \\
& (\hat{\beta}^t)^\top x + \eta \geq \hat{\alpha}^t, \; t \in \mathcal{T}_k && \text{(4.3b)} \\
& (\beta^t)^\top x \geq \alpha^t, \;\; t \notin \mathcal{T}_k && \text{(4.3c)} \\
& x \geq 0, \eta, \; \theta \text{ free}
\end{aligned}
$$

where $\mathcal{T}_k$ is the set of iteration indices at which optimality cuts (4.3a, 4.3b) are generated and constraint (4.3c) is the feasibility cut.

Let the optimal solution to the master program at iteration k be x^k, η^k, and θ^k its corresponding optimal value is l^k. Then the second-stage subproblem is given as follows:

Second-Stage Subproblem

$$
\begin{aligned}
f(x^k, \omega) := c^\top x^k + \underset{y(\omega)}{\text{Min}} \quad & q(\omega)^\top y(\omega) && \text{(4.4)} \\
\text{s.t.} \quad & W(\omega)y(\omega) \geq r(\omega) - T(\omega)x^k \quad \leftarrow \pi^k(\omega) \\
& y(\omega) \geq 0
\end{aligned}
$$

where $\pi^k(\omega)$ is the vector of dual solutions for the corresponding set of constraints. Now the ASD-SEP Algorithm (based on Ahmed (2006)) can be stated.

Algorithm 2: ASD-SEP

1. Initialization
   (a) Select relative tolerance $\epsilon \geq 0$.
   (b) Set iteration counter $k \leftarrow 0$, $LB \leftarrow -\infty$, and $UB \leftarrow \infty$.
   (c) $\lambda \leftarrow$ specific value in [0,1]
   (d) Obtain $x^0 = argmin\{(1-\lambda)c^\top x | Ax = b, x \geq 0\}$
   (e) Setup master program (4.3)
2. For $\omega \in \Omega$ solve the second-stage subproblem (4.4)
   (a) If any subproblem is infeasible, generate the feasibility cut:
      i. Get the dual extreme ray, $\gamma^k(\omega)$
      ii. Compute $\alpha^k = (\gamma^k(\omega))^\top r(\omega)$ and $\beta^k = (\gamma^k(\omega))^\top T(\omega)$ and continue to Step 6.
   (b) Otherwise, the subproblems are feasible $\forall \omega$, generate the optimality cut:
      i. Get the dual solution, $\pi^k(\omega)$.
      ii. Compute $\mu(x^k) = \mathbb{E}[f(x^k, \tilde{\omega})]$
      $= c^\top x^k + \sum_{\omega \in \Omega} p(\omega)[q(\omega)^\top y(\omega)^k]$,
      the cost function subgradient
      $\zeta(\omega)^k = \partial f(x^k, \omega) = c^\top - (\pi^k(\omega))^\top T(\omega)$,
      and $u^k = \mathbb{E}[\zeta(\omega)^k] = c^\top - \sum_{\omega \in \Omega} p(\omega)(\pi^k(\omega))^\top T(\omega)$.
      iii. Compute $\nu(x^k) = \mathbb{E}[\max\{f(x^k, \omega), \mu(x^k)\}]$ and the subgradient $v^k = \sum_{\omega \in \Omega} p(\omega)[\iota(\omega)\zeta(\omega)^k + (1 - \iota(\omega))u^k]$, where $\iota(\omega) = 1$ if

$f(x^k, \omega) > \mu(x^k)$ and 0 otherwise.

iv. Compute $\alpha^k = \sum_{\omega \in \Omega} p(\omega)\pi(\omega)^\top r(\omega)$, $\beta^k = -u^k, \sigma^k = -v^k$, and $\hat{\alpha}^k = \sum_{\omega \in \Omega} p(\omega)[\iota(\omega)\pi(\omega)^\top r(\omega) + (1 - \iota(\omega))\alpha^k]$

3. Compute upperbound UB $= \min\{(1-\lambda)\mu(x^k) + \lambda\nu(x^k), UB\}$.
4. If the UB is updated, set the incumbent solution x^* to x^k.
5. Check termination criteria
   (a) If UB-LB $\leq \epsilon|$UB$|$, STOP and report optimal solution $x^*, \mu(x^*)$, and $\nu(x^*)$.
   (b) Otherwise continue.
6. Update the master program and solve.
   (a) If a subproblem was infeasible, add $(\hat{\beta}^k)^\top\ x^k \geq \hat{\alpha}^k$.
   (b) Otherwise, add $(\beta^k)^\top\ x^k + \eta \geq \alpha^k$, $(\hat{\beta}^k)^\top\ x^k + \eta \geq \hat{\alpha}^k$, and k to $\mathcal{T}_k$.
   (c) Solve the master program to obtain its optimal value l^{k+1} and solution $(x^{k+1}, \theta^{k+1}, \eta^{k+1})$.
7. Compute the lowerbound LB $= \max\{l^k, LB\}$
8. Set $k \leftarrow k+1$ and return to Step 2.

The difference in the two algorithms above is in the number of cuts used in the master program to approximate the objective function. The ASD-SEP Algorithm uses a distinct cut to approximate the $\mathbb{E}[f(x,\tilde{\omega})]$ and ASD statistic separately. The ASD-AGG Algorithm aggregates the cuts from the ASD-SEP Algorithm into one cut by weighting the coefficients appropriately based on the risk weight λ.

4.3 L-Shaped Algorithm for Quantile Deviation

First, the L-Shaped Algorithm for Quantile Deviation (QDEV-AGG) is described (a direct use of the L-shaped Algorithm) by introducing its master (first-stage) program and scenario (second-stage) subproblems. Then the algorithm, which uses a single optimality cut, is explicitly stated in the next section.

Master program

$$\begin{aligned} l^k := \text{Min} \quad & (1-\lambda\varepsilon_1)c^\top x + \lambda\varepsilon_1 z + \eta \\ \text{s.t.} \quad & Ax \geq b \\ & (\beta^t)^\top \ x + \sigma^t z + \eta \geq \alpha^t, \ t \in \mathcal{T}_k && \text{(4.5a)} \\ & (\beta^t)^\top \ x + \sigma^t z \geq \alpha^t, \ t \notin \mathcal{T}_k && \text{(4.5b)} \\ & x \geq 0, \delta_l \leq z, \delta_h \geq z, \eta \text{ free} \end{aligned}$$

In program (4.5) $\mathcal{T}_k$ is the set of iteration indices at which the optimality cut (4.5a) is generated, and δ_l and δ_h are a fixed upper and lower bounds for the z variable. The variable η approximates the second-stage subproblem $\forall\omega \in \Omega$ scenarios. The feasibility cut (4.5b) is generated instead of the optimality cut when all the scenario subproblems are not feasible.

Let the optimal solution to the master program at iteration k be x^k, z^k, and η^k, and the corresponding optimal value be l^k. Then the second-stage subproblem is given as follows:

Second-Stage subproblem

$$
\begin{aligned}
Q(x^k,\omega) := \text{Min} \quad & (1-\lambda\varepsilon_1)q(\omega)^\top y(\omega) + (\lambda\varepsilon_1+\varepsilon_2)\nu(\omega) \\
\text{s.t.} \quad & W(\omega)y(\omega) \geq r(\omega) - T(\omega)x^k \quad \leftarrow \pi^k(\omega) \\
& -q(\omega)^\top y(\omega) + \nu(\omega) \geq c^\top x^k - z^k \quad \leftarrow \acute{\pi}^k(\omega) \\
& y(\omega) \geq 0 \\
& \nu(\omega) \geq 0
\end{aligned} \tag{4.6}
$$

where $\pi^k(\omega)$ and $\acute{\pi}^k(\omega)$ are the dual variables for the corresponding constraints. The QDEV-AGG Algorithm is stated as follows.

Algorithm 3: QDEV-AGG

---

1. Initialization
   (a) Select relative tolerance $\epsilon \geq 0$, and appropriately select δ_l and δ_h to set the range of target value z.
   (b) Set iteration counter $k \leftarrow 0$, lowerbound $LB \leftarrow -\infty$, and upperbound $UB \leftarrow \infty$.
   (c) Select $\lambda \in [0, \frac{1}{\varepsilon_1}]$, $\psi \in (0,1)$, and $\varepsilon_1, \varepsilon_2 > 0$ such that $\psi = \varepsilon_2/(\varepsilon_1+\varepsilon_2)$.
   (d) Obtain $x^0, z^0 = argmin\{c^\top x + z | Ax \geq b, x \geq 0, \delta_l \leq z^k \leq \delta_h\}$.
   (e) Setup master program (4.5).
2. For $\omega \in \Omega$ solve the second-stage subproblem (4.6)
   (a) If any subproblem is infeasible, generate a feasibility cut.

i. Get the dual extreme rays, $\gamma^k(\omega)$ and $\hat{\gamma}^k(\omega)$ for the current infeasible instance.

ii. Compute $\alpha^k = (\gamma^k(\omega))^\top r(\omega)$ and $\beta^k = (\gamma^k(\omega))^\top T(\omega) - \hat{\gamma}^k(\omega) * c^\top$ and $\sigma^k = \hat{\gamma}^k(\omega)$ and continue to Step 6.

(b) Otherwise, the subproblems are feasible $\forall\omega$, generate the optimality cut.

i. Get the dual solutions, $\pi^k(\omega)$ and $\acute{\pi}^k(\omega)$.

ii. Compute cut coefficients: $\alpha^k = (\pi^k(\omega))^\top r(\omega)$,
$\beta^k = (\pi^k(\omega))^\top T(\omega) - \acute{\pi}^k(\omega) * c^\top$, and $\sigma^k = \acute{\pi}^k(\omega)$.

iii. Compute $\mu(x^k, y^k(\omega)) = c^\top x^k + \sum_{\omega\in\Omega} p(\omega)q(\omega)y^k(\omega)$
and $\nu^k = \sum_{\omega\in\Omega} p(\omega)\nu(\omega)$.

3. Calculate $m^k = (1-\lambda\varepsilon_1)c^\top x^k + \lambda\varepsilon_1 z^k + \lambda(\varepsilon_1+\varepsilon_2)\nu^k$.

4. $UB = \min\{m^k, UB\}$. If the UB is updated set the incumbent solution x^* to x^k, z^* to z^k, ν^* to ν^k, and μ^* to μ^k.

5. Check termination criteria.

(a) If UB-LB $\leq \epsilon|$UB$|$, STOP and report optimal solution x^*, z^*, μ^*, ν^*.

(b) Otherwise, continue.

6. Update master program and solve.

(a) If a subproblem was infeasible add $(\beta^k)^\top\ x^k + \sigma^k z^k \geq \alpha^k$ to master program (4.5).

(b) Otherwise, add $(\beta^k)^\top\ x^k + \sigma^k z^k + \eta \geq \alpha^k$ and k to $\mathcal{T}_k$ to master program (4.5).

(c) Solve master program to obtain optimal value l^{k+1} and solution $(x^{k+1}, z^{k+1}, \eta^{k+1})$

7. Compute the LB $= \max\{l^{k+1}, LB\}$.

8. Set $k \leftarrow k+1$ and return to Step 2.

4.4 Separate Cut L-shaped Algorithm for Quantile Deviation

We will now move to solving mean-QDEV SLP using the separate cut version of the L-shaped method based on Ahmed (2006). The Separate Cut L-Shaped Algorithm for Quantile Deviation (QDEV-SEP), uses a separate cut for both the $\mathbb{E}[f(x,\tilde{\omega})]$ and QDEV statistics. Let k denote the current algorithm iteration index, θ be the variable approximating the $\mathbb{E}[f(x,\tilde{\omega})]$, and η be the variable approximating QDEV measure. Then the QDEV-SEP master program is:

Master program

$$\begin{aligned}
l^k := \text{Min} \quad & \lambda\varepsilon_1 z + (1-\lambda\varepsilon_1)\theta + (\lambda\varepsilon_1 + \varepsilon_2)\eta \\
\text{s.t.} \quad & Ax \geq b \\
& (\beta^t)^\top x + \theta \geq \alpha^t, \; t \in \mathcal{T}_k && (4.7a) \\
& (\hat{\beta}^t)^\top x + \sigma^t z + \eta \geq \hat{\alpha}^t, \; t \in \mathcal{T}_k && (4.7b) \\
& (\beta^t)^\top x \geq \alpha^t, \; t \notin \mathcal{T}_k && (4.7c) \\
& x \geq 0, \delta_l \leq z \leq \delta_h, \theta, \eta \text{ free}
\end{aligned}$$

where $\mathcal{T}_k$ is the iteration index at which the optimality cuts (4.7a, 4.7b) are generated and δ_l and δ_h are a fixed upper and lower bounds for the z variable.

Let the optimal solution to the master program at iteration k be x^k, z^k, η^k, and θ^k its corresponding optimal value l^k. Then the second-stage subproblem is given as follows:

Second-Stage subproblem

$$\begin{aligned} Q(x^k,\omega) := c^\top x^k + \text{Min} \quad & q(\omega)^\top y(\omega) \\ \text{s.t.} \quad & W(\omega)y(\omega) \geq r(\omega) - T(\omega)x^k \quad \leftarrow \pi^k(\omega) \\ & y(\omega) \geq 0 \end{aligned} \tag{4.8}$$

where $\pi^k(\omega)$ is the vector of dual solutions for the corresponding set of constraints. The QDEV-SEP Algorithm can now be stated.

Algorithm 4: QDEV-SEP

---

1. Initialization
   (a) Select relative tolerance $\epsilon \geq 0$, and δ_l and δ_h to set the range of target value z.
   (b) Set iteration counter $k \leftarrow 0$, lowerbound $LB \leftarrow -\infty$, and upperbound $UB \leftarrow \infty$.
   (c) Select $\lambda \in [0, \frac{1}{\varepsilon_1}]$, $\psi \in (0,1)$, and ε_1, $\varepsilon_2 > 0$ such that $\psi = \varepsilon_2/(\varepsilon_1+\varepsilon_2)$.
   (d) Obtain $x^0, z^0 = argmin\{c^\top x + z | Ax \geq b, x \geq 0, \delta_l \leq z^k \leq \delta_h\}$.
   (e) Setup master program (4.7).

2. For $\omega \in \Omega$ solve the second-stage subproblem (4.8).

   (a) If any subproblem is infeasible, generate a feasibility cut.

      i. Get the dual extreme ray, γ^k of the subproblem that is infeasible.

      ii. Compute $\alpha^k = (\gamma^k)^\top r(\omega)$ and $\beta^k = (\gamma^k)^\top T(\omega)$ and continue to Step 6.

   (b) Otherwise the subproblems are feasible $\forall\omega$, generate the optimality cuts.

      i. Get the dual solutions, $\pi^k(\omega)$.

      ii. Compute $\mu(x^k) = \mathbb{E}[f(x, \tilde{\omega})]$, the cost function subgradient $\zeta(\omega)^k = \partial f(x^k, \omega) = c^\top - (\pi^k(\omega))^\top T(\omega)$, and $u^k = \mathbb{E}[\zeta(\omega)^k] = p(\omega)[c^\top - (\pi^k(\omega))^\top T(\omega)]$

      iii. Compute $\nu(x^k, z^k) = \mathbb{E}[\max\{f(x^k, \omega) - z^k, 0\}]$, and the subgradient with respect to x^k :
      $v_x^k = p(\omega)[c^\top - (\pi^k(\omega))^\top T(\omega)]\iota(\omega)$, and a subgradient with respect to z^k : $v_z^k = -p(\omega)\iota(\omega)$, where $\iota(\omega) = 1$, if $f(x^k, \omega) > z^k$ and 0 otherwise.

      iv. Compute $\alpha^k = \sum_{\omega\in\Omega} p(\omega)\pi(\omega)^\top r(\omega)$, $(\beta^k)^\top = u^k$,
      $(\hat{\beta}^t)^\top = -v_x^k$, $\sigma^k = -v_z^k$, and
      $\hat{\alpha}^k = \sum_{\omega\in\Omega} p(\omega)[\iota(\omega)\pi(\omega)^\top r(\omega)]$.

3. Calculate $m^k = (1 - \lambda\varepsilon_1)\mu(x^k) + \lambda\varepsilon_1 z^k + \lambda(\varepsilon_1 + \varepsilon_2)\nu^k(x^k, z^k)$.

4. $UB = \min\{m^k, UB\}$, if the UB is updated set the incumbent solution x^* to x^k, z^* to z^k, ν^* to ν^k, and μ^* to μ^k.

5. Check termination criteria.

(a) If UB-LB $\leq \epsilon|\text{UB}|$, STOP and report optimal solution x^*, z^*, μ^*, ν^*.

(b) Otherwise, continue.

6. Update master program and solve.

   (a) If a subproblem was infeasible, add $(\beta^k)^\top x^k \geq \alpha^k$ to master program (4.7).

   (b) Otherwise, add $(\beta^k)^\top x^k + \sigma^k z^k + \eta \geq \alpha^k$ and k to $\mathcal{T}_k$ to master program (4.7).

   (c) Solve the master program to obtain optimal value l^{k+1} and solution $(x^{k+1}, z^{k+1}, \theta^{k+1}, \eta^{k+1})$.

7. Compute the LB $= \max\{l^k, LB\}$.

8. Set $k \leftarrow k+1$ and return to Step 2.

Note: In our implementation of the QDEV-AGG and QDEV-SEP algorithms, the quantile value ψ was set to 0.5 to allow accurate comparison with the algorithms for ASD that have a target value of $\mathbb{E}[f(x,\tilde{\omega})]$. In addition the upper and lower bounds of the target z, δ_l and δ_h were set to fixed values that are shown in Section A.2 of the appendix.

5. COMPUTATIONAL RESULTS

5.1 Design of Experiments

In this section we describe the test instances used in the computational experiments and then report computational results. Recall the objectives of this computational study were to investigate: (a) computational performance of ASD-AGG, ASD-SEP, QDEV-AGG, and QDEV-SEP algorithms; (b) effect of using of ASD and QDEV; (c) conditions that justify the use of the risk-averse approach over the risk-neutral one. To evaluate the performance of the algorithms, the following performance parameters were used: *computational* (CPU) *time*, *objective value* at termination, and number of *algorithm iterations*. Runtime was measured in seconds.

The algorithms were implemented in C++ using the IBM CPLEX Callable Library version 12.0 (IBM, 2009) 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 *LPobjectClass*, except *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.

To change the level of risk, the risk parameter λ was varied from values of 0 to 1 in increments of 0.1 for each instance. This allowed us to approximate or trace the efficient frontier of λ for each instance. To analyze the impact of problem size in the study, instances with low (_l), medium (_m), and high (_h) number of scenarios were created from the extremely large instances that had too many scenarios to solve . Since the probability distribution (in the STOCH file) of most of the original test instances was uniform, a nonuniform distribution was created for some instances to see the effect of the risk-neutral and risk-averse methods on the instance solution. This was done by appropriately changing the value of the probabilities in the original STOCH file. Next we describe the test instances.

5.2 Test Instances

The instances used in the study are found in Higle and Sen (1996) and Linderoth et al. (2006) and can be accessed from http://pages.cs.wisc.edu/∼swright/stochastic/sampling/. These seven well-known SLP instances with fixed recourse are as follows: *cep1* (Higle and Sen, 1996), *pgp2* (Higle and Sen, 1996), *gbd* (Dantzig, 1963), *LandS* (Louveaux and Smeers, 1998), *20term* (Mak et al., 1999), *ssn* (Sen et al., 1994), and *storm* (Mulvey and Ruszcynski, 1995). Instance *cep1* is a capacity expansion planning problem, while *LandS* is a modification of a simple problem in electrical investment planning. Instance *pgp2* considers a problem in power generation planning, and *ssn* is a problem concerning telecommunications network design. Finally, *storm* is a problem used by the US Military to plan the allocation of aircraft routes during the Gulf War of 1991. More detailed information about the objective and stochastic elements of each instance are found later in the section.

Instance *cep1* is a two-stage machine capacity 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 i.i.d. random variables coming from a known distribution. The problem's objective is to minimize weekly amortized expansion cost (per hour) plus the expected weekly production costs.

Problem *pgp2* is an electrical capacity expansion problem that seeks to select the minimum cost strategy for investing in electricity generated from gas-fired, coal-fired, and nuclear generators. This strategy is modeled with a two-stage stochastic program. The first-stage variables model the annualized capital cost ($/kw) based on the specific type of generator acquired. 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 program's variables select the number of aircraft (from the four types) assigned to a particular route, while its constraints bound the availability of the aircrafts. The second-stage program's variables indicate the number of carried and bumped passengers on each of the five routes and its constraints balance demand for the routes. The demand, which is the righthand side (RHS) values of second-stage balance constraints, is stochastic.

LandS models an electrical investment planning problem. 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 3 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. The random demand follows the marginal distribution of 0.04(k-1), where k = 1, 2,..., 100 for the three RHS values of constraints.

Problem *20term* models operations of motor freight carriers. 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.

Instance *ssn* is a budget-constrained telephone-switching network expansion problem. The objective of this model 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 freight scheduling problem modeled over two periods that plans flights over a set of network routes with uncertain amounts of cargo. Flight routes are scheduled at 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 occurred, allocates the cargo delivery routes and seeks to satisfy any unmet demand while minimizing holding and penalty cost. The RHS realizations of the demand were distributed uniformly at a ± 20% interval from the corresponding values in the core (.cor) file.

Because *20term*, *ssn*, and *storm* have a large number of scenarios smaller or truncated versions of these instances were created. Smaller sets of the total scenarios were used that had larger differences in the span between the smallest and largest RHS values for a particular row in the STOCH file. This aided in selecting rows for low, medium, and high scenario sizes. For example the truncated storm instances were selected by choosing rows with greater than 250, 200, and 150 difference between smallest and largest values of random RHS for the low (_l), medium (_m), and high (_h) size instances respectively. Preliminary runs were done with the ASD-AGG Algorithm to choose scenarios sizes that would allow the instances to run in less than approximately six hours. The scenario size information is shown in Table 5.1.

Table 5.1: Truncated instances

Name	Scenarios	First-Stage (Const., Var.)	Second-Stage (Const., Var.)
20Tr_l	512	(3, 64)	(4, 17)
20Tr_m	1024	(3, 64)	(4,17)
20Tr_h	2048	(3, 64)	(4,17)
stormTr_l	625	(185, 121)	(528, 1259)
stormTr_m	15625	(185, 121)	(528, 1259)
stormTr_h	390625	(185, 121)	(528, 1259)
ssnTr_l	735	(1, 89)	(175, 706)
ssnTr_m	5145	(1, 89)	(175, 706)
ssnTr_h	36015	(1, 89)	(175, 706)

The size and description of all instances, including the nonuniform instances, are shown in Table 5.2. Instances *cep1*, *pgp2*, *gbd*, and *20term* (20Tr_l, 20Tr_m) were selected to create nonuniform instances because of their fairly uniform stochastic files, nonexistent to small changes in $\mathbb{E}[f(x,\tilde{\omega})]$ and first-stage solutions across the tested values of $\lambda \in [0, 1]$, or both. This was done by using trial and error to appropriately change the value of the marginal probabilities in the STOCH file to yield less uniform distributions. By comparing results of instances with nonuniform and uniform distributions the impact of the risk-neutral and mean-risk approaches could be further studied. The nonuniform instances are cep1a and cep1sk for instance *cep1*, pgp2e and pgp2f for *pgp2*, gbd_sk3 for *gbd*, 20Tr_lsk1 for 20Tr_l, and 20Tr_msk1 and 20Tr_msk2 for 20Tr_m.

Table 5.2: Test instance sizes

Name	Application	Scenarios	First-Stage (Const., Var.)	Second-Stage (Const., Var.)
cep1	Capacity Expansion Planning	216	(9, 8)	(7, 15)
cep1a		216	(9, 8)	(7, 15)
cep1sk		216	(9, 8)	(7, 15)
pgp2	Power Generation Planning	576	(2, 4)	(7, 12)
pgp2e		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 Design	10^{70}	(1, 89)	(175, 706)
ssnTr_l		735	(1, 89)	(175, 706)
ssnTr_m		5145	(1, 89)	(175, 706)
ssnTr_h		36015	(1, 89)	(175, 706)
storm	Cargo Flight Scheduling	6×10^{81}	(185, 121)	(528, 1259)
stormTr_l		625	(185, 121)	(528, 1259)
stormTr_m		15625	(185, 121)	(528, 1259)
stormTr_h		390625	(185, 121)	(528, 1259)

5.3 Results

We present study results that are organized based on the study objectives. First, computational performance of the ASD-AGG, ASD-SEP, QDEV-AGG, and QDEV-SEP algorithms are described. Then the impact problem size (based on increasing scenario size) has on performance parameters of iterations and CPU time is investigated. Next we report optimal instance objective values over the different risk levels. This leads to a contrast of the robustness of using ASD and QDEV as risk measures in the mean-risk models. Finally, the comparison of risk-averse and risk-neutral approaches and the conditions to justify use of one approach over the other are discussed.

The instances described in Table 5.2 were used as the the starting point for obtaining results. Next, a computer program was created in C++ using CPLEX Callable library that read in the independent formatted .COR (core), .STO (stochastic), and .TIM (time) files for each of the instances to create the appropriately decomposed Mean-ASD and Mean-QDEV SLPs. Then the program was adapted to solve the decomposed master program and subproblems using the subgradient cutting plane algorithms from Chapter 4. These programs were then solved with CPLEX 12.0 for the risk tradeoff parameter $\lambda = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ for each instance. This allowed us to approximate the efficient frontier of λ for each instance. For each run we recorded the number of algorithm iterations, computation time (CPU) in seconds (sec). We then calculated the average (AVG) 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 $\psi = 0.5$ in the QDEV model by having $\varepsilon_1 = \varepsilon_2 = 1$. Recall that $\psi = \varepsilon_2/(\varepsilon_1 + \varepsilon_2)$ and $\varepsilon_1, \varepsilon_2 > 0$. This allowed a fair comparison between the ASD and QDEV approaches.

Throughout this section the unweighted mean will equal the value of

$$\mathbb{E}[f(x,\tilde{\omega})] = c^\top x + \sum_{\omega\in\Omega} p_\omega q_\omega^\top y_\omega.$$

It gives no weight to the risk measure and the objective value is based solely on the expected value solution. This is the unweighted version of the first and second terms of the objective function for Mean-ASD SLP (3.11a) and first and third terms of (3.13b) for Mean-QDEV SLP. Conversely, the first and second terms of the objective function (3.11a) form the weighted mean (Wt. Mean), while the second term is defined as the weighted ASD (Wt. ASD) in the Mean-ASD SLP. Similarly, the weighted mean (Wt. Mean) is defined by the first and third terms of the objective function(3.13b), while the second and last term is defined as the weighted QDEV (Wt. QDEV)in the Mean-QDEV SLP. These weighted terms are scaled based on the tradeoff parameter λ.

Over each risk level, the value of $\mathbb{E}[f(x,\tilde{\omega})]$ and its corresponding first-stage decision vector x are important for understanding the following results. Their final values are important because changes in these variables along with corresponding changes in the objective value indicate a change in decisions and allocation of resources to optimally solve each instance. Thus larger changes in the $\mathbb{E}[f(x,\tilde{\omega})]$, first-stage decision variables, and objective value will indicate a stronger case for the risk-averse approach. The following paragraphs, tables, and figures describe key findings from the study.

Analysis of the decomposition algorithms

Table 5.3 compares the ASD-AGG and ASD-SEP algorithms in terms of average (AVG) and standard deviation (STDEV) (computed based on optimal values for

λ from 0.0 to 1.0) of iterations and CPU time for algorithm termination. ASD-AGG algorithm takes more iterations on average than ASD-SEP algorithm. The ASD-AGG Algorithm takes shorter CPU time on average for all storm truncated (stormTr), cep1sk, 20Tr_l and ssnTr_h instances. The ASD-SEP Algorithm takes less CPU time on average for cep1, cep1a, 20 truncated instances except 20Tr_l, ssn truncated instances except ssnTr_h, and all pgp2 and gbd instances. Based on the average and standard deviation of iterations and CPU time, neither version of the ASD algorithm dominates the other.

Table 5.3: Performance of ASD-AGG and ASD-SEP algorithms

		ASD-AGG				**ASD-SEP**			
		Iters		CPU(s)		Iters		CPU(s)	
Instance	Scenarios	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev
cep1	216	2.00	0.00	0.02	0.01	2.00	0.00	0.02	0.00
cep1a	216	2.00	0.00	0.03	0.01	2.00	0.00	0.02	0.00
cep1sk	216	6.00	0.00	0.04	0.03	6.00	0.00	0.05	0.02
pgp2	576	33.91	2.63	0.57	0.08	31.45	2.30	0.55	0.09
pgp2e	576	37.91	2.63	0.52	0.07	35.36	2.50	0.51	0.06
pgp2f	576	37.64	3.29	0.52	0.10	33.09	3.83	0.45	0.07
gbd	6.5×10^5	33.55	3.72	258.18	30.52	31.55	2.02	242.43	24.18
gbd_sk3	6.5×10^5	27.09	7.37	186.67	51.26	22.73	5.00	154.02	33.74
LandS	10^5	31.00	1.61	389.40	18.61	30.18	1.25	382.81	32.09
20Tr_l	512	2250.91	332.88	1227.78	217.13	2185.73	572.10	1350.58	507.60
20Tr_lsk1	512	1738.00	152.01	972.01	108.19	1587.00	166.80	914.05	112.60
20Tr_m	1024	1912.09	316.13	1784.00	318.54	1804.36	233.99	1739.86	244.41
20Tr_msk1	1024	2108.00	217.13	2310.71	287.26	1783.18	246.02	1995.83	382.34
20Tr_msk2	1024	1865.55	238.40	1861.99	297.92	1676.91	238.75	1738.45	321.48
20Tr_h	2048	1757.00	106.73	3088.27	218.56	1505.45	165.88	2631.08	273.72
ssnTr_l	735	352.91	149.69	90.51	38.57	317.09	151.55	86.45	41.47
ssnTr_m	5145	983.45	1339.86	1647.08	2318.09	890.73	990.84	1491.98	1699.64
ssnTr_h	36015	788.40	300.45	6929.49	2620.53	1013.55	423.16	8876.52	3665.14
stormTr_l	625	16.00	0.00	2.36	0.09	16.00	0.00	2.42	0.09
stormTr_m	15625	13.00	0.00	47.68	0.35	13.00	0.00	48.25	0.47
stormTr_h	390625	13.00	0.00	1193.81	5.32	13.00	0.00	1195.29	10.24

Table 5.4 summarizes the performance of the QDEV-AGG and QDEV-SEP algorithms in terms of its average (AVG) and standard deviation (STDEV)(computed

based on optimal values for λ from 0.0 to 1.0) of iterations and CPU time. QDEV-SEP Algorithm takes less iterations and CPU time on average than the QDEV-AGG Algorithm. With regard to CPU time, QDEV-SEP Algorithm dominates QDEV-AGG Algorithm based on average and standard deviation for LandS, and all storm and gbd instances. Based on the average and standard deviation of the number of iterations it takes for the algorithm to terminate, QDEV-SEP Algorithm does not dominate QDEV-AGG Algorithm.

Table 5.4: Performance of QDEV-AGG and QDEV-SEP algorithms

		QDEV-AGG				**QDEV-SEP**			
		Iters		CPU(s)		Iters		CPU(s)	
Instance	Scenarios	Avg	Stdev	Avg	Stdev	Avg	Stdev	Avg	Stdev
cep1	216	2.91	0.30	0.03	0.02	2.91	0.30	0.03	0.02
cep1a	216	6.45	1.51	0.07	0.02	6.45	1.51	0.07	0.04
cep1sk	216	11.73	2.05	0.11	0.03	9.82	1.78	0.10	0.06
pgp2	576	50.09	7.76	0.87	0.16	37.00	6.47	0.67	0.14
pgp2e	576	58.36	8.10	0.84	0.14	43.36	8.83	0.62	0.15
pgp2f	576	56.55	8.08	0.79	0.18	42.45	6.73	0.59	0.12
gbd	6.5×10^5	55.09	10.02	519.80	88.33	40.36	9.84	297.45	57.51
gbd_sk3	6.5×10^5	35.91	5.70	320.28	51.18	28.18	4.90	197.56	30.80
LandS	10^5	47.64	6.34	651.46	58.95	34.18	5.67	453.63	92.88
20Tr_l	512	3112.73	464.63	1895.36	283.70	2289.36	821.20	1338.53	539.05
20Tr_lsk1	512	2779.45	717.35	1700.16	445.33	1957.27	682.48	1180.37	461.43
20Tr_m	1024	3045.18	779.73	3146.00	748.37	2186.45	753.56	2188.09	806.54
20Tr_msk1	1024	2959.27	806.92	3347.66	874.85	2186.18	563.99	2473.89	621.80
20Tr_msk2	1024	2518.45	702.65	2966.15	785.77	2212.82	675.52	2403.39	808.36
20Tr_h	2048	2645.09	639.44	5066.86	1016.04	2070.55	733.13	3681.41	1298.40
ssnTr_l	735	1041.18	486.43	288.99	133.31	975.27	534.83	265.77	160.64
ssnTr_m	5145	1694.18	773.04	2853.00	1290.11	1532.55	577.68	2514.10	970.38
ssnTr_h	36015	1915.70	1453.51	17538.67	12339.23	1676.82	474.47	14395.34	3999.08
stormTr_l	625	38.00	8.11	8.23	2.08	26.45	8.37	3.93	1.14
stormTr_m	15625	39.55	9.08	213.94	53.71	24.64	9.40	89.02	32.93
stormTr_h	390625	38.82	9.09	4592.92	1128.29	24.36	9.37	2175.38	805.88

The impact of scenario size on ASD-AGG, ASD-SEP, QDEV-AGG, and QDEV-SEP algorithms' results can also be observed from Table 5.3 and 5.4 for ASD and QDEV respectively. Note the table's rows organize instances by increasing scenario size. Both tables across all instances demonstrated an increase in average computational time with increasing scenario size. The truncated instances of *ssn* (ssnTr) showed an increase in total average iterations with increasing problem size for all versions of the algorithms. The truncated instances of 20 (20Tr) showed a decrease in average iterations with increasing problem size for both versions of the algorithm for ASD. Conversely, there was an increase in average iterations with increasing problem size for 20Tr instances for both versions of the algorithm for QDEV. Average iterations for storm truncated (stormTr) instances changed very little with increasing problem size for both algorithms.

Effect of λ on optimal values of objective function for ASD and QDEV

Table 5.5 and 5.6 show the overall results of using the decomposition algorithm to solve the mean-ASD and mean-QDEV SLPs, respectively. The optimal objective value is shown for each value of the risk tradeoff parameter λ. Recall that λ represents the weight given to the dispersion statistic in the objective function of both ASD and QDEV models. For ASD $(1 - \lambda)$ represents the weight placed on the expected value term, $\mathbb{E}[f(x, \tilde{\omega})]$. Thus the optimal value for each instance corresponding to the *risk-neutral* case (expectation only) is represented under the $\lambda = 0$ column. The objective function values that tradeoff the weight given to expected cost and risk measure are shown in the $\lambda = 0.1$ to $\lambda = 0.9$ columns. The $\lambda = 1$ column, shows objective results when the model is completely *risk-averse* meaning no weight is given to the $\mathbb{E}[f(x, \tilde{\omega})]$ in the objective function and its value is based solely on the dispersion statistic. Observe for all instances other than ssn, the objective values increase with the increasing risk levels of λ.

Table 5.5: Objective value results using ASD with increasing λ

	λ										
Instance	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
cep1	355160	372480	389800	407120	424441	441761	459081	476401	493721	511041	528362
cep1a	394890	412087	429283	446480	463677	480874	498071	515267	532464	549661	566858
cep1sk	320344	335925	351506	367088	382669	398250	413831	429412	444993	460575	476156
pgp2	447.324	449.992	452.659	455.327	457.994	460.662	463.283	465.898	468.513	471.119	473.699
pgp2e	413.94	418.95	423.96	428.98	433.97	438.97	443.91	448.85	453.73	458.48	463.15
pgp2f	457.03	462.54	468.05	473.54	479.00	484.47	489.83	495.12	500.41	505.67	510.86
gbd	1655.630	1680.750	1705.870	1731.000	1756.120	1781.230	1806.340	1831.440	1856.540	1881.570	1906.550
gbd_sk3	2371.28	2412.07	2451.53	2491	2529.35	2558.24	2583.17	2606.05	2627.44	2648.58	2669.72
LandS	225.629	228.026	230.412	232.787	235.151	237.505	239.848	242.179	244.501	246.811	249.110
20Tr_l	242632	242793	242953	243114	243274	243434	243594	243749	243901	244050	244196
20Tr_sk1	242126	242236	242346	242456	242566	242676	242785	242895	243005	243115	243225
20Tr_m	243346	243505	243663	243822	243980	244138	244297	244455	244614	244772	244930
20Tr_msk1	239370	239562	239754	239942	240108	240266	240421	240575	240729	240883	241037
20Tr_msk2	243568	243746	243925	244103	244282	244461	244639	244818	244997	245172	245337
20Tr_h	243791	243981	244159	244330	244501	244672	244839	245005	245172	245339	245506
stormTr_l	13014000	13024700	13035300	13046000	13056700	13067300	13078000	13088600	13099300	13109900	13120600
stormTr_m	13696600	13708300	13719900	13731500	13743200	13754800	13766500	13778100	13789700	13801400	13813000
stormTr_h	13918300	13930400	13942600	13954700	13966800	13979000	13991100	14003300	14015400	14027500	14039700
ssnTr	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 5.6: Objective value results using QDEV with increasing λ

	λ										
Instance	**0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
cep1	355160	387111	419063	451014	482965	514917	546868	578819	610771	642722	674673
cep1a	394890	428262	461633	495005	528377	561749	595121	628492	661864	695236	728608
cep1sk	320344	348956	377568	406179	434791	463403	492015	520627	549238	577850	606462
pgp2	447.324	452.638	457.952	463.234	468.446	473.624	478.770	483.915	489.037	494.149	499.259
pgp2e	413.935	423.571	433.201	442.733	452.170	461.410	470.428	479.330	488.108	496.702	505.030
pgp2f	457.026	467.798	478.491	489.151	499.526	509.771	519.786	529.736	539.686	549.547	559.067
gbd	1655.63	1703.30	1750.96	1798.58	1846.00	1893.08	1939.95	1986.74	2033.34	2078.99	2123.29
gbd_sk3	2371.28	2451.03	2528.85	2583.11	2627.43	2669.71	2711.93	2754.08	2796.23	2838.38	2880.27
LandS	225.629	230.408	235.144	239.836	244.483	249.086	253.644	258.158	262.629	267.059	271.45
20Tr_l	242632	242952	243270	243588	243897	244193	244480	244768	245056	245344	245631
20Tr_lsk1	242126	242345	242564	242784	243003	243222	243441	243660	243879	244098	244317
20Tr_m	243346	243663	243979	244296	244613	244929	245246	245563	245879	246196	246512
20Tr_msk1	239370	239744	240103	240417	240723	241028	241331	241635	241938	242232	242516
20Tr_msk2	243568	243918	244269	244620	244970	245320	245646	245964	246263	246547	246814
20Tr_h	243791	244158	244499	244836	245169	245502	245835	246166	246478	246788	247098
stormTr_l	13014000	13035300	13056700	13078000	13099300	13120600	13141900	13163200	13184500	13205800	13227100
stormTr_m	13696600	13719900	13743200	13766500	13789700	13813000	13836300	13859600	13882800	13906100	13929400
stormTr_h	13918300	13942600	13966800	13991100	14015400	14039700	14064000	14088300	14112500	14136800	14161100
ssnTr_l/m/h	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

We also analyzed how the optimal values for each instance varied from $\lambda = 0$ to $\lambda = 1$. Table 5.7 summarizes the span of the objective values between the completely risk neutral ($\lambda = 0$) and *risk-averse* ($\lambda = 1$) cases for ASD and QDEV respectively. The % span (or percentage difference) measures the difference (or span) between the objective values at $\lambda = 0$ and $\lambda = 1$ over *risk-neutral* objective value. Observe that the percentage span for QDEV is larger (almost twice) than that for ASD. This demonstrates the broader sensitivity of the QDEV model to changes in the risk tradeoff parameter λ.

Table 5.7: Span of optimal objective function values for range of λ values

		ASD		**QDEV**	
Instance	$\lambda = 0$	$\lambda = 1$	% Span	$\lambda = 1$	% Span
cep1	355160	528362	48.87	674673	89.96
cep1a	394890	566858	43.55	728608	84.51
cep1sk	320344	476156	48.64	606462	89.32
pgp2	447.324	473.699	5.90	499.259	11.61
pgp2e	413.94	463.15	11.89	505.030	22.01
pgp2f	457.03	510.86	11.78	559.067	22.33
gbd	1655.630	1906.550	15.16	2123.29	28.25
gbd_sk3	2371.28	2669.72	12.59	2880.27	21.46
LandS	225.629	249.110	10.41	271.45	20.31
20Tr_l	242632	244196	0.64	245631	1.24
20Tr_sk1	242126	243225	0.45	244317	0.90
20Tr_m	243346	244930	0.65	246512	1.30
20Tr_msk1	239370	241037	0.70	242516	1.31
20T_msk2	243568	245337	0.73	246814	1.33
20Tr_h	243791	245506	0.70	247098	1.36
stormTr_l	13014000	13120600	0.82	13227100	1.64
stormTr_m	13696600	13813000	0.85	13929400	1.70
stormTr_h	13918300	14039700	0.87	14161100	1.74
ssnTr_l/m/h	0.00	0.00	0.00	0.00	0.00

Comparison of risk-averse and risk-neutral approaches

As previously discussed at the beginning of this section, large changes in objective value, $\mathbb{E}[f(x,\tilde{\omega})]$, and first-stage decision vector x (over each risk level) indicate a stronger case for the mean-risk approach. Note changes in the first-stage solutions without changes in both objective value and $\mathbb{E}[f(x,\tilde{\omega})]$, indicates an alternative optimal solution and still will not indicate a case for the mean-risk approach. But large changes in both objective value and $\mathbb{E}[f(x,\tilde{\omega})]$ (over all or most risk levels) implies changes in the associated first-stage decision vector x. These are the conditions for the mean-risk (risk-averse) approach to be preferred over the risk-neutral one. Thus to further evaluate the appropriateness of the mean-risk approach over the risk neutral one, we discuss changes in $\mathbb{E}[f(x,\tilde{\omega})]$ over the different risk levels for each of the instances. First, we will discuss the related results for original and truncated instances with ASD and QDEV. Then results will be compared for select instances that have nonuniform distribution (see discussion at the end of Section 5.2). The results below are based on ASD-AGG and QDEV-AGG algorithms, which have the same or very similar results for the separate cut version of the algorithms. The appendix contains tables summarizing results for all instances and versions of the algorithms.

For original and truncated instances the impact of the $\mathbb{E}[f(x,\tilde{\omega})]$ over different λ varies. For *cep1*, *stormTr_l/m/h*, there is no change in $\mathbb{E}[f(x,\tilde{\omega})]$ over the different risk levels. Other instances had changes as follows: for $\lambda = 0$ to 0.5 the value of $\mathbb{E}[f(x,\tilde{\omega})]$ is constant at 447.32, while from $\lambda = 0.6$ to 1 it varies from 447.60 to 447.90 for *pgp2*; $\mathbb{E}[f(x,\tilde{\omega})] = 225.629$ at $\lambda = 0$, while from $\lambda = 0.1$ to 1 it varies from 225.637 to 226.153 for *LandS*; and $\mathbb{E}[f(x,\tilde{\omega})] = 242632$ at $\lambda = 0$, while from $\lambda = 0.1$ to 1 it varies from 242633 to 242754 for *20Tr_l*. Instances *gbd* performs similarly to *pgp2*, while *20Tr_h* performs similarly to *LandS*. Instance *20Tr_m* with $\mathbb{E}[f(x,\tilde{\omega})]$ of 243346, changes one point to 243347 for $\lambda = 0.6$ only. Instance *ssnTr_l/m/h* has no change in $\mathbb{E}[f(x,\tilde{\omega})]$ or objective value and alternative first-stage solutions for each λ value.

There are some differences in results for $\mathbb{E}[f(x,\tilde{\omega})]$ when using QDEV. The results show that for $\lambda = 0$ to 0.2 the value of $\mathbb{E}[f(x,\tilde{\omega})]$ remains constant at 447.324, but for $\lambda = 0.3$ to 1 it varies from 447.597 to 469.253 for *pgp2*; for $\lambda = 0$ to 0.1 the value of $\mathbb{E}[f(x,\tilde{\omega})]$ remains constant at 1655.63, but for $\lambda = 0.2$ to 1 it varies from 1655.71 to 1794.24 for *gbd*; and $\mathbb{E}[f(x,\tilde{\omega})] = 242632$ at $\lambda = 0$, but for $\lambda = 0.1$ to 1 it varies from 242634 to 244068 for *20Tr_l*. Problems *LandS* and *20Tr_h* perform similarly to *20Tr_l*. Instance *20Tr_m* has mean = 243346 that changes for some λ values by one point. $\mathbb{E}[f(x,\tilde{\omega})]$ of *stormTr_l* equals 13014000 and changes for $\lambda =$ 0.5-0.6 to 13014100 and for $\lambda = 1$ to 13120300. For *stormTr_m*, *stormTr_h* $\mathbb{E}[f(x,\tilde{\omega})]$ only changes for $\lambda = 1$. All other results for $\mathbb{E}[f(x,\tilde{\omega})]$, for original and truncated instances, are similar to descriptions for ASD model in the previous paragraph.

The following nonuniform instances were created, because the original and truncated instances had distributions that were uniform: *cep1a* and *cep1sk* for instance *cep1*, *pgp2e* and *pgp2f* for *pgp2*, *gbd_sk3* for *gbd*, *20Tr_lsk1* for *20Tr_l*, and *20Tr_msk1* and *20Tr_msk2* for *20Tr_m*. This allows us to provide further details on the differences between the ASD and QDEV models by showing results for ASD-AGG and QDEV-AGG algorithms to solve instances with uniform and nonuniform distributions. We will specifically compare *cep1* and *cep1a* as well as *pgp2* and *pgp2e* to provide representative illustrations of all the results.

Table 5.8 and 5.9 show results of the ASD model with instances *cep1* (with uniform distribution) and *cep1a* (nonuniform distribution), respectively. The columns of the tables show for each tradeoff parameter λ the overall objective value (Obj.), the mean value $\mathbb{E}[f(x,\tilde{\omega})]$, the ASD value $\phi_{ASD}(x)$, and the corresponding mean weighted value $(1-\lambda)\mathbb{E}[f(x,\tilde{\omega})]$ and ASD weighted value $\lambda\phi_{ASD}(x)$. The ASD model's first-stage optimal solution vector $x = [x_1, x_2, ..., x_8]$, for each value of λ, is shown in Table 5.10 for *cep1* and *cep1a*. Now, Table 5.11 and 5.12 show results of the QDEV model with instances for *cep1* (with uniform distribution) and *cep1a* (nonuniform distribution), respectively. The columns of the tables show for each tradeoff parameter λ Obj., $\mathbb{E}[f(x,\tilde{\omega})]$, expected quantile deviation $\mathbb{E}[\nu(\omega)]$, the target z, weighted mean $(1-\lambda\varepsilon_1)\mathbb{E}[f(x,\tilde{\omega})]$, weighted expected quantile deviation $\lambda(\varepsilon_1+\varepsilon_2)\nu(\omega)$, and weighted target $\lambda\varepsilon_1 z$. The QDEV model's first-stage optimal solution vector $x = [x_1, x_2, ..., x_8]$ and its target value z, for each value of λ, is shown in Table 5.13 for *cep1* and *cep1a*.

Observe from Table 5.8 and 5.9 the value of $\mathbb{E}[f(x,\tilde{\omega})]$ remains constant across all λ values. In addition, Table 5.13 for *cep1* and *cep1a* shows the first-stage optimal solutions are the same for both instances. This means both the uniform and nonuniform $\mathbb{E}[f(x,\tilde{\omega})]$ results for *cep1* are not sensitive to the risk level. These results are further illustrated through Figure 5.1. 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 $\lambda\phi_{ASD}(x)$. Observe how the 'Mean' value is constant on both graphs. Now moving to the results for QDEV, observe Table 5.11 and Table 5.12. For both uniform distributed and nonuniform distributed plots, the value of $\mathbb{E}[f(x,\tilde{\omega})]$ remains constant across all λ values except for the completely risk-averse risk level at $\lambda = 1.0$. Table 5.13 for *cep1* and *cep1a* demonstrates that the optimal first-stage solutions remain the same for all values except for the target value z which is constant for each instance. Figure 5.2 shows that the 'Mean' value is constant on both graphs until $\lambda = 1.0$.

Table 5.8: Results of ASD-AGG for cep1

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\phi_{ASD}(x)$	$(1-\lambda)*$ $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda\phi_{ASD}(x)$
0.0	355160	355160	528362	355160	0
0.1	372480	355160	528362	319644	52836
0.2	389800	355160	528362	284128	105672
0.3	407120	355160	528362	248612	158508
0.4	424441	355160	528362	213096	211345
0.5	441761	355160	528362	177580	264181
0.6	459081	355160	528362	142064	317017
0.7	476401	355160	528362	106548	369853
0.8	493721	355160	528362	71032	422689
0.9	511041	355160	528362	35516	475525
1.0	528362	355160	528362	0	528362

Table 5.9: Results of ASD-AGG for cep1a

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\phi_{ASD}(x)$	$(1-\lambda)*$ $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda\phi_{ASD}(x)$
0.0	394890	394890	566858	394890	0
0.1	412087	394890	566858	355401	56686
0.2	429283	394890	566858	315912	113372
0.3	446480	394890	566858	276423	170057
0.4	463677	394890	566858	236934	226743
0.5	480874	394890	566858	197445	283429
0.6	498071	394890	566858	157956	340115
0.7	515267	394890	566858	118467	396801
0.8	532464	394890	566858	78978	453486
0.9	549661	394890	566858	39489	510172
1.0	566858	394890	566858	0	566858

Table 5.10: First-stage solution of ASD-AGG for cep1 and cep1a

λ	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0.0	0	0	1833.33	2500	0	0	2333.33	3000
0.1	0	0	1833.33	2500	0	0	2333.33	3000
0.2	0	0	1833.33	2500	0	0	2333.33	3000
0.3	0	0	1833.33	2500	0	0	2333.33	3000
0.4	0	0	1833.33	2500	0	0	2333.33	3000
0.5	0	0	1833.33	2500	0	0	2333.33	3000
0.6	0	0	1833.33	2500	0	0	2333.33	3000
0.7	0	0	1833.33	2500	0	0	2333.33	3000
0.8	0	0	1833.33	2500	0	0	2333.33	3000
0.9	0	0	1833.33	2500	0	0	2333.33	3000
1.0	0	0	1833.33	2500	0	0	2333.33	3000

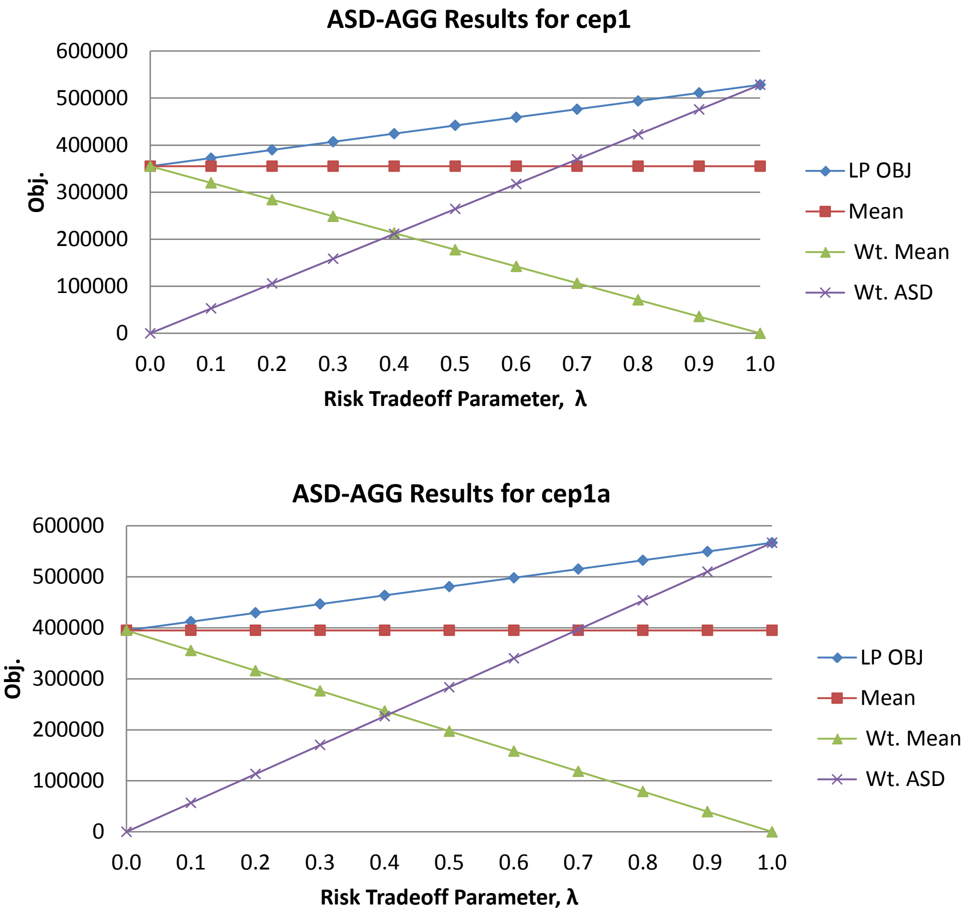


Figure 5.1: ASD optimal value versus risk tradeoff parameter λ

Table 5.11: Results of QDEV-AGG for cep1

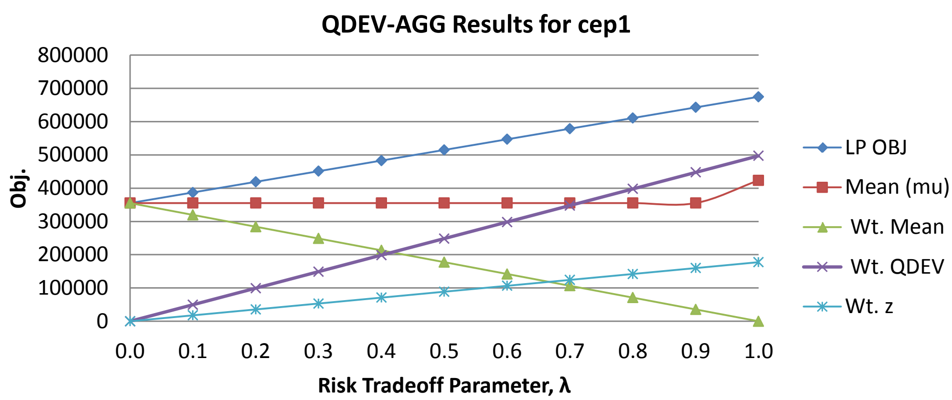
λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$
0.0	355160	355160	355160	0	355160	0	0
0.1	387111	355160	248547	177580	319644	49709	17758
0.2	419063	355160	248547	177580	284128	99419	35516
0.3	451014	355160	248547	177580	248612	149128	53274
0.4	482965	355160	248547	177580	213096	198837	71032
0.5	514917	355160	248547	177580	177580	248547	88790
0.6	546868	355160	248547	177580	142064	298256	106548
0.7	578819	355160	248547	177580	106548	347965	124306
0.8	610771	355160	248547	177580	71032	397675	142064
0.9	642722	355160	248547	177580	35516	447384	159822
1.0	674673	423446	248547	177580	0	497093	177580

Table 5.12: Results of QDEV-AGG for cep1a

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$
0.0	394890	394890	394890	0	394890	0	0
0.1	428262	394890	261046	206517	355401	52209	20652
0.2	461633	394890	260521	207567	315912	104208	41513
0.3	495005	394890	260521	207567	276423	156312	62270
0.4	528377	394890	260521	207567	236934	208416	83027
0.5	561749	394890	260521	207567	197445	260521	103783
0.6	595121	394890	260521	207567	157956	312625	124540
0.7	628492	394890	260521	207567	118467	364729	145297
0.8	661864	394890	260521	207567	78978	416833	166053
0.9	695236	394890	260521	207567	39489	468937	186810
1.0	728608	466076	260521	207567	0	521041	207567

Table 5.13: First-stage solution of QDEV-AGG for cep1 and cep1a

λ	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	z (cep1)	z (cep1a)
0	0	0	1833.33	2500	0	0	2333.33	3000	0	0
0.1	0	0	1833.33	2500	0	0	2333.33	3000	177580	206517
0.2	0	0	1833.33	2500	0	0	2333.33	3000	177580	207567
0.3	0	0	1833.33	2500	0	0	2333.33	3000	177580	207567
0.4	0	0	1833.33	2500	0	0	2333.33	3000	177580	207567
0.5	0	0	1833.33	2500	0	0	2333.33	3000	177580	207567
0.6	0	0	1833.33	2500	0	0	2333.33	3000	177580	207567
0.7	0	0	1833.33	2500	0	0	2333.33	3000	177580	207567
0.8	0	0	1833.33	2500	0	0	2333.33	3000	177580	207567
0.9	0	0	1833.33	2500	0	0	2333.33	3000	177580	207567
1	0	0	1833.33	2500	0	0	2333.33	3000	177580	207567



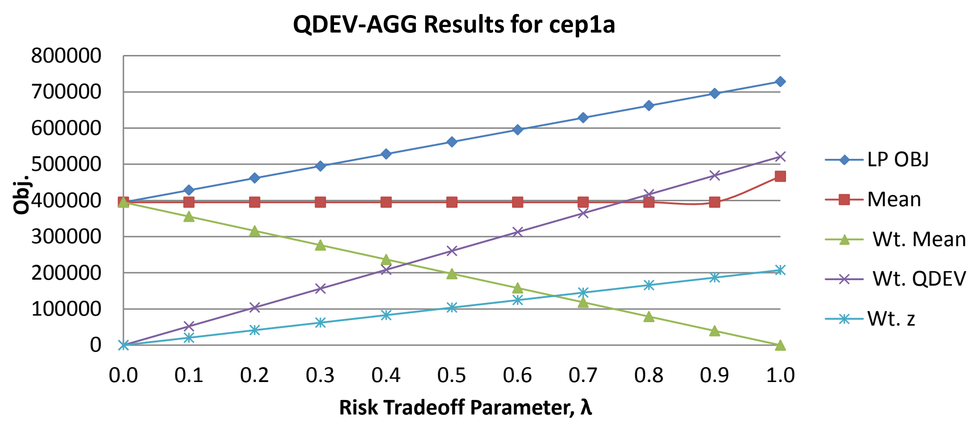


Figure 5.2: QDEV optimal value versus risk tradeoff parameter λ

Now turn to the results for *pgp2*. Table 5.14 and 5.15 show results of the ASD model (including first-stage solutions) for instances *pgp2* (with uniform distribution) and *pgp2e* (nonuniform distribution), respectively. Table 5.16 and Table 5.17 show results of the QDEV model for *pgp2*(uniform distribution) with the latter table displaying its first-stage solution. Table 5.18 and Table 5.19 show results for instance *pgp2e* (nonuniform distribution) with the latter table displaying its first-stage optimal solutions. The column headings were already described with the tables for the *cep1* related instances.

We begin with the ASD model results in Table 5.14. As can be seen in Table 5.14, 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 $\lambda = 0.9$ which coincides with a corresponding change in x_1 at both λ and x_2 at $\lambda = 0.9$. This means that the optimal solution is not sensitive to the risk level. On the other hand, in Table 5.15 we see that the mean value $\mathbb{E}[f(x, \tilde{\omega})]$ varies from $\lambda = 0.4$ to $\lambda = 1.0$ coinciding with multiple changes in first-stage decisions. The tradeoff between the mean value and the deviation value is better seen using the plot in Figure 5.3. 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 $\lambda\phi_{ASD}(x)$. Observe how the 'Mean' value is fairly constant for *pgp2* as compared to *pgp2e*. In fact we saw that the expected value $\mathbb{E}[f(x, \tilde{\omega})]$ for different λ values changed with corresponding changes in the first-stage optimal solution x.

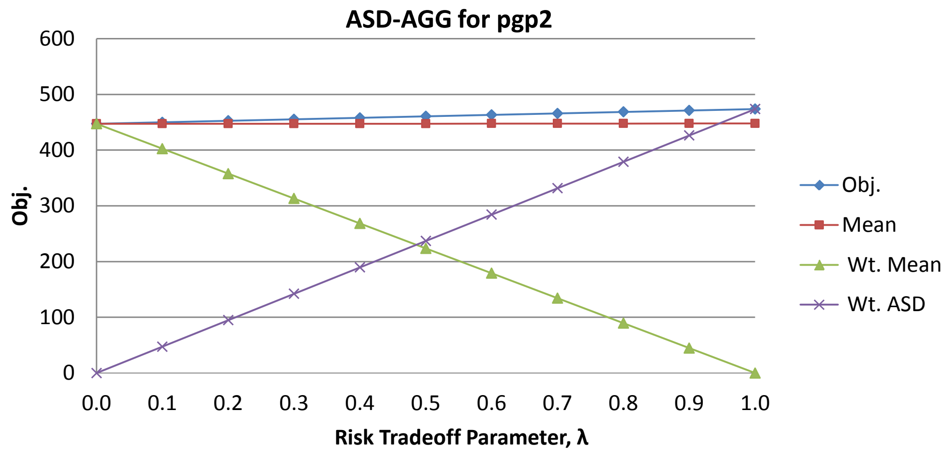
In Table 5.16 we see that there is more variation in the mean value $\mathbb{E}[f(x, \tilde{\omega})]$ and objective value as compared to the ASD results in Table 5.14. In fact now we see a big change from $\lambda = 0.0$ to $\lambda = 1.0$ in the objective value. Nevertheless, the $\mathbb{E}[f(x, \tilde{\omega})]$ from $\lambda = 0.0$ to $\lambda = 0.9$ does not change significantly due to the uniform distribution for this instance. Thus the changes seen in the first-stage optimal solutions in Table 5.17 for $\lambda = 0.0$ to $\lambda = 0.9$ are also not significant. In Table 5.18 we see that for the nonuniform distribution case the objective value and $\mathbb{E}[f(x,\tilde{\omega})]$ vary from $\lambda = 0.0$ to $\lambda = 1.0$. This means that the optimal solutions shown in Table 5.19 are very responsive to the level of risk. Figure 5.4 plots the tradeoff between the mean value and the QDEV value. In the plot 'Wt. QDEV' is the value of $\lambda\phi_{QDEV_\alpha}(x)$ and 'Wt. z' is the value of $\lambda\varepsilon_1 z$. Now we can clearly see the relative variation of the 'Mean' value for *pgp2* compared to *pgp2e*.

Table 5.14: Results of ASD-AGG for pgp2

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\phi_{ASD}(x)$	$(1-\lambda)*$ $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda\phi_{ASD}(x)$	x_1	x_2	x_3	x_4
						(first-stage sol.)			
0.0	447.32	447.32	474.00	447.32	0.00	1.5	5.5	5.0	5.5
0.1	449.99	447.32	474.00	402.59	47.40	1.5	5.5	5.0	5.5
0.2	452.66	447.32	474.00	357.86	94.80	1.5	5.5	5.0	5.5
0.3	455.33	447.32	474.00	313.13	142.20	1.5	5.5	5.0	5.5
0.4	457.99	447.32	474.00	268.40	189.60	1.5	5.5	5.0	5.5
0.5	460.66	447.33	474.00	223.66	237.00	1.5	5.5	5.0	5.5
0.6	463.28	447.60	473.74	179.04	284.25	0.5	5.5	6.0	5.5
0.7	465.90	447.60	473.74	134.28	331.62	0.5	5.5	6.0	5.5
0.8	468.51	447.60	473.74	89.52	378.99	0.5	5.5	6.0	5.5
0.9	471.12	447.90	473.70	44.79	426.33	0.0	5.5	6.5	5.5
1	473.70	447.90	473.70	0.00	473.70	0.0	5.5	6.5	5.5

Table 5.15: Results of ASD-AGG for pgp2e

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\phi_{ASD}(x)$	$(1-\lambda)*$ $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda\phi_{ASD}(x)$	x_1	x_2	x_3	x_4
						(first-stage sol.)			
0.0	413.94	413.94	464.08	413.94	0.00	1.5	6.5	3.5	9.0
0.1	418.95	413.94	464.08	372.54	46.41	1.5	6.5	3.5	9.0
0.2	423.96	413.94	464.08	331.15	92.82	1.5	6.5	3.5	9.0
0.3	428.98	413.94	464.08	289.76	139.22	1.5	6.5	3.5	9.0
0.4	433.97	414.00	463.93	248.40	185.57	1.5	7.0	3.5	8.5
0.5	438.97	414.00	463.93	207.00	231.96	1.5	7.0	3.5	8.5
0.6	443.91	414.28	463.66	165.71	278.20	2.5	6.0	3.5	8.5
0.7	448.85	414.28	463.66	124.29	324.56	2.5	6.0	3.5	8.5
0.8	453.73	415.06	463.4	83.01	370.72	2.0	5.5	4.5	8.5
0.9	458.48	416.42	463.15	41.64	416.83	1.5	5.5	5.0	9.0
1.0	463.15	416.42	463.15	0.00	463.15	1.5	5.5	5.0	9.0



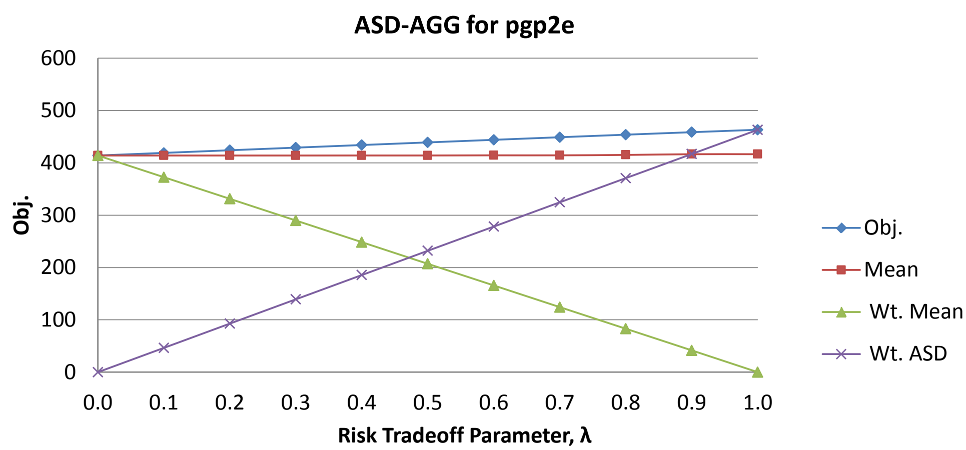


Figure 5.3: ASD optimal value versus risk tradeoff parameter λ

Table 5.16: Results of QDEV-AGG for pgp2

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$
0.0	447.32	447.32	447.32	0.00	447.32	0.00	0.00
0.1	452.64	447.32	28.48	443.50	402.60	5.70	44.40
0.2	457.95	447.32	28.48	443.50	357.86	11.39	88.70
0.3	463.23	447.60	27.51	444.70	313.32	16.51	133.41
0.4	468.45	447.60	27.52	444.69	268.56	22.01	177.88
0.5	473.62	447.90	27.03	445.30	224.00	27.03	222.65
0.6	478.77	447.90	27.03	445.30	179.16	32.43	267.18
0.7	483.92	447.90	27.03	445.30	134.37	37.84	311.71
0.8	489.04	448.14	26.98	445.30	89.63	43.17	356.24
0.9	494.15	448.14	26.98	445.30	44.81	48.57	400.77
1.0	499.26	469.25	27.00	445.30	0.00	53.96	445.30

Table 5.17: First-stage solution of QDEV-AGG for pgp2

λ	x_1	x_2	x_3	x_4	z
0.0	1.5	5.5	5.0	5.5	0.0
0.1	1.5	5.5	5.0	5.5	443.5
0.2	1.5	5.5	5.0	5.5	443.5
0.3	0.5	5.5	6.0	5.5	444.7
0.4	0.5	5.5	6.0	5.5	444.7
0.5	0.0	5.5	6.5	5.5	445.3
0.6	0.0	5.5	6.5	5.5	445.3
0.7	0.0	5.5	6.5	5.5	445.3
0.8	1.0	4.5	6.5	5.5	445.3
0.9	1.0	4.5	6.5	5.5	445.3
1.0	1.7	3.8	6.5	5.5	445.3

Table 5.18: Results of QDEV-AGG for pgp2e

λ	Obj.	$\mathbb{E}[f(x,\tilde{\omega})]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(x,\tilde{\omega})]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$
0.0	413.94	413.94	413.94	0.00	413.94	0.00	0.00
0.1	423.57	413.94	61.97	386.35	372.54	12.39	38.64
0.2	433.20	414.00	61.75	386.50	331.20	24.70	77.30
0.3	442.73	414.30	61.31	386.50	290.00	36.78	115.95
0.4	452.17	415.07	60.07	387.70	249.04	48.05	155.08
0.5	461.41	415.41	59.56	388.30	207.70	59.56	194.15
0.6	470.43	416.42	57.57	391.30	166.57	69.08	234.78
0.7	479.33	417.84	56.69	392.30	125.35	79.37	274.61
0.8	488.11	419.17	55.02	395.30	83.83	88.03	316.24
0.9	496.70	420.82	54.32	396.50	42.08	97.77	356.85
1.0	505.03	450.98	53.97	397.10	0.00	107.93	397.10

Table 5.19: First-stage solution of QDEV-AGG for pgp2e

λ	x_1	x_2	x_3	x_4	z
0.0	1.5	6.5	3.5	9.0	0.0
0.1	1.5	6.5	3.5	9.0	386.3
0.2	1.5	7.0	3.5	8.5	386.5
0.3	2.5	6.0	3.5	8.5	386.5
0.4	2.0	5.5	4.5	8.5	387.7
0.5	1.5	5.5	5.0	8.5	388.3
0.6	1.5	5.5	5.0	9.0	391.3
0.7	2.5	5.5	5.0	8.0	392.3
0.8	2.5	5.5	5.0	8.5	395.3
0.9	1.5	5.5	6.0	8.5	396.5
1.0	1.0	5.5	6.5	8.5	397.1

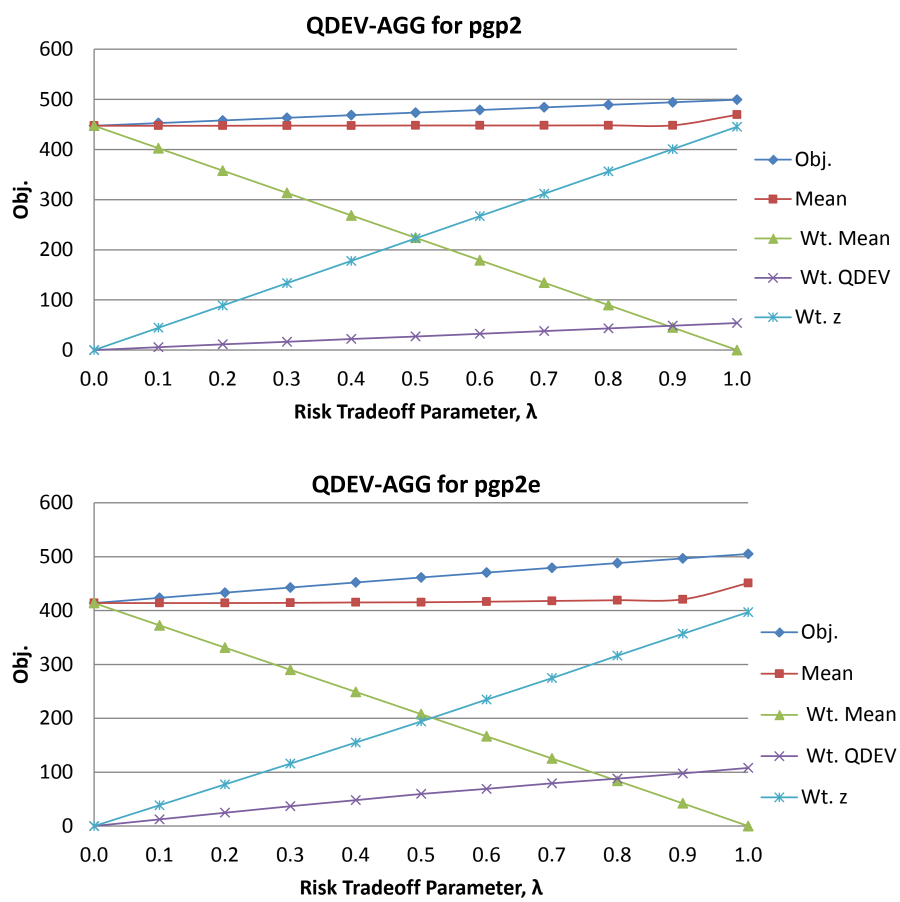


Figure 5.4: QDEV optimal value versus risk tradeoff parameter λ

When comparing *pgp2*, *gbd*, and *20Tr_m* to its nonuniform distributed instances, the objective values, mean ($\mathbb{E}[f(x, \tilde{\omega})]$), and first-stage solutions of the nonuniform distributed instances all change significantly from $\lambda = 0.0$ to $\lambda = 1.0$. This is true for both ASD and QDEV algorithms. Instance *20Tr_l* had more significant changes in objective value, mean ($\mathbb{E}[f(x, \tilde{\omega})]$) and first-stage solutions, over different values of λ, then the nonuniform version *20Tr_l_sk1* for both ASD and QDEV algorithms. For each of these instances there was a broader change in the $\mathbb{E}[f(x, \tilde{\omega})]$ for the QDEV model than for the ASD model.

5.4 Discussion

We now discuss the insights from the reported computational results. In terms of algorithm performance, ASD-SEP Algorithm 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. 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. With regard to size analysis, the number of algorithm iterations does not necessarily increase with the number of scenarios, but CPU time increases with increasing problem size.

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 that with ASD instances no algorithm is superior. However, for QDEV instance clearly QDEV-SEP is the preferred algorithm. It performs equivalently or better in terms of average algorithm iterations and CPU time. When comparing ASD and QDEV models it was observed that the ASD model had a smaller range of objective values compared to the QDEV model. In addition, the percentage span for QDEV was shown to be nearly twice the one for ASD. This indicates the flexibility of the QDEV model to changes in the risk tradeoff parameter λ. This suggests that the ASD model is relatively more conservative compared to the QDEV model. This was also observed in Ahmed (2006) for a 100 scenario instance of gbd.

To provide further information on the differences between the ASD and QDEV models, we showed detailed results using the ASD-AGG and QDEV-AGG algorithms to solve instances with *uniform* and *nonuniform* distributions, respectively. This enabled us to investigate the appropriate usage of the risk-averse and risk-neutral approaches. Recall that for all instances other than the *ssn* truncated instances, optimal objective values increase as the risk-level λ increases. Clearly *ssn* instances are not sensitive to risk and therefore, the risk-neutral approach is appropriate. Results for all other (except for *ssn* truncated instances) original and truncated instances, show no or insignificant change in $\mathbb{E}[f(x, \tilde{\omega})]$ (or first-stage solutions) with increasing risk-level λ, indicating the solution and decisions for the model are remaining the same despite the change in risk-level. This insensitivity to the risk-level is attributed to the uniform marginal distributions of most original and truncated problems, and it indicates the risk-neutral approach is appropriate. But once nonuniform distributions were created for *cepl*, *pgp2*, *gbd*, and *20Tr_m*, their

$\mathbb{E}[f(x, \tilde{\omega}$ (and its corresponding first-stage solutions) began to change significantly with increasing λ values. This indicated the risk-averse approach was appropriate.

Furthermore, if we analyze the results for *cep1*, *cep1a*, *pgp2* and *pgp2e* in context of information specific to each application, we gain another level of insight into the meaning of these results. Recall *cep1* and *cep1a* are the uniform and nonuniform version of a machine capacity expansion problem. Its variables x_1, x_2, x_3, and x_4 represent the weekly hours of additional capacity assigned to machine 1, 2, 3, and 4 respectively. First-stage variables x_5, x_6, x_7, and x_8 represent the total weekly hours of capacity (including additional capacity) corresponding to machine 1, 2, 3, and 4 respectively. In the ASD and QDEV results for *cep1* and *cep1a* (see Table 5.10 and 5.13), we saw that variables x_3, x_4, x_7, and x_8 were the only first-stage variables taking on nonzero values that remained constant for all λ values. This indicates that despite the difference in the uniform and nonuniform distributed stochastic demand, machines 3 and 4 were always the appropriate machines to handle additional machining hours to process parts. Thus the risk-averse approach is unnecessary for this application. Perhaps if the values and not just the probabilities of the demand distribution (represented in STOCH files) were significantly changed for this instance we may see a need for adding capacity to machines 1 or 2 and may see more of a need for the risk-averse approach.

As previously stated, *pgp2* (uniform) and *pgp2e* (nonuniform) are power generation planning problems that seek to select the minimum cost strategy for investing in electricity. First-stage variables x_1, x_2, x_3, and x_4, shown in Table 5.14 for *pgp2* and Table 5.15 for *pgp2e*, correspond to the power in kilowatts (kw) when investing in electricity produced from gas-fired, coal-fired, and nuclear powered generators. Using the ASD-AGG algorithm for *pgp2* suggests a constant power output of 5.5 kw at all risk levels for investment in coal-fired generation, a decreasing power output of 1.5

kw at $\lambda = 0.0$ to 0 kw at $\lambda = 1.0$ for gas-fired generation, an increasing power output of 5.0 kw at $\lambda = 0.0$ to 6.5 kw at $\lambda = 1.0$ for nuclear generation, and a constant power output of 5.5 kw at all risk levels for investment in type-4 generation. Using ASD-AGG Algorithm for *pgp2e* power outputs of the following are observable: 1.5 kw at λ from 0.0 to 0.5, 2.50 kw for λ at 0.6 and 0.7, 2.0 kw at $\lambda = 0.8$, 1.50 kw at λ from 0.9 to 1.0 for gas-fired generation; 6.5 kw at λ from 0.0 to 0.3 that decreases to 5.5 kw at λ from 0.8 to 1.0 for coal-fired generation; remains constant at 3.5 kw for λ at 0.0 to 0.7, 4.5 kw at $\lambda = 0.8$, and 5.0 kw at λ at 0.9 and 1.0 for nuclear generation; and 9.0 kw at λ from 0.0 to 0.3, 8.5 kw at λ from 0.4 to 0.8, and 9.0 kw at λ for 0.9 and 1.0 for type-4 generation. When comparing decisions for pgp2e with pgp2 there is consistent use of gas-fired generation that shows a small increase of use, increasing use of coal-fired generation at risk-levels from $\lambda = 0.0$ to $\lambda = 0.7$ and same value at $\lambda = 0.8$ to 1.0, decreasing use of nuclear generation at all risk levels, and there was an increase in type-4 generation at all risk levels. There is more variability in the power output for each of the generation types over the different risk levels λ for *pgp2* and *pgp2e* (see Table 5.17 and 5.19) when using the QDEV-AGG Algorithm instead of ASD-AGG Algorithm. Additionally, *pgp2e* has more diverse decisions for power output with increasing risk level than instance *pgp2*. The difference in the first-stage decisions across different risk levels (along with corresponding changes in optimal mean cost and objective function values) underscores the need for the mean-risk approach for this application, particularly for the instances with nonuniform distributed demand for electricity.

6. CONCLUSIONS AND FUTURE RESEARCH

6.1 Conclusions

Mean-risk stochastic programs include a risk measure in the objective to model risk averseness for optimization problems under uncertainty. We began this dissertation in its introduction, by motivating the need for the additional algorithms and a computational study for this class of stochastic programs. Then a literature review of stochastic programming and mean-risk stochastic programming was presented. Next, we defined the problem setting and presented some common mean-risk measures, their objectives, and deterministic equivalent problems. Then decomposition algorithms for absolute semideviation and quantile deviation were presented that use aggregated and separate optimality cuts. We derived the Aggregated Cut Subgradient-Based Algorithm (ASD-AGG) and implemented the L-shaped Algorithm for Quantile Deviation (QDEV-AGG). Based on (Ahmed (2006)), we implemented separate optimality cut versions of the algorithms for each risk measure called the Separate Cut Subgradient-Based Algorithm (ASD-SEP) and Separate Cut L-shaped Algorithm for Quantile Deviation (QDEV-SEP).

In this dissertation we report on a computational study of mean-risk two-stage stochastic linear programs with recourse based on absolute semideviation and quantile deviation. In this study we perform an empirical investigation of decomposition algorithms for this class of stochastic programs. In particular, we report on the performance of the ASD-AGG, ASD-SEP, QDEV-AGG, and QDEV-SEP algorithms. In addition, we analyze how the instance solutions vary across different levels of risk and obtain insights of when it is appropriate to use a given mean-risk measure.

The aggregated cut algorithms, ASD-AGG and QDEV-AGG have more iterations than the separate cut algorithms, ASD-SEP and QDEV-SEP. In terms of CPU time, the ASD-AGG and ASD-SEP algorithms have similar performance. However, the QDEV-SEP algorithm outperforms the QDEV-AGG algorithm. CPU time is shown to increase with increasing scenario size whereas no such conclusion could be made with regard to the number of iterations. The empirical study provides several insights. The computational results reveal 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 marginal distributions. However, when the distributions are changed to be nonuniform, the risk-neutral approach is found to be no longer 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.

6.2 Future Research

Deriving algorithms for mean-risk stochastic programs is still an open question in the literature. These algorithms have broad application for science and engineering problems as well as other areas that involve decision making under uncertainty. This work presented decomposition algorithms that divide the problems stage-wise using subgradient optimization. Other algorithms could be derived by decomposing the mean-risk problem scenario-wise or using another method of decomposition. Future work along this line of work is to perform a similar computational study for mean-risk stochastic integer programs. This study could include computations with additional risk measures. It would also be important to establish when it is appropriate to use the risk-averse approach over the risk-neutral one in this setting. However, this class of stochastic programs is much more challenging to solve and there is still a need to develop scalable algorithms that can be used in a computational study.

Further research could also involve extension of the above algorithms to solve the larger scenario instances of *20term*, *ssn*, and *storm*. In this study, we solved truncated versions of these larger instances instead of the full size problems. The Sample Average Approximation Algorithm (Kleywegt et al. (2001), Ruszczynski and Shapiro (2003)) and Progressing Hedging Algorithm (Rockafellar and Wets, 1991) are two algorithms that use techniques that are successful in solving large-scale instances. Principles of both these algorithms can be incorporated into extensions in the mean-risk setting to allow larger size problems to be solved.

REFERENCES

Ahmed, S. 2004. Mean-risk objectives in stochastic programming. Tech. rep., Georgia Institute of Technology, Industrial and Systems Engineering.

Ahmed, S. 2006. Convexity and decompostion of mean-risk stochastic programs. *Mathematical Programming* **106**(3) 433–446.

Artzner, P., F. Delbean, J.M Eber, D. Heath. 1999. Coherent measures of risk. *Mathematical Finance* **9** 203–228.

Bazaraa, M., J. Jarvis, H. Sherali. 2005. *Linear Programming And Network Flows*. 3rd ed. John Wiley and Sons, Inc.

Benders, J. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* **4** 238–252.

Berneau, B. 1964. Programme de risque minimal en programmation linéaire stochastique. *Comptes Rendus de Academie des Sciences Paris* **259**(981-983).

Birge, J. R., F. V. Louveaux. 1997. *Introduction to Stochastic Programming*. Springer, New York.

Caroe, C. 1998. Decomposition in stochastic integer programming. Ph.D. thesis, University of Copenhagen.

Charnes, A., W. W. Cooper. 1963. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations Research* **11**(1) pp. 18–39.

Dantzig, G. 1955. Linear programming under uncertainty. *Management Science* **1**(3/4) 197–206.

Dantzig, G.B. 1963. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey.

Dantzig, G.B., P. Wolfe. 1961. The decomposition algorithm for linear programs. *Econometrica* **29** 767–778.

Fishburn, P.C. 1977. Mean-risk analysis with risk associated with below target returns. *The American Economic Review* **67** 116–126.

Higle, Julie, Suvrajeet Sen. 1996. *Stochastic Decomposition*. Kluwer Academic Publishers, 101 Phillip Drive, Norwell, MA 02061.

IBM. 2009. *CPLEX 12.1 IBM ILOG CPLEX Callable Library Version 12.1 C API Reference Manual*. IBM, USA.

Kelley, J. 1960. The cutting-plane method for solving convex programs. *SIAM Journal of Applied Mathmatics* **8**(4) 703–712.

Kleywegt, A., A. Shapiro, T. Homem-de Mello. 2001. The sample average approximation method for stochastic discrete optimization. *SIAM J. on Optimization* **12**(2) 479–502.

Konno, Hiroshi, Hiroaki Yamazaki. 1991. Mean-absolute deviation portfolio optimization model and its applications to tokyo stock market. *Management Science* **37**(5) 519–531.

Kristoffersen, T. 2005. Deviation measures in linear two-stage stochastic programming. *Mathematical Methods of Operations Research* **62(2)** 255–274.

Krokhmal, P., M. Zabarankin, S. Uryasev. 2011. Modeling and optimization of risk. *Surveys in Operations Research and Managment Science* **16** 49–66.

Linderoth, J., S. Wright. 2003. Decomposition algorithms for stochastic programming on a computational grid. *Computational Optimization and Applications* **24(2)** 207–250.

Linderoth, J.T., A. Shapiro, S.J. Wright. 2006. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research* **142** 215–241.

Louveaux, F., Y Smeers. 1998. *Optimal Investments for Electricity Generation: A Stochastic Model and a Test Problem*, Chap. Numerical techniques for stochastic optimization problems. Springer-Verlag, Berlin, 445-452.

Mak, W.K., D.P. Morton, R.K. Wood. 1999. Monte carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters* **24** 47–56.

Markert, Andreas, Rüdiger Schultz. 2005. On deviation measures in stochastic integer programming. *Operations Research Letters* **33**(5) 441–449.

Markowitz, H. 1959. *Porfolio Selection*. John Wiley & Sons, New York.

Markowitz, H. 1987. *Mean-Variance Analysis in Portfolio Choice and Capital Markets*. Blackwell, Oxford, UK.

Markowitz, H.M. 1952. Portfolio selection. *Journal of Finance* **7**(1) 77–91.

Miller, N. 2008. Mean-risk portfolio optimization problems with risk-adjusted measures. dissertation, The State University of New Jersey.

Mulvey, J.M, A. Ruszczynski. 1995. A new scenario decomposition method for large scale stochasitc optimization. *Operations Research* **43** 477–490.

Nemhauser, G., L. Wolsey. 1999. *Integer and Combinatorial Optimization*. Discrete Matehmatics and Optimization, Wiley- Interscience.

Ogryczak, W., A. Ruszcynski. 1999. From stochastic dominance to mean-risk model: Semideviations as risk measures. *European Journal of Operational Research* **116** 33–50.

Ogryczak, W., A. Ruszcynski. 2001. On consistency of stochastic dominance and mean-semideviation models. *Mathematical Programming* **89** 217–232.

Ogryczak, W., A. Ruszcynski. 2002. Dual stochastic dominance and related mean-risk models. *SIAM Journal on Optimization* **13** 60–78.

Ogryczak, Wlodzimierz. 2000. Multiple criteria linear programmig model for portfolio selection. *Annals of Operations Research* **97** 143–162.

Porter, R. Burr. 1974. Semivariance and stochastic dominance: A comparison. *American Economic Review* **64**(1) 200 – 204.

Rockafellar, R. T., S. Uryasev. 2000. Optimization of conditional value-at-risk. *The Journal at Risk* **2** 21–41.

Rockafellar, R. T., R. J-B Wets. 1991. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research* (119-147).

Ruszcynski, A., A. Shapiro. 2006. Optimization of convex risk functions. *Mathematics of Operations Research* **31**(3) 433–452.

Ruszczynski, A., A. Shapiro, eds. 2003. *Stochastic Programming*, *Handbooks in Operations Research and Management Science*, vol. 10. North Holland.

Schultz, R. 2003. Stochastic programming with integer variables. *Mathematical Programming* **97** 285–309.

Schultz, Rüdiger, Stephan Tiedemann. 2003. Risk aversion via excess probabilities in stochastic programs with mixed-integer recourse. *SIAM J. on Optimization* **14(1)** 115–138.

Sen, S., R.D. Doverspike, S. Cosares. 1994. Network planning with random demand. *Telecommunications Systems* **3**(11-30).

Shapiro, Alexander, Shabbir Ahmed. 2004. On a class of minimax stochastic programs. *SIAM Journal on Optimization* 1237–1249.

Slyke, R. Van, R.-B. Wets. 1969. L-shaped linear programs with application to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* **17** 638–663.

Yitzhaki, Shlomo. 1982. Stochastic dominance, mean variance, and gini's mean difference. *The American Economic Review* **72**(1) pp. 178–185.

Young, Martin R. 1998. A minimax portfolio selection rule with linear programming solution. *Management Science* **44**(5) pp. 673–683.

APPENDIX A

ADDITIONAL COMPUTATIONAL RESULTS

A.1 Detailed Results Tables

This appendix contains the following abbreviations in table headings: $\mathbb{E}[f(\cdot)] := \mathbb{E}[f(x,\tilde{\omega})]$, $\phi_{ASD} := \phi_{ASD}(x)$, $\mathbb{E}[\nu(\omega)] := \sum_{\omega\in\Omega} p(\omega)\nu(\omega)$, run := total runtime of algorithm in seconds.

Table A.1: Results of ASD-AGG for cep1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	355160	355160	528362	355160	0.00	2	0.03	0.343
0.1	372480	355160	528362	319644	52836.2	2	0.05	0.406
0.2	389800	355160	528362	284128	105672	2	0.02	0.312
0.3	407120	355160	528362	248612	158508	2	0.00	0.297
0.4	424441	355160	528362	213096	211345	2	0.02	0.328
0.5	441761	355160	528362	177580	264181	2	0.03	0.328
0.6	459081	355160	528362	142064	317017	2	0.02	0.313
0.7	476401	355160	528362	106548	369853	2	0.02	0.344
0.8	493721	355160	528362	71032	422689	2	0.05	0.375
0.9	511041	355160	528362	35516	475525	2	0.02	0.359
1	528362	355160	528362	0	528362	2	0.02	0.437

Table A.2: Results of ASD-AGG for cep1a

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	394890	394890	566858	394890	0.0	2	0.00	0.328
0.1	412087	394890	566858	355401	56685.8	2	0.05	0.297
0.2	429283	394890	566858	315912	113372	2	0.02	0.547
0.3	446480	394890	566858	276423	170057	2	0.03	0.297
0.4	463677	394890	566858	236934	226743	2	0.03	0.312
0.5	480874	394890	566858	197445	283429	2	0.05	0.312
0.6	498071	394890	566858	157956	340115	2	0.03	0.297
0.7	515267	394890	566858	118467	396801	2	0.03	0.328
0.8	532464	394890	566858	78978	453486	2	0.02	0.296
0.9	549661	394890	566858	39489	510172	2	0.03	0.297
1	566858	394890	566858	0	566858	2	0.02	0.312

Table A.3: Results of ASD-AGG for cep1sk

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	320344	320344	476156	320344	0.0	6	0.05	0.610
0.1	335925	320344	476156	288310	47615.6	6	0.06	0.657
0.2	351506	320344	476156	256275	95231.2	6	0.02	0.578
0.3	367088	320344	476156	224241	142847.0	6	0.00	0.766
0.4	382669	320344	476156	192206	190462.0	6	0.02	0.657
0.5	398250	320344	476156	160172	238078.0	6	0.08	0.64
0.6	413831	320344	476156	128138	285694.0	6	0.05	0.625
0.7	429412	320344	476156	96103.2	333309.0	6	0.08	0.625
0.8	444993	320344	476156	64068.8	380925.0	6	0.03	0.594
0.9	460575	320344	476156	32034.4	428540.0	6	0.06	0.625
1	476156	320344	476156	0	476156.0	6	0.02	0.594

Table A.4: Results of ASD-SEP for cep1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	355160	355160	528362	355160	0.0	2	0.03	0.500
0.1	372480	355160	528362	319644	52836.2	2	0.03	0.438
0.2	389800	355160	528362	284128	105672.0	2	0.02	0.438
0.3	407120	355160	528362	248612	158508.0	2	0.02	0.359
0.4	424441	355160	528362	213096	211345.0	2	0.03	0.406
0.5	441761	355160	528362	177580	264181.0	2	0.02	0.360
0.6	459081	355160	528362	142064	317017.0	2	0.03	0.391
0.7	476401	355160	528362	106548	369853.0	2	0.03	0.359
0.8	493721	355160	528362	71032	422689.0	2	0.03	0.375
0.9	511041	355160	528362	35516	475525.0	2	0.03	0.406
1	528362	355160	528362	0	528362.0	2	0.00	0.422

Table A.5: Results of ASD-SEP for cep1a

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	394890	394890	566858	394890	0	2	0.02	0.672
0.1	412087	394890	566858	355401	56685.8	2	0.00	0.406
0.2	429283	394890	566858	315912	113372	2	0.02	0.360
0.3	446480	394890	566858	276423	170057	2	0.00	0.407
0.4	463677	394890	566858	236934	226743	2	0.00	0.36
0.5	480874	394890	566858	197445	283429	2	0.03	0.375
0.6	498071	394890	566858	157956	340115	2	0.02	0.328
0.7	515267	394890	566858	118467	396801	2	0.03	0.406
0.8	532464	394890	566858	78978	453486	2	0.03	0.328
0.9	549661	394890	566858	39489	510172	2	0.03	0.344
1	566858	394890	566858	0	566858	2	0.03	0.407

Table A.6: Results of ASD-SEP for cep1sk

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	320344	320344	476156	320344	0	6	0.05	0.890
0.1	335925	320344	476156	288310	47615.6	6	0.05	0.891
0.2	351506	320344	476156	256275	95231.2	6	0.06	0.859
0.3	367088	320344	476156	224241	142847	6	0.08	0.844
0.4	382669	320344	476156	192206	190462	6	0.06	0.844
0.5	398250	320344	476156	160172	238078	6	0.06	0.859
0.6	413831	320344	476156	128138	285693	6	0.03	0.890
0.7	429412	320344	476156	96103.2	333309	6	0.03	0.843
0.8	444993	320344	476156	64068.8	380925	6	0.08	0.843
0.9	460575	320344	476156	32034.4	428540	6	0.02	0.875
1	476156	320344	476156	0	476156	6	0.08	0.797

Table A.7: Results of QDEV-AGG for cep1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	355160	355160	355160	0	355160	0.0	0	2	0.000	0.031
0.1	387111	355160	248547	177580	319644	49709.3	17758	3	0.031	0.047
0.2	419063	355160	248547	177580	284128	99418.7	35516	3	0.047	0.062
0.3	451014	355160	248547	177580	248612	149128.0	53274	3	0.015	0.063
0.4	482965	355160	248547	177580	213096	198837.0	71032	3	0.031	0.062
0.5	514917	355160	248547	177580	177580	248547.0	88790	3	0.031	0.063
0.6	546868	355160	248547	177580	142064	298256.0	106548	3	0.031	0.062
0.7	578819	355160	248547	177580	106548	347965.0	124306	3	0.000	0.047
0.8	610771	355160	248547	177580	71032	397675.0	142064	3	0.016	0.046
0.9	642722	355160	248547	177580	35516	447384.0	159822	3	0.031	0.047
1.0	674673	423446	248547	177580	0	497093.0	177580	3	0.046	0.046

Table A.8: Results of QDEV-AGG for cep1a

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	394890	394890	394890	0	394890	0.0	0	2	0.031	0.031
0.1	428262	394890	261046	206517	355401	52209.1	20652	6	0.078	0.078
0.2	461633	394890	260521	207567	315912	104208.0	41513	7	0.062	0.093
0.3	495005	394890	260521	207567	276423	156312.0	62270	7	0.094	0.094
0.4	528377	394890	260521	207567	236934	208416.0	83027	7	0.078	0.110
0.5	561749	394890	260521	207567	197445	260521.0	103783	7	0.094	0.109
0.6	595121	394890	260521	207567	157956	312625.0	124540	7	0.079	0.094
0.7	628492	394890	260521	207567	118467	364729.0	145297	7	0.079	0.094
0.8	661864	394890	260521	207567	78978	416833.0	166053	7	0.078	0.093
0.9	695236	394890	260521	207567	39489	468937.0	186810	7	0.078	0.094
1.0	728608	466076	260521	207567	0	521041.0	207567	7	0.047	0.063

Table A.9: Results of QDEV-AGG for cep1sk

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	320344	320344	320344	0	320344	0	0	6	0.048	0.079
0.1	348956	320344	213560	179342	288310	42712	17934	11	0.063	0.141
0.2	377568	320344	213561	179340	256275	85424.5	35868	12	0.142	0.204
0.3	406179	320344	213562	179340	224241	128137	53802	12	0.156	0.156
0.4	434791	320344	213566	179331	192207	170853	71732	12	0.109	0.156
0.5	463403	320344	213565	179333	160172	213565	89667	13	0.126	0.172
0.6	492015	320344	213563	179338	128138	256275	107603	12	0.140	0.156
0.7	520627	320345	213564	179336	96103.4	298990	125535	12	0.126	0.156
0.8	549238	320345	213564	179336	64068.9	341702	143469	13	0.124	0.171
0.9	577850	320345	213564	179337	32034.5	384415	161403	14	0.126	0.188
1.0	606462	392015	213564	179337	0	427127	179337	12	0.093	0.125

Table A.10: Results of QDEV-SEP for cep1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	355160	355160	0	57800000	355160	0	0	2	0.016	0.109
0.1	387111	355160	248547	177580	319644	49709.3	17758	3	0.015	0.079
0.2	419063	355160	248547	177580	284128	99418.7	35516	3	0.063	0.063
0.3	451014	355160	248547	177580	248612	149128	53274	3	0.047	0.063
0.4	482965	355160	248547	177580	213096	198837	71032	3	0.00	0.079
0.5	514917	355160	248547	177580	177580	248547	88790	3	0.032	0.078
0.6	546868	355160	248547	177580	142064	298256	106548	3	0.031	0.078
0.7	578819	355160	248547	177580	106548	347965	124306	3	0.015	0.062
0.8	610771	355160	248547	177580	71032	397675	142064	3	0.015	0.078
0.9	642722	355160	248547	177580	35516	447384	159822	3	0.047	0.062
1.0	674673	355160	248547	177580	0	497093	177580	3	0.031	0.078

Table A.11: Results of QDEV-SEP for cep1a

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	394890	394890	0	5.44E+08	394890	0	0	2	0.000	0.063
0.1	428262	394890	261046	206517	355401	52209.1	20651.7	6	0.109	0.109
0.2	461633	394890	260521	207567	315912	104208	41513.3	7	0.094	0.125
0.3	495005	394890	260521	207567	276423	156312	62270	7	0.015	0.141
0.4	528377	394890	260521	207567	236934	208416	83026.7	7	0.094	0.172
0.5	561749	394890	260521	207567	197445	260521	103783	7	0.093	0.125
0.6	595121	394890	260521	207567	157956	312625	124540	7	0.032	0.14
0.7	628492	394890	260521	207567	118467	364729	145297	7	0.079	0.125
0.8	661864	394890	260521	207567	78978	416833	166053	7	0.062	0.125
0.9	695236	394890	260521	207567	39489	468937	186810	7	0.094	0.125
1.0	728608	394890	260521	207567	0	521041	207567	7	0.063	0.140

Table A.12: Results of QDEV-SEP for cep1sk

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	320344	320344	293118	28593.9	320344	0	0	6	0.030	0.109
0.1	348956	320344	213560	179342	288310	42712	17934.2	9	0.000	0.172
0.2	377568	320344	213643	179175	256275	85457.4	35835	9	0.141	0.157
0.3	406179	320344	213643	179175	224241	128186	53752.5	9	0.141	0.157
0.4	434791	320344	213643	179175	192206	170915	71670	9	0.078	0.172
0.5	463403	320344	213643	179175	160172	213643	89587.5	9	0.062	0.187
0.6	492015	320344	213643	179175	128138	256372	107505	11	0.109	0.187
0.7	520627	320344	213643	179175	96103.2	299101	125422	11	0.108	0.187
0.8	549238	320344	213643	179175	64068.8	341830	143340	11	0.172	0.188
0.9	577850	320345	213564	179337	32034.5	384415	161404	12	0.172	0.203
1.0	606462	320345	213564	179337	0	427127	179337	12	0.047	0.219

Table A.13: Results of ASD-AGG for pgp2

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	447.32	447.32	474.00	447.32	0.00	35	0.55	3.28
0.1	449.99	447.32	474.00	402.59	47.40	30	0.55	3.03
0.2	452.66	447.32	474.00	357.86	94.80	29	0.53	2.58
0.3	455.33	447.32	474.00	313.13	142.20	37	0.69	3.33
0.4	457.99	447.32	474.00	268.40	189.60	33	0.42	2.89
0.5	460.66	447.33	474.00	223.66	237.00	35	0.53	3.09
0.6	463.28	447.60	473.74	179.04	284.25	35	0.53	3.09
0.7	465.90	447.60	473.74	134.28	331.62	32	0.53	2.89
0.8	468.51	447.60	473.74	89.52	378.99	37	0.61	3.28
0.9	471.12	447.90	473.70	44.79	426.33	35	0.62	3.14
1.0	473.70	447.90	473.70	0.00	473.70	35	0.68	3.17

Table A.14: Results of ASD-AGG for pgp2e

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	413.94	413.94	464.08	413.94	0.00	39	0.48	3.33
0.1	418.95	413.94	464.08	372.54	46.41	36	0.56	3.06
0.2	423.96	413.94	464.08	331.15	92.82	34	0.47	3.09
0.3	428.98	413.94	464.08	289.76	139.22	36	0.47	3.19
0.4	433.97	414.00	463.93	248.40	185.57	37	0.50	3.41
0.5	438.97	414.00	463.93	207.00	231.96	40	0.48	3.48
0.6	443.91	414.28	463.66	165.71	278.20	38	0.57	3.49
0.7	448.85	414.28	463.66	124.29	324.56	41	0.61	3.50
0.8	453.73	415.06	463.40	83.01	370.72	43	0.67	3.95
0.9	458.48	416.42	463.15	41.64	416.83	37	0.50	3.31
1.0	463.15	416.42	463.15	0.00	463.15	36	0.46	3.14

Table A.15: Results of ASD-AGG for pgp2f

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	457.03	457.03	512.15	457.03	0.00	35	0.36	3.13
0.1	462.54	457.03	512.15	411.32	51.22	33	0.39	2.97
0.2	468.05	457.03	512.15	365.62	102.43	32	0.41	3.33
0.3	473.54	457.15	511.78	320.00	153.54	38	0.49	3.23
0.4	479.00	457.15	511.78	274.29	204.71	37	0.53	3.14
0.5	484.47	457.15	511.78	228.57	255.89	40	0.64	3.84
0.6	489.83	458.02	511.04	183.21	306.62	39	0.57	3.58
0.7	495.12	458.14	510.97	137.44	357.68	40	0.61	3.70
0.8	500.41	458.14	510.97	91.63	408.78	37	0.47	3.36
0.9	505.67	458.46	510.92	45.85	459.83	40	0.58	3.52
1.0	510.86	459.40	510.86	0.00	510.86	43	0.65	3.94

Table A.16: Results of ASD-SEP for pgp2

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	447.32	447.32	474.00	447.32	0.00	35	0.63	4.78
0.1	449.99	447.33	474.00	402.59	47.40	29	0.48	4.03
0.2	452.66	447.32	474.00	357.86	94.80	29	0.39	3.95
0.3	455.33	447.32	474.00	313.13	142.20	29	0.50	4.02
0.4	458.00	447.33	474.00	268.40	189.60	30	0.51	4.42
0.5	460.66	447.32	474.00	223.66	237.00	34	0.56	4.75
0.6	463.28	447.60	473.74	179.04	284.25	31	0.50	4.48
0.7	465.90	447.60	473.74	134.28	331.62	31	0.51	5.34
0.8	468.51	447.60	473.74	89.52	378.99	31	0.64	4.81
0.9	471.12	447.90	473.70	44.79	426.33	32	0.68	4.67
1.0	473.70	447.90	473.70	0.00	473.70	35	0.67	5.14

Table A.17: Results of ASD-SEP for pgp2e

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	413.94	413.94	464.08	413.94	0.00	39	0.55	5.52
0.1	418.95	413.94	464.08	372.54	46.41	34	0.47	4.55
0.2	423.96	413.94	464.08	331.15	92.82	33	0.47	4.44
0.3	428.98	413.94	464.08	289.76	139.22	32	0.47	4.27
0.4	433.97	414.00	463.93	248.40	185.57	34	0.56	4.61
0.5	438.97	414.00	463.93	207.00	231.96	36	0.55	4.95
0.6	443.91	414.28	463.66	165.71	278.20	33	0.47	4.86
0.7	448.85	414.28	463.66	124.29	324.56	36	0.52	9.23
0.8	453.73	415.06	463.40	83.01	370.72	36	0.42	4.84
0.9	458.48	416.42	463.15	41.64	416.83	40	0.63	5.50
1.0	463.15	416.42	463.15	0.00	463.15	36	0.48	4.69

Table A.18: Results of ASD-SEP for pgp2f

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	457.03	457.03	512.15	457.03	0.00	35	0.39	4.88
0.1	462.54	457.03	512.15	411.32	51.22	34	0.47	5.22
0.2	468.05	457.03	512.15	365.62	102.43	31	0.45	4.53
0.3	473.54	457.15	511.78	320.00	153.54	28	0.42	4.30
0.4	479.00	457.15	511.78	274.29	204.71	32	0.41	4.67
0.5	484.47	457.15	511.78	228.57	255.89	33	0.45	4.84
0.6	489.83	458.02	511.04	183.21	306.62	34	0.47	4.92
0.7	495.12	458.14	510.97	137.44	357.68	32	0.48	4.70
0.8	500.41	458.14	510.97	91.63	408.78	30	0.35	4.84
0.9	505.67	458.46	510.92	45.85	459.83	32	0.46	4.86
1.0	510.86	459.40	510.86	0.00	510.86	43	0.63	6.34

Table A.19: Results of QDEV-AGG for pgp2

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	447.324	447.324	447.324	0	447.324	0	0	35	0.626	0.797
0.1	452.638	447.324	28.482	443.5	402.592	5.69634	44.35	42	0.766	0.969
0.2	457.952	447.324	28.484	443.495	357.859	11.3937	88.699	54	1.061	1.25
0.3	463.234	447.597	27.510	444.7	313.318	16.5062	133.41	54	0.873	1.234
0.4	468.447	447.597	27.515	444.691	268.558	22.0121	177.876	51	0.813	1.188
0.5	473.624	447.896	27.025	445.301	223.948	27.0252	222.651	48	0.735	1.109
0.6	478.770	447.896	27.026	445.3	179.159	32.4309	267.18	50	0.940	1.125
0.7	483.915	447.896	27.026	445.3	134.369	37.8361	311.71	54	0.969	1.296
0.8	489.037	448.138	26.981	445.3	89.6277	43.1694	356.24	48	0.783	1.063
0.9	494.149	448.138	26.981	445.3	44.8138	48.5656	400.77	49	0.831	1.125
1.0	499.259	469.253	26.979	445.3	0	53.9587	445.3	66	1.174	1.469

Table A.20: Results of QDEV-AGG for pgp2e

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	413.935	413.935	413.935	0	413.935	0	0	39	0.503	0.766
0.1	423.571	413.936	61.970	386.349	372.543	12.394	38.6349	55	0.891	1.094
0.2	433.201	414.003	61.747	386.5	331.203	24.6989	77.3	64	0.844	1.266
0.3	442.733	414.284	61.307	386.5	289.999	36.7844	115.95	61	0.800	1.235
0.4	452.170	415.062	60.066	387.7	249.037	48.0531	155.08	62	0.829	1.266
0.5	461.410	415.408	59.556	388.3	207.704	59.5563	194.15	63	0.985	1.297
0.6	470.428	416.419	57.567	391.3	166.567	69.0806	234.78	54	0.797	1.109
0.7	479.330	417.841	56.691	392.3	125.352	79.3679	274.61	61	0.889	1.219
0.8	488.108	419.171	55.021	395.3	83.8342	88.0333	316.24	66	1.044	1.313
0.9	496.702	420.820	54.317	396.5	42.082	97.7703	356.85	66	0.891	1.344
1.0	505.030	450.983	53.965	397.1	0	107.93	397.1	51	0.766	1.031

Table A.21: Results of QDEV-AGG for pgp2f

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	457.026	457.026	457.026	0	457.026	0	0	35	0.375	0.672
0.1	467.798	457.026	67.293	430.153	411.324	13.4586	43.0153	53	0.594	1.031
0.2	478.491	457.148	67.205	429.45	365.719	26.8819	85.89	60	0.766	1.156
0.3	489.151	457.318	66.885	429.666	320.123	40.1311	128.9	59	0.873	1.156
0.4	499.526	458.144	64.324	432.95	274.887	51.4595	173.18	60	0.891	1.234
0.5	509.771	459.396	64.198	431.75	229.698	64.1982	215.875	65	0.986	1.265
0.6	519.786	460.084	63.918	431.75	184.034	76.7018	259.05	52	0.700	1.032
0.7	529.736	460.084	63.919	431.75	138.025	89.4861	302.225	59	0.906	1.156
0.8	539.686	460.084	63.918	431.75	92.0168	102.269	345.4	59	0.860	1.156
0.9	549.547	462.837	62.193	434.796	46.2837	111.947	391.317	57	0.906	1.125
1.0	559.067	498.005	60.884	437.3	0	121.767	437.3	63	0.843	1.234

Table A.22: Results of QDEV-SEP for pgp2

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	447.324	447.324	28.4817	443.5	447.324	0	0	35	0.624	1.032
0.1	452.638	447.324	28.4817	443.5	402.592	5.69634	44.35	30	0.563	0.828
0.2	457.952	447.324	28.4817	443.5	357.859	11.3927	88.7	31	0.500	0.812
0.3	463.234	447.597	27.5112	444.699	313.318	16.5067	133.41	31	0.531	0.828
0.4	468.446	447.597	27.5103	444.7	268.558	22.0083	177.88	35	0.578	0.922
0.5	473.624	447.896	27.0258	445.3	223.948	27.0258	222.65	37	0.685	0.968
0.6	478.77	447.896	27.0258	445.3	179.159	32.4309	267.18	37	0.750	1
0.7	483.915	447.896	27.0258	445.3	134.369	37.8361	311.71	41	0.747	1.157
0.8	489.037	448.138	26.9809	445.3	89.6277	43.1694	356.24	36	0.626	0.984
0.9	494.149	448.138	26.9809	445.3	44.8138	48.5656	400.77	41	0.750	1.125
1	499.259	448.738	26.9793	445.3	0	53.9586	445.3	53	0.983	1.406

Table A.23: Results of QDEV-SEP for pgp2e

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	413.935	413.935	58.6442	393.485	413.935	0	0	39	0.643	0.938
0.1	423.57	413.935	62.1635	385.96	372.542	12.4327	38.596	32	0.380	0.812
0.2	433.202	414.003	61.7663	386.463	331.203	24.7065	77.2925	34	0.485	0.844
0.3	442.733	414.284	61.3263	386.463	289.999	36.7958	115.939	40	0.563	1.015
0.4	452.17	415.062	60.0663	387.7	249.037	48.0531	155.08	37	0.543	0.938
0.5	461.41	415.408	59.5563	388.3	207.704	59.5563	194.15	40	0.563	1.031
0.6	470.428	416.419	57.5671	391.3	166.567	69.0806	234.78	45	0.627	1.141
0.7	479.33	417.841	56.6913	392.3	125.352	79.3679	274.61	43	0.642	1.141
0.8	488.108	419.171	55.0208	395.3	83.8342	88.0333	316.24	56	0.812	1.437
0.9	496.702	420.82	54.3168	396.5	42.082	97.7703	356.85	52	0.676	1.328
1.0	505.03	421.796	53.9649	397.1	0	107.93	397.1	59	0.910	1.531

Table A.24: Results of QDEV-SEP for pgp2f

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	457.026	457.026	137.229	325.95	457.026	0	0	35	0.468	0.813
0.1	467.798	457.026	67.6448	429.45	411.324	13.529	42.945	39	0.548	0.969
0.2	478.491	457.148	67.2049	429.45	365.719	26.8819	85.89	38	0.565	0.984
0.3	489.151	457.321	66.9849	429.45	320.125	40.191	128.835	38	0.577	0.937
0.4	499.526	458.143	63.9762	433.648	274.886	51.1809	173.459	41	0.454	1.047
0.5	509.771	459.396	64.1983	431.75	229.698	64.1983	215.875	47	0.674	1.141
0.6	519.786	460.084	63.919	431.749	184.034	76.7028	259.049	39	0.517	1.016
0.7	529.736	460.084	63.9182	431.75	138.025	89.4855	302.225	39	0.497	0.953
0.8	539.686	460.084	63.9182	431.75	92.0168	102.269	345.4	46	0.654	1.203
0.9	549.546	462.838	62.1904	434.8	46.2838	111.943	391.32	46	0.689	1.171
1.0	559.067	464.043	60.8836	437.3	0	121.767	437.3	59	0.842	1.469

Table A.25: Results of ASD-AGG for gbd

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	1655.63	1655.63	1906.85	1655.63	0	30	294.829	422.250
0.1	1680.75	1655.63	1906.85	1490.07	190.685	32	238.422	366.593
0.2	1705.87	1655.63	1906.85	1324.5	381.37	31	233.285	357.812
0.3	1731	1655.63	1906.85	1158.94	572.056	35	261.731	404.516
0.4	1756.12	1655.63	1906.85	993.377	762.741	30	221.552	343.407
0.5	1781.23	1655.71	1906.76	827.856	953.378	32	239.11	368.485
0.6	1806.34	1655.71	1906.76	662.285	1144.05	32	241.464	370.781
0.7	1831.44	1655.71	1906.76	496.714	1334.73	31	232.881	358.547
0.8	1856.54	1656.01	1906.67	331.201	1525.34	41	311.442	475.437
0.9	1881.57	1656.57	1906.57	165.657	1715.91	39	294.747	452.359
1.0	1906.55	1656.77	1906.55	0	1906.55	36	270.494	414.844

Table A.26: Results of ASD-AGG for gbd-sk3

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	2371.28	2371.28	2780.27	2371.28	0	27	189.47	299.984
0.1	2412.07	2372.61	2767.24	2135.35	276.724	22	150.753	241.719
0.2	2451.53	2372.61	2767.24	1898.09	553.447	26	174.36	282.110
0.3	2491	2372.61	2767.23	1660.83	830.17	27	180.778	289.281
0.4	2529.35	2399.03	2724.82	1439.42	1089.93	32	210.063	339.579
0.5	2558.24	2425.57	2690.9	1212.79	1345.45	38	245.267	397.031
0.6	2583.17	2440.39	2678.36	976.157	1607.02	35	280.663	426.907
0.7	2606.05	2452.41	2671.9	735.722	1870.33	36	229.411	374.453
0.8	2627.44	2458.32	2669.72	491.663	2135.78	18	119.345	193.297
0.9	2648.58	2458.32	2669.72	245.832	2402.75	19	154.023	235.468
1.0	2669.72	2458.32	2669.72	0	2669.72	18	119.199	193.421

Table A.27: Results of ASD-SEP for gbd

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	1655.63	1655.63	1906.85	1655.63	0	30	300.54	423.750
0.1	1680.75	1655.63	1906.85	1490.07	190.685	30	224.428	345.562
0.2	1705.87	1655.63	1906.85	1324.5	381.37	29	214.669	333.969
0.3	1731	1655.63	1906.85	1158.94	572.056	32	233.241	363.610
0.4	1756.12	1655.63	1906.85	993.377	762.741	30	223.436	344.172
0.5	1781.23	1655.71	1906.76	827.856	953.378	32	237.143	365.047
0.6	1806.34	1655.71	1906.76	662.285	1144.05	31	232.056	359.328
0.7	1831.44	1655.72	1906.76	496.715	1334.73	31	233.957	359.829
0.8	1856.54	1655.99	1906.67	331.197	1525.34	34	257.579	396.781
0.9	1881.57	1656.58	1906.57	165.658	1715.91	32	239.108	369.985
1.0	1906.55	1656.77	1906.55	0	1906.55	36	270.532	414.219

Table A.28: Results of ASD-AGG for gbd_sk3

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	2371.28	2371.28	2780.27	2371.28	0	27	190.902	300.625
0.1	2412.07	2372.61	2767.23	2135.35	276.723	22	181.181	275.813
0.2	2451.53	2372.61	2767.23	1898.09	553.447	22	146.118	236.547
0.3	2491	2372.61	2767.24	1660.83	830.171	22	149.585	237.969
0.4	2529.35	2399.06	2724.78	1439.43	1089.91	18	118.173	191.375
0.5	2558.24	2425.53	2690.94	1212.77	1345.47	28	183.284	305.219
0.6	2583.17	2440.39	2678.36	976.155	1607.02	29	185.734	303.046
0.7	2606.05	2452.68	2671.78	735.803	1870.25	30	193.150	315.313
0.8	2627.44	2458.32	2669.72	491.663	2135.78	17	112.961	183.610
0.9	2648.58	2458.32	2669.72	245.832	2402.75	17	112.780	183.187
1.0	2669.72	2458.32	2669.72	0	2669.72	18	120.311	194.094

Table A.29: Results of QDEV-AGG for gbd

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	1655.63	1655.63	1655.63	0	1655.63	0	0	30	286.943	514.084
0.1	1703.3	1655.63	338.392	1455.58	1490.07	67.6784	145.558	50	493.504	872.624
0.2	1750.96	1655.71	339.254	1453.44	1324.57	135.702	290.689	48	472.165	830.249
0.3	1798.58	1655.71	339.556	1452.83	1159	203.734	435.85	55	552.799	953.669
0.4	1846.01	1657.69	334.303	1459.87	994.615	267.443	583.948	56	554.437	972.051
0.5	1893.08	1657.69	334.548	1459.38	828.847	334.548	729.689	61	596.479	1041.730
0.6	1939.95	1658.92	335.211	1456.88	663.567	402.253	874.129	59	578.395	1022.430
0.7	1986.74	1660.53	334.687	1457.18	498.159	468.562	1020.02	60	564.077	1001.630
0.8	2033.34	1660.54	334.539	1457.47	332.109	535.262	1165.97	59	509.858	941.851
0.9	2078.99	1675.98	331.435	1460.9	167.598	596.583	1314.81	69	603.322	1104.240
1.0	2123.29	1794.24	329.017	1465.26	0	658.034	1465.26	59	505.822	941.701

Table A.30: Results of QDEV-AGG for gbd_sk3

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	2371.28	2371.28	2371.28	0	2371.28	0	0	27	255.964	460.647
0.1	2451.03	2372.61	303.31	2550.19	2135.35	60.6619	255.019	36	329.771	602.426
0.2	2528.85	2394.05	294.984	2478.08	1915.24	117.994	495.615	46	416.832	760.814
0.3	2583.11	2439.55	230.764	2456.56	1707.69	138.459	736.969	46	409.156	754.345
0.4	2627.43	2458.32	208.494	2464.11	1474.99	166.795	985.644	36	320.819	592.161
0.5	2669.71	2458.32	209.215	2462.67	1229.16	209.215	1231.33	33	297.021	544.505
0.6	2711.93	2459.02	210.019	2460.5	983.608	252.023	1476.3	33	290.265	541.239
0.7	2754.08	2459.02	210.037	2460.46	737.706	294.052	1722.32	33	285.485	533.303
0.8	2796.23	2459.02	210.035	2460.47	491.804	336.056	1968.37	33	284.158	534.037
0.9	2838.38	2459.63	209.964	2460.54	245.963	377.935	2214.49	38	336.293	623.535
1.0	2880.27	2666.43	213.832	2452.61	0	427.664	2452.61	34	297.266	557.208

Table A.31: Results of QDEV-SEP for gbd

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	1655.63	1655.63	763.756	892.009	1655.63	0	0	30	226.780	363.235
0.1	1703.3	1655.63	339.081	1454.2	1490.07	67.8161	145.42	29	220.867	351.125
0.2	1750.96	1655.71	339.058	1453.83	1324.57	135.623	290.766	35	272.998	428.250
0.3	1798.58	1655.71	339.555	1452.84	1159	203.733	435.852	35	269.729	426.062
0.4	1846	1657.69	336.14	1456.19	994.616	268.912	582.478	38	290.306	459.297
0.5	1893.08	1657.7	335.482	1457.51	828.852	335.482	728.753	37	286.623	449.203
0.6	1939.95	1658.92	334.737	1457.83	663.567	401.684	874.698	38	286.787	455.454
0.7	1986.74	1660.54	334.811	1456.92	498.162	468.736	1019.84	39	288.224	461.563
0.8	2033.34	1660.54	334.707	1457.13	332.108	535.531	1165.7	48	343.870	555.093
0.9	2078.99	1675.88	331.461	1460.85	167.588	596.63	1314.77	56	385.693	630.766
1.0	2123.29	1684.57	329.017	1465.26	0	658.034	1465.26	59	400.099	660.860

Table A.32: Results of QDEV-SEP for gbd_sk3

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	2371.28	2371.28	131.828	3001.58	2371.28	0	0	27	199.567	319.750
0.1	2451.03	2372.61	307.426	2541.96	2135.35	61.4853	254.196	24	168.215	274.828
0.2	2528.85	2393.21	294.127	2483.17	1914.56	117.651	496.634	34	234.660	384.281
0.3	2583.11	2439.43	231.416	2455.54	1707.6	138.85	736.662	36	246.844	403.844
0.4	2627.43	2458.32	207.604	2465.89	1474.99	166.083	986.357	24	162.863	271.907
0.5	2669.71	2458.32	207.833	2465.43	1229.16	207.833	1232.72	27	186.036	307.281
0.6	2711.93	2459.02	210.27	2460	983.608	252.324	1476	23	187.931	292.438
0.7	2754.08	2459.02	210.151	2460.24	737.706	294.211	1722.17	23	158.949	264.063
0.8	2796.23	2459.02	210.854	2458.83	491.804	337.367	1967.06	26	178.973	294.266
0.9	2838.38	2459.69	210.28	2459.9	245.969	378.505	2213.91	32	217.916	356.172
1.0	2880.27	2463.02	213.832	2452.61	0	427.664	2452.61	34	231.241	379.766

Table A.33: Results of ASD-AGG for LandS

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	225.629	225.629	249.65	225.629	0	31	448.55	695.928
0.1	228.026	225.637	249.531	203.073	24.9531	33	405.118	576.790
0.2	230.412	225.647	249.469	180.518	49.8939	35	429.878	607.914
0.3	232.787	225.683	249.361	157.978	74.8084	30	370.449	524.400
0.4	235.151	225.709	249.315	135.425	99.7259	30	370.141	525.087
0.5	237.505	225.767	249.243	112.883	124.621	30	374.129	527.853
0.6	239.848	225.826	249.196	90.3305	149.517	31	381.259	543.259
0.7	242.179	225.889	249.161	67.7667	174.413	30	370.096	526.946
0.8	244.501	225.969	249.133	45.1939	199.307	30	371.143	526.118
0.9	246.811	226.083	249.114	22.6083	224.202	30	377.545	532.258
1.0	249.11	226.153	249.11	0	249.11	31	385.114	546.290

Table A.34: Results of ASD-SEP for LandS

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	225.629	225.629	249.65	225.629	0	31	438.234	599.501
0.1	228.026	225.637	249.531	203.073	24.9531	31	377.497	538.939
0.2	230.412	225.647	249.469	180.518	49.8939	31	379.420	539.439
0.3	232.787	225.683	249.361	157.978	74.8084	31	380.545	546.517
0.4	235.151	225.709	249.315	135.425	99.7259	29	358.105	511.362
0.5	237.505	225.767	249.243	112.883	124.621	28	348.118	491.862
0.6	239.848	225.826	249.196	90.3305	149.517	29	352.176	505.705
0.7	242.179	225.889	249.161	67.7667	174.413	30	443.064	602.359
0.8	244.501	225.97	249.133	45.1939	199.307	29	356.872	508.378
0.9	246.811	226.083	249.114	22.6083	224.202	32	394.067	559.626
1.0	249.11	226.153	249.11	0	249.11	31	382.839	545.189

Table A.35: Results of QDEV-AGG for LandS

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	225.629	225.629	225.629	0	225.629	0	0	30	505.904	735.583
0.1	230.408	225.648	25.0982	223.059	203.083	5.01964	22.3059	52	692.144	1077.590
0.2	235.144	225.709	24.8356	223.212	180.567	9.93426	44.6423	51	689.475	1070.680
0.3	239.836	225.841	24.5928	223.305	158.088	14.7557	66.9916	50	670.115	1041.570
0.4	244.483	225.986	24.5015	223.225	135.592	19.6012	89.2901	47	636.628	986.960
0.5	249.086	226.214	24.3355	223.286	113.107	24.3355	111.643	51	685.204	1070.440
0.6	253.644	226.408	24.2133	223.375	90.5633	29.056	134.025	52	708.231	1095.160
0.7	258.158	226.728	24.0259	223.576	68.0183	33.6362	156.503	46	626.580	967.803
0.8	262.629	227.045	23.8802	223.765	45.4091	38.2083	179.012	47	640.801	995.256
0.9	267.059	227.358	23.6923	224.086	22.7358	42.6461	201.678	52	704.327	1099.710
1.0	271.45	236.934	23.5785	224.293	0	47.157	224.293	46	606.675	951.897

Table A.36: Results of QDEV-SEP for LandS

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	225.629	225.629	20.6963	232.776	225.629	0	0	31	405.317	578.797
0.1	230.408	225.647	24.9997	223.257	203.083	4.99994	22.3257	29	374.575	538.343
0.2	235.144	225.709	24.8296	223.225	180.567	9.93185	44.645	31	406.962	579.547
0.3	239.836	225.834	24.5247	223.458	158.084	14.7148	67.0374	29	380.286	543.046
0.4	244.483	225.989	24.4394	223.345	135.593	19.5515	89.3379	27	353.758	505.344
0.5	249.086	226.214	24.3396	223.278	113.107	24.3396	111.639	36	468.995	669.110
0.6	253.644	226.408	24.2451	223.311	90.5634	29.0941	133.986	36	472.638	674.578
0.7	258.158	226.731	23.9727	223.681	68.0193	33.5618	156.577	33	427.395	612.812
0.8	262.629	227.031	23.8064	223.916	45.4062	38.0902	179.133	40	519.569	743.656
0.9	267.059	227.345	23.7165	224.039	22.7345	42.6896	201.635	38	497.470	710.266
1.0	271.45	227.762	23.5785	224.293	0	47.157	224.293	46	682.972	938.734

Table A.37: Results of ASD-AGG for 20Tr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	242632	242632	244245	242632	0	2386	1355.61	1739.84
0.1	242793	242633	244236	218370	24423.6	2867	1684.12	2250.39
0.2	242953	242633	244235	194106	48847	1846	1003.99	1244.42
0.3	243114	242633	244235	169843	73270.5	2347	1328.86	1704.55
0.4	243274	242633	244235	145580	97694	1988	1076.69	1349.78
0.5	243434	242633	244235	121317	122118	2004	1083.13	1361.45
0.6	243594	242649	244223	97059.5	146534	2173	1157.63	1493.92
0.7	243749	242683	244206	72804.9	170944	2278	1209.67	1564.70
0.8	243901	242708	244200	48541.6	195360	2072	1079.17	1375.92
0.9	244050	242708	244199	24270.8	219779	2802	1500.99	2041.19
1.0	244197	242754	244197	0	244197	1997	1025.69	1302.03

Table A.38: Results of ASD-AGG for 20Tr_sk1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	242126	242126	243225	242126	0	1535	859.449	1045.75
0.1	242236	242126	243225	217914	24322.5	1622	894.524	1099.84
0.2	242346	242126	243225	193701	48645	1823	1022.1	1276.92
0.3	242456	242126	243225	169488	72967.5	2022	1225.54	1537.30
0.4	242566	242126	243225	145276	97290.1	1894	1066.31	1343.28
0.5	242676	242126	243225	121063	121613	1678	941.691	1160.83
0.6	242786	242126	243225	96850.5	145935	1674	917.526	1133.56
0.7	242895	242126	243225	72637.8	170257	1641	890.863	1102.50
0.8	243005	242126	243225	48425.2	194580	1566	870.852	1061.31
0.9	243115	242126	243225	24212.6	218902	1832	990.651	1252.14
1.0	243225	242126	243225	0	243225	1831	1012.61	1273.14

Table A.39: Results of ASD-SEP for 20Tr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	242632	242632	244245	242632	0	2194	1332.77	7445.67
0.1	242793	242633	244236	218370	24423.6	2567	1677.19	10411.00
0.2	242954	242633	244235	194107	48847	2191	1374.57	7741.59
0.3	243114	242633	244235	169843	73270.5	2035	1208.01	6649.07
0.4	243274	242633	244235	145580	97694	1377	786.412	3307.08
0.5	243434	242633	244235	121317	122118	1963	1131.87	6072.80
0.6	243594	242648	244224	97059.4	146534	1928	1119.17	5952.97
0.7	243749	242683	244206	72804.8	170944	2042	1170.16	6572.59
0.8	243901	242695	244203	48539	195362	2001	1155.67	6205.75
0.9	244050	242708	244199	24270.8	219779	2055	1163.15	6458.45
1.0	244196	242754	244196	0	244196	3690	2737.43	20242.10

Table A.40: Results of ASD-SEP for 20Tr_sk1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	242126	242126	243225	242126	0	1823	1087.61	5758.50
0.1	242236	242126	243225	217914	24322.5	1463	850.198	4032.67
0.2	242346	242126	243225	193701	48645	1454	812.809	3870.15
0.3	242456	242126	243225	169488	72967.5	1417	801.15	3771.42
0.4	242566	242126	243225	145276	97290	1616	965.376	4732.44
0.5	242676	242126	243225	121063	121612	1467	839.003	3909.75
0.6	242785	242126	243225	96850.5	145935	1743	1030.33	5316.89
0.7	242895	242126	243225	72637.8	170257	1381	764.675	3395.08
0.8	243005	242126	243225	48425.3	194580	1673	939.224	4915.52
0.9	243115	242126	243225	24212.6	218902	1567	893.715	5227.13
1.0	243225	242126	243225	0	243225	1853	1070.42	5838.74

Table A.41: Results of QDEV-AGG for 20Tr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	242632	242632	242632	0.0	242632	0.0	0.0	3041	1933.42	2225.68
0.1	242953	242634	1820.43	242180	218370	364.085	24218	2968	1983.81	2267.25
0.2	243270	242633	1770.55	242278	194107	708.218	48455.5	2880	1845.15	2119.77
0.3	243589	242648	1781.6	242223	169853	1068.96	72666.9	2862	1796.67	2074.77
0.4	243897	242687	1700.65	242313	145612	1360.52	96925.3	2199	1304.57	1516.15
0.5	244193	242755	1570.54	242491	121377	1570.54	121245	3107	1852.21	2150.82
0.6	244481	242755	1566.37	242500	97102.1	1879.64	145500	3077	1779.11	2075.91
0.7	244769	242755	1561.03	242510	72826.6	2185.44	169757	3280	1988.67	2300.19
0.8	245057	242755	1551.75	242529	48551	2482.81	194023	3934	2370.87	2749.67
0.9	245344	242754	1565.04	242502	24275.4	2817.07	218251	3803	2282.99	2647.83
1.0	245632	244068	1564.28	242504	0.0	3128.56	242504	3089	1711.44	2011.71

Table A.42: Results of QDEV-AGG for 20Tr_lsk1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	242126	242126	242126	0	242126	0	0	1765	1107.74	1399.97
0.1	242346	242126	1159.46	241999	217914	231.892	24199.9	2158	1360.99	1689.9
0.2	242565	242127	1164.87	241989	193702	465.947	48397.8	1997	1185.17	1492.43
0.3	242784	242127	1162.64	241993	169489	697.584	72597.9	2313	1399.22	1750.45
0.4	243003	242127	1160.78	241997	145276	928.626	96798.6	2943	1802.43	2260.43
0.5	243223	242127	1160.77	241997	121063	1160.77	120998	3041	1808.59	2295.49
0.6	243442	242127	1161.25	241996	96850.8	1393.5	145198	3637	2140.9	2694.28
0.7	243660	242126	1158.35	242001	72637.9	1621.69	169401	3517	2174.68	2714.9
0.8	243880	242127	1160.17	241998	48425.3	1856.27	193598	3526	2176.51	2720.51
0.9	244099	242127	1158.93	242001	24212.7	2086.08	217801	3532	2263.83	2806.35
1.0	244318	243160	1158.72	242001	0	2317.44	242001	2145	1281.69	1611.7

Table A.43: Results of QDEV-SEP for 20Tr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	242632	242632	242632	0	242632	0	0	1738	1010.23	1033.09
0.1	242952	242633	1795.82	242227	218370	359.163	24222.7	1875	1110.63	1133.84
0.2	243270	242633	1794.25	242229	194107	717.699	48445.8	2350	1463.98	1492.43
0.3	243588	242647	1788.66	242209	169853	1073.19	72662.7	2610	1703.42	1735.64
0.4	243897	242684	1706.95	242301	145610	1365.56	96920.6	1764	979.70	1000.95
0.5	244193	242754	1568.79	242494	121377	1568.79	121247	1971	1114.26	1138.33
0.6	244480	242754	1562.87	242506	97101.6	1875.45	145503	1888	1061.89	1086.4
0.7	244768	242754	1565.96	242499	72826.2	2192.34	169750	1855	992.765	1015.75
0.8	245056	242754	1560.75	242510	48550.8	2497.2	194008	2100	1134.77	1162.64
0.9	245344	242754	1559.71	242512	24275.4	2807.48	218261	2423	1337.03	1369.1
1.0	245631	242754	1560.12	242511	0	3120.23	242511	4609	2815.17	2872.5

Table A.44: Results of QDEV-SEP for 20Tr_lsk1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	242126	242126	242126	0	242126	0	0	1512	854.828	872.729
0.1	242345	242126	1167.01	241983	217913	233.401	24198.3	1506	888.294	905.883
0.2	242564	242126	1167.34	241983	193701	466.936	48396.6	1560	916.584	935.583
0.3	242784	242126	1167.35	241983	169488	700.407	72594.8	1517	875.493	894.382
0.4	243003	242126	1167.33	241983	145276	933.862	96793.1	1936	1171.870	1195.48
0.5	243222	242126	1167.34	241983	121063	1167.34	120991	1843	1077.020	1100.35
0.6	243441	242126	1167.34	241983	96850.4	1400.81	145190	1755	1065.170	1087.57
0.7	243660	242126	1167	241984	72637.8	1633.79	169388	1855	1099.760	1122.22
0.8	243879	242126	1167.34	241983	48425.2	1867.75	193586	2002	1201.660	1226.92
0.9	244098	242126	1167.32	241983	24212.6	2101.18	217785	2136	1342.210	1368.59
1.0	244317	242126	1167.23	241983	0	2334.46	241983	3908	2491.150	2538.97

Table A.45: Results of ASD-AGG for 20Tr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	243346	243346	244930	243346	0	2084	1939.88	2269.14
0.1	243505	243346	244930	219012	24493	1665	1569.33	1786.56
0.2	243663	243346	244930	194677	48986.1	1861	1770.81	2034.78
0.3	243822	243346	244930	170342	73479.1	1716	1629.01	1853.05
0.4	243980	243346	244930	146008	97972.1	2572	2498.1	2984.70
0.5	244138	243346	244930	121673	122465	1476	1351.49	1523.89
0.6	244297	243347	244931	97338.6	146958	1932	1863.79	2150.27
0.7	244455	243346	244930	73003.9	171451	1934	1776.2	2057.47
0.8	244614	243346	244930	48669.3	195944	1549	1370.13	1558.58
0.9	244772	243346	244930	24334.6	220437	2019	1882.35	2188.95
1.0	244930	243346	244930	0	244930	2225	1972.93	2333.53

Table A.46: Results of ASD-AGG for 20Tr_msk1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	239370	239370	241289	239370	0	1684	1740.3	1981.33
0.1	239562	239370	241289	215433	24128.9	1927	2168.9	2473.72
0.2	239754	239370	241289	191496	48257.8	2036	2311.98	2651.21
0.3	239942	239428	241141	167600	72342.4	2043	2225.32	2570.91
0.4	240108	239469	241066	143681	96426.4	2467	2861.05	3348.28
0.5	240266	239483	241048	119742	120524	2217	2390.97	2788.58
0.6	240421	239494	241039	95797.7	144623	2307	2488.22	2914.09
0.7	240575	239496	241038	71848.9	168726	1912	2037.14	2337.16
0.8	240729	239496	241038	47899.2	192830	2184	2360.98	2747.80
0.9	240883	239500	241037	23950	216933	2151	2290.23	2664.53
1	241037	239501	241037	0	241037	2260	2542.71	2951.22

Table A.47: Results of ASD-AGG for 20Tr_msk2

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	243568	243568	245354	243568	0	1368	1251.36	1415.59
0.1	243746	243568	245354	219211	24535.4	1562	1471.94	1680.48
0.2	243925	243568	245354	194854	49070.8	1989	1952.05	2281.39
0.3	244103	243567	245354	170497	73606.2	1695	1644.57	1886.05
0.4	244282	243568	245354	146141	98141.7	1942	1889.76	2208.18
0.5	244461	243567	245354	121784	122677	2228	2295.98	2698.00
0.6	244640	243568	245354	97427.1	147213	1895	1945.64	2242.08
0.7	244818	243568	245354	73070.3	171748	1910	1907.05	2204.25
0.8	244997	243568	245354	48713.5	196283	2044	2132.13	2473.63
0.9	245172	243658	245340	24365.8	220806	1937	2007.58	2314.14
1	245337	243723	245337	0	245337	1951	1983.84	2300.13

Table A.48: Results of ASD-SEP for 20Tr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	243346	243346	244930	243346	0	1989	1999.04	6846.66
0.1	243505	243347	244931	219012	24493.1	1731	1710.03	5595.60
0.2	243663	243346	244930	194677	48986.1	1688	1631.54	5363.56
0.3	243822	243346	244930	170342	73479.1	1560	1453.13	4645.73
0.4	243980	243347	244931	146008	97972.2	1543	1464.60	4470.68
0.5	244138	243346	244930	121673	122465	1818	1796.56	6099.36
0.6	244297	243347	244931	97338.6	146958	1719	1659.31	5573.13
0.7	244455	243346	244930	73003.9	171451	1665	1566.64	5033.08
0.8	244614	243346	244930	48669.3	195944	1713	1608.30	5391.24
0.9	244772	243346	244930	24334.6	220437	2227	2181.21	8402.16
1	244931	243347	244931	0	244931	2195	2068.15	8102.19

Table A.49: Results of ASD-SEP for 20Tr_msk1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	239370	239370	241289	239370	0	1962	2265.95	7562.76
0.1	239562	239370	241289	215433	24128.9	1662	1836.37	5672.99
0.2	239754	239370	241289	191496	48257.8	2291	2860.88	9995.89
0.3	239942	239428	241141	167600	72342.4	1839	2137.46	6950.58
0.4	240108	239470	241064	143682	96425.6	1784	1984.83	6552.51
0.5	240266	239482	241049	119741	120525	1696	1830.18	5969.88
0.6	240421	239494	241039	95797.6	144623	1681	1844.13	5882.74
0.7	240575	239496	241038	71848.8	168726	1462	1540.5	4511.22
0.8	240729	239496	241038	47899.3	192830	1540	1654.57	4976.73
0.9	240883	239500	241037	23950	216933	1616	1667.41	5275.32
1	241037	239502	241037	0	241037	2082	2331.81	8192.13

Table A.50: Results of ASD-SEP for 20Tr_msk2

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	243568	243568	245354	243568	0	1845	1922.14	6581.87
0.1	243746	243568	245354	219211	24535.4	1844	1940.91	6676.80
0.2	243925	243568	245354	194854	49070.8	1631	1640.48	5535.23
0.3	244103	243568	245354	170497	73606.2	1432	1444.51	5000.11
0.4	244282	243568	245354	146141	98141.6	1333	1248.87	3704.62
0.5	244461	243568	245354	121784	122677	1360	1357.17	4034.15
0.6	244639	243567	245354	97427	147212	2024	2261.24	7913.56
0.7	244818	243568	245354	73070.3	171748	1787	1920.57	6346.21
0.8	244997	243568	245354	48713.5	196283	1966	2119.69	7288.27
0.9	245172	243659	245340	24365.9	220806	1547	1589.54	4822.78
1	245337	243723	245337	0	245337	1677	1677.78	5636.58

Table A.51: Results of QDEV-AGG for 20Tr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	243346	243346	243346	0	243346	0	0	1719	1890.11	2218.67
0.1	243664	243347	1627.83	243258	219012	325.567	24325.8	2242	2387.64	2816.89
0.2	243981	243347	1623.29	243267	194678	649.314	48653.4	2163	2374.27	2785.85
0.3	244297	243347	1660.25	243193	170343	996.152	72957.9	3026	3121.27	3700.61
0.4	244613	243347	1655.75	243201	146008	1324.6	97280.5	2625	2639.01	3137.77
0.5	244930	243347	1653.2	243207	121674	1653.2	121603	3301	3346.80	3976.41
0.6	245246	243347	1650.54	243212	97338.8	1980.65	145927	3611	3463.80	4147.61
0.7	245563	243347	1659.56	243194	73004.2	2323.39	170236	3859	3916.87	4654.3
0.8	245880	243347	1643.4	243226	48669.4	2629.44	194581	3749	3876.76	4586.75
0.9	246196	243347	1639.9	243233	24334.7	2951.82	218910	4130	4286.54	5071.14
1	246513	244861	1652.04	243209	0	3304.09	243209	3072	3302.89	3886.09

Table A.52: Results of QDEV-AGG for 20Tr_msk1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	239371	239371	239371	0	239371	0	0	1588	1773.27	2075.17
0.1	239745	239371	2349.45	238409	215434	469.89	23840.9	2284	2762.57	3197.42
0.2	240104	239450	1893.71	238935	191560	757.485	47787	2292	2707.04	3142.11
0.3	240418	239491	1670.9	239239	167644	1002.54	71771.7	2270	2485.3	2918.64
0.4	240725	239501	1644.37	239273	143701	1315.5	95709.3	2596	2788.93	3282.05
0.5	241029	239508	1823.21	238904	119754	1823.21	119452	3313	3735.19	4369.19
0.6	241332	239510	1826.8	238894	95803.9	2192.16	143336	3985	4389.39	5151.22
0.7	241636	239510	1821.78	238903	71852.9	2550.5	167232	4108	4139.36	4927.31
0.8	241939	239559	1759.96	239015	47911.7	2815.93	191212	3524	4048.32	4716.59
0.9	242233	239602	1709.15	239107	23960.2	3076.47	215197	3496	4288.05	4956.31
1.0	242517	240904	1613.34	239290	0	3226.68	239290	3096	3706.81	4294.48

Table A.53: Results of QDEV-AGG for 20Tr_msk2

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	243569	243569	243569	0	243569	0	0	1326	1491.02	1739.48
0.1	243920	243569	2123.51	242831	219212	424.702	24283.1	1834	2318.01	2672.22
0.2	244270	243569	1994.4	243086	194855	797.759	48617.3	2055	2624.86	3023.13
0.3	244621	243569	2036.47	243002	170498	1221.88	72900.6	2080	2367.77	2765.39
0.4	244971	243569	2024.81	243024	146141	1619.85	97209.7	2145	2573.92	2986.97
0.5	245321	243583	2030.29	242999	121791	2030.29	121499	2940	3268.72	3829.97
0.6	245648	243726	1785.36	243358	97490.4	2142.43	146015	3248	3933.73	4552.25
0.7	245966	243783	1694.23	243513	73134.8	2371.92	170459	3740	4193.97	4906.83
0.8	246276	243978	1451.83	243947	48795.5	2322.92	195157	2636	3233.92	3735.97
0.9	246548	244064	1329.35	244165	24406.4	2392.83	219749	3030	3540.22	4117.22
1	246815	245613	1201.73	244411	0	2403.46	244411	2669	3081.48	3594.75

Table A.54: Results of QDEV-SEP for 20Tr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	243346	243346	243346	0	243346	0	0	1950	1993.86	2073.45
0.1	243663	243346	1659.02	243194	219012	331.805	24319.4	1852	1918.97	1992.88
0.2	243979	243346	1659.78	243193	194677	663.911	48638.5	1767	1720.45	1790.28
0.3	244296	243346	1661.11	243190	170342	996.665	72957	1604	1537.95	1602.34
0.4	244613	243346	1653.75	243205	146008	1323	97281.9	1858	1848.37	1924.02
0.5	244929	243346	1659.73	243193	121673	1659.73	121596	1932	1900.85	1976.8
0.6	245246	243347	1659.95	243192	97338.6	1991.94	145915	1906	1866.08	1941.39
0.7	245563	243347	1658.11	243196	73004	2321.36	170237	2056	1976.20	2060.59
0.8	245879	243346	1658.96	243194	48669.3	2654.33	194555	2287	2337.97	2429.47
0.9	246196	243346	1658.75	243195	24334.6	2985.74	218875	2504	2477.86	2588.42
1.0	246512	243346	1661.63	243189	0	3323.27	243189	4335	4490.39	4664.34

Table A.55: Results of QDEV-SEP for 20Tr_msk1

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	239370	239370	239370	0	239370	0	0	2231	2679.67	2771.38
0.1	239744	239370	2335.36	238437	215433	467.072	23843.7	2062	2384.13	2467.11
0.2	240103	239447	1898.62	238930	191557	759.447	47786	1781	1991.96	2061.75
0.3	240417	239488	1682.36	239219	167642	1009.42	71765.8	1830	2060.08	2135.5
0.4	240723	239505	1751.42	239049	143703	1401.13	95619.5	1932	2158.33	2238.39
0.5	241028	239507	1830.35	238887	119754	1830.35	119443	1882	2128.84	2205.56
0.6	241331	239509	1823.8	238899	95803.6	2188.56	143339	1704	1816.87	1886.55
0.7	241635	239509	1823.79	238899	71852.7	2553.31	167229	2164	2467.44	2561.63
0.8	241938	239553	1767.61	238999	47910.6	2828.17	191199	2240	2572.72	2666.34
0.9	242232	239598	1712.89	239099	23959.8	3083.2	215189	2491	2867.11	2982.35
1.0	242516	239694	1607.59	239300	0	3215.18	239300	3731	4085.63	4243.69

Table A.56: Results of QDEV-SEP for 20Tr_msk2

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	243568	243568	243568	0	243568	0	0	1611	1612.54	1677.28
0.1	243918	243567	2058.27	242957	219211	411.653	24295.7	1970	2054.21	2132.44
0.2	244269	243568	2057.8	242958	194854	823.122	48591.6	2158	2410.27	2497.25
0.3	244620	243568	2057.19	242959	170497	1234.31	72887.8	1894	2000.33	2077.45
0.4	244970	243568	2058.51	242957	146141	1646.8	97182.7	2182	2404.79	2495.38
0.5	245320	243583	2028.59	243001	121791	2028.59	121500	1534	1588.14	1649.05
0.6	245646	243723	1735.14	243457	97489.2	2082.17	146074	1974	2083.38	2165.44
0.7	245964	243782	1699.56	243501	73134.5	2379.38	170451	2092	2318.91	2405.05
0.8	246263	243944	1477.32	243888	48788.9	2363.71	195111	2293	2494.21	2587.02
0.9	246547	244064	1330.25	244162	24406.4	2394.45	219746	2588	2930.68	3036.2
1.0	246814	244201	1211.83	244390	0	2423.65	244390	4045	4539.78	4707.34

Table A.57: Results of ASD-AGG for 20Tr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	243791	243791	245743	243791	0	1811	3471.49	3763.95
0.1	243981	243791	245692	219412	24569.2	1633	2986.44	3228.89
0.2	244159	243817	245526	195054	49105.3	1909	3443.62	3766.67
0.3	244330	243817	245526	170672	73657.9	1617	2907.14	3151.19
0.4	244501	243818	245526	146291	98210.6	1803	3185.12	3475.44
0.5	244672	243837	245506	121919	122753	1726	3061.95	3333.30
0.6	244839	243838	245506	97535	147304	1655	2919.14	3168.97
0.7	245005	243837	245506	73151.2	171854	1794	2964.13	3249.61
0.8	245172	243837	245506	48767.5	196405	1671	2813.89	3067.36
0.9	245339	243837	245506	24383.7	220955	1931	3233.77	3562.59
1	245506	243838	245506	0	245506	1777	2984.27	3266.25

Table A.58: Results of ASD-SEP for 20Tr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	243791	243791	245743	243791	0	1631	3097.48	6622.62
0.1	243982	243791	245692	219412	24569.2	1462	2656.35	5478.35
0.2	244159	243817	245526	195054	49105.3	1421	2569.38	5378.66
0.3	244330	243818	245526	170672	73657.9	1316	2323.45	4617.28
0.4	244501	243817	245526	146290	98210.5	1704	2965.28	6731.01
0.5	244672	243838	245506	121919	122753	1474	2541.26	5729.83
0.6	244838	243837	245506	97535	147303	1295	2303.66	4546.29
0.7	245005	243837	245506	73151.2	171854	1568	2773.07	5976.97
0.8	245172	243838	245506	48767.5	196405	1329	2252.48	4469.11
0.9	245339	243837	245506	24383.7	220955	1550	2623.16	5648.74
1	245506	243837	245506	0	245506	1810	2836.31	6861.05

Table A.59: Results of QDEV-AGG for 20Tr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	243791	243791	243791	0	243791	0	0	1697	3561.91	4327.65
0.1	244159	243809	1934.26	243439	219428	386.852	24343.9	2206	4860.52	5863.35
0.2	244500	243819	1872.03	243481	195055	748.81	48696.2	2193	4472.50	5465.08
0.3	244838	243839	1818.62	243531	170687	1091.17	73059.4	1980	3769.98	4658.60
0.4	245170	243838	1769.08	243630	146303	1415.26	97452	2492	4613.97	5728.6
0.5	245503	243839	1773.46	243621	121919	1773.46	121811	3228	5738.01	6974.22
0.6	245836	243839	1819.69	243528	97535.7	2183.63	146117	2944	5319.25	6451.67
0.7	246167	243960	1642.28	243828	73188	2299.19	170680	3498	6384.62	7710.92
0.8	246480	244003	1588.37	243922	48800.6	2541.38	195138	3461	6594.76	7910.01
0.9	246789	244003	1581.26	243937	24400.3	2846.27	219543	3198	5968.15	7381.50
1	247099	245520	1578.36	243942	0	3156.73	243942	2199	4451.84	5432.59

Table A.60: Results of QDEV-SEP for 20Tr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	243791	243791	243791	0	243791	0	0	1513	2704.12	2800.99
0.1	244158	243809	1957.67	243386	219428	391.534	24338.6	1444	2651.70	2741.89
0.2	244499	243818	1879.53	243465	195054	751.81	48692.9	1726	3181.05	3291.89
0.3	244836	243837	1808.39	243550	170686	1085.03	73065	1316	2165.22	2249.31
0.4	245169	243837	1803.84	243559	146302	1443.08	97423.6	1618	2885.67	2990.83
0.5	245502	243837	1797.49	243572	121919	1797.49	121786	1678	3061.73	3169.64
0.6	245835	243838	1808.5	243550	97535.1	2170.2	146130	2164	3941.42	4079.30
0.7	246166	243959	1624.55	243862	73187.8	2274.37	170704	2555	4731.92	4901.53
0.8	246478	244002	1579.61	243938	48800.4	2527.37	195151	2361	4059.50	4213.42
0.9	246788	244002	1580.98	243936	24400.2	2845.76	219542	2591	4346.30	4510.26
1.0	247098	244002	1583.32	243931	0	3166.64	243931	3810	6766.92	7057.00

Table A.61: Results of ASD-AGG for ssnTr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.00	0.00	0.00	0.00	0.00	0.00	298	76.00	308.32
0.10	0.00	0.00	0.00	0.00	0.00	436	105.64	566.32
0.20	0.00	0.00	0.00	0.00	0.00	498	127.48	694.07
0.30	0.00	0.00	0.00	0.00	0.00	197	50.41	169.49
0.40	0.00	0.00	0.00	0.00	0.00	188	48.21	154.73
0.50	0.00	0.00	0.00	0.00	0.00	231	59.57	213.80
0.60	0.00	0.00	0.00	0.00	0.00	279	73.82	284.33
0.70	0.00	0.00	0.00	0.00	0.00	320	82.41	356.77
0.80	0.00	0.00	0.00	0.00	0.00	667	174.35	1159.15
0.90	0.00	0.00	0.00	0.00	0.00	285	74.30	306.55
1.00	0.00	0.00	0.00	0.00	0.00	483	123.38	625.62

Table A.62: Results of ASD-SEP for ssnTr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.00	0.00	0.00	0.00	0.00	0.00	183	49.08	87.46
0.10	0.00	0.00	0.00	0.00	0.00	289	79.29	153.99
0.20	0.00	0.00	0.00	0.00	0.00	364	94.77	200.49
0.30	0.00	0.00	0.00	0.00	0.00	178	47.94	84.53
0.40	0.00	0.00	0.00	0.00	0.00	211	55.87	99.96
0.50	0.00	0.00	0.00	0.00	0.00	665	182.27	441.05
0.60	0.00	0.00	0.00	0.00	0.00	218	57.60	105.41
0.70	0.00	0.00	0.00	0.00	0.00	348	98.30	199.78
0.80	0.00	0.00	0.00	0.00	0.00	203	58.00	101.15
0.90	0.00	0.00	0.00	0.00	0.00	499	133.80	299.54
1.00	0.00	0.00	0.00	0.00	0.00	330	94.02	181.10

Table A.63: Results of QDEV-AGG for ssnTr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	324	87.81	114.48
0.1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	853	261.93	335.28
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1629	439.85	573.94
0.3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1217	334.76	431.23
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	917	242.61	315.55
0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	708	193.47	252.78
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1883	524.04	675.41
0.7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1068	300.87	387.69
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	686	194.69	249.16
0.9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1571	433.35	559.89
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	597	165.48	213.53

Table A.64: Results of QDEV-SEP for ssnTr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	0.000	0.00	1.406	-1.406	0.00	0.00	0.00	2131	616.232	655.602
0.1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	769	210.425	224.175
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	802	198.489	213.049
0.3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	814	226.066	241.034
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	267	69.521	73.938
0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	816	217.971	232.940
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1177	318.544	339.004
0.7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1022	275.781	294.004
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	864	205.374	221.44
0.9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	386	95.947	103.345
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1680	489.119	519.897

Table A.65: Results of ASD-AGG for ssnTr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.00	0.00	0.00	0.00	0.00	0.00	710	1151.71	2271.02
0.10	0.00	0.00	0.00	0.00	0.00	591	953.15	1748.72
0.20	0.00	0.00	0.00	0.00	0.00	209	342.57	491.39
0.30	0.00	0.00	0.00	0.00	0.00	519	844.79	1493.23
0.40	0.00	0.00	0.00	0.00	0.00	299	495.70	756.63
0.50	0.00	0.00	0.00	0.00	0.00	273	442.62	662.82
0.60	0.00	0.00	0.00	0.00	0.00	4862	8393.06	53992.40
0.70	0.00	0.00	0.00	0.00	0.00	309	505.17	777.45
0.80	0.00	0.00	0.00	0.00	0.00	604	985.13	1836.52
0.90	0.00	0.00	0.00	0.00	0.00	1521	2440.42	6894.98
1.00	0.00	0.00	0.00	0.00	0.00	921	1563.58	3356.59

Table A.66: Results of ASD-SEP for ssnTr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.00	0.00	0.00	0.00	0.00	0.00	1301	2168.11	3296.56
0.10	0.00	0.00	0.00	0.00	0.00	3724	6366.14	13640.70
0.20	0.00	0.00	0.00	0.00	0.00	802	1337.99	1863.74
0.30	0.00	0.00	0.00	0.00	0.00	469	769.42	1003.35
0.40	0.00	0.00	0.00	0.00	0.00	313	517.61	643.72
0.50	0.00	0.00	0.00	0.00	0.00	434	699.29	897.25
0.60	0.00	0.00	0.00	0.00	0.00	846	1399.77	1949.32
0.70	0.00	0.00	0.00	0.00	0.00	335	558.27	698.28
0.80	0.00	0.00	0.00	0.00	0.00	392	650.24	824.66
0.90	0.00	0.00	0.00	0.00	0.00	869	1431.20	2031.64
1.00	0.00	0.00	0.00	0.00	0.00	313	513.73	639.08

Table A.67: Results of QDEV-AGG for ssnTr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	0.00	0.00	0.00	0.00	15.83	0.00	0.00	270	444.74	577.83
0.1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2646	4355.95	5647.23
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1160	2046.77	2615.38
0.3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1449	2328.13	3039.81
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2281	3833.48	4955.20
0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2462	4005.35	5224.06
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2776	4780.38	6141.84
0.7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1171	1958.41	2536.00
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1740	3045.06	3910.59
0.9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1159	1984.16	2549.77
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1522	2600.60	3347.44

Table A.68: Results of QDEV-SEP for ssnTr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	0.0000	0.00	0.0694	-0.0694	0.00	0.00	0.00	2151	3425.34	3692.00
0.1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2178	3521.95	3788.19
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	734	1181.32	1271.3
0.3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1626	2590.09	2789.24
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1567	2614.48	2807.34
0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	825	1298.54	1399.13
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1241	2078.86	2231.56
0.7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1060	1737.7	1866.96
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1084	1762.11	1894.71
0.9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2201	3829.39	4099.11
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2191	3615.35	3885.35

Table A.69: Results of ASD-AGG for ssnTr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.00	0.00	0.00	0.00	0.00	0.00	829	7276.66	9437.93
0.10	0.00	0.00	0.00	0.00	0.00	817	7062.98	9159.55
0.20	0.00	0.00	0.00	0.00	0.00	441	3882.58	4707.08
0.30	0.00	0.00	0.00	0.00	0.00	835	7344.59	9505.34
0.40	0.00	0.00	0.00	0.00	0.00	614	5407.70	6868.06
0.50	0.00	0.00	0.00	0.00	0.00	945	8251.08	11067.60
0.60	0.00	0.00	0.00	0.00	0.00	726	6423.75	8175.85
0.70	0.00	0.00	0.00	0.00	0.00	5100	43633.20	95405.80
0.80	0.00	0.00	0.00	0.00	0.00	461	4070.21	4942.06
0.90	0.00	0.00	0.00	0.00	0.00	710	6368.76	8014.25
1.00	0.00	0.00	0.00	0.00	0.00	1506	13206.60	19068.70

Table A.70: Results of ASD-SEP for ssnTr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.00	0.00	0.00	0.00	0.00	0.00	964	8419.03	9820.86
0.10	0.00	0.00	0.00	0.00	0.00	612	5453.01	6417.97
0.20	0.00	0.00	0.00	0.00	0.00	788	7026.87	8321.54
0.30	0.00	0.00	0.00	0.00	0.00	1084	9522.66	11161.60
0.40	0.00	0.00	0.00	0.00	0.00	888	7713.51	8957.97
0.50	0.00	0.00	0.00	0.00	0.00	657	5802.88	6649.75
0.60	0.00	0.00	0.00	0.00	0.00	1558	13207.00	15806.40
0.70	0.00	0.00	0.00	0.00	0.00	1159	10173.50	11908.50
0.80	0.00	0.00	0.00	0.00	0.00	996	8811.89	10233.20
0.90	0.00	0.00	0.00	0.00	0.00	506	4420.77	5040.54
1.00	0.00	0.00	0.00	0.00	0.00	1937	17090.60	20735.60

Table A.71: Results of QDEV-AGG for ssnTr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	632	5872.36	7998.0
0.1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	838	7586.46	10412.5
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1703	16078.00	21817.8
0.3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1908	18164.30	24588.5
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2427	21111.80	29394.8
0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1403	13239.00	17975.0
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11300	98817.80	137021.0
0.7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1281	11777.00	16102.6
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1529	14356.50	19525.9
0.9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5790	50064.00	69521.4
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1646	17137.30	23498.2

Table A.72: Results of QDEV-SEP for ssnTr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1076	9509.35	10424.6
0.1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1622	14008.5	15416.4
0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1801	15769	17340.7
0.3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1453	12514.5	13756.1
0.4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1319	11318	12448.8
0.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1041	8964.81	9843.72
0.6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1587	13598.8	14954.8
0.7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2583	22506.1	24694.9
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2294	19025.9	20960.1
0.9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1956	16242.4	17894.9
1.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1713	14891.4	16353.3

Table A.73: Results of ASD-AGG for stormTr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	13014000	13014000	13120600	13014000	0	16	2.329	3.031
0.1	13024700	13014000	13120600	11712600	1312060	16	2.440	3.016
0.2	13035300	13014000	13120600	10411200	2624120	16	2.271	3.016
0.3	13046000	13014000	13120600	9109830	3936170	16	2.340	3.016
0.4	13056700	13014000	13120600	7808420	5248230	16	2.439	3.031
0.5	13067300	13014000	13120600	6507020	6560290	16	2.266	3.016
0.6	13078000	13014000	13120600	5205620	7872350	16	2.514	3.031
0.7	13088600	13014000	13120600	3904210	9184410	16	2.466	3.031
0.8	13099300	13014000	13120600	2602810	10496500	16	2.327	3.031
0.9	13109900	13014000	13120600	1301400	11808500	16	2.326	3.031
1	13120600	13014000	13120600	0	13120600	16	2.268	3.016

Table A.74: Results of ASD-SEP for stormTr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0.0	13014000	13014000	13120600	13014000	0	16	2.443	3.297
0.1	13024700	13014000	13120600	11712600	1312060	16	2.425	3.281
0.2	13035300	13014000	13120600	10411200	2624120	16	2.341	3.297
0.3	13046000	13014000	13120600	9109830	3936170	16	2.405	3.281
0.4	13056700	13014000	13120600	7808420	5248230	16	2.478	3.296
0.5	13067300	13014000	13120600	6507020	6560290	16	2.563	3.282
0.6	13078000	13014000	13120600	5205620	7872350	16	2.333	3.313
0.7	13088600	13014000	13120600	3904210	9184410	16	2.278	3.344
0.8	13099300	13014000	13120600	2602810	10496500	16	2.532	3.328
0.9	13109900	13014000	13120600	1301400	11808500	16	2.454	3.312
1.0	13120600	13014000	13120600	0	13120600	16	2.371	3.282

Table A.75: Results of QDEV-AGG for stormTr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	13014000	13014000	13014000	0	13014000	0	0	16	2.373	12.890
0.1	13035400	13014000	103247	13020700	11712600	20649	1302070	37	7.933	30.625
0.2	13056700	13014000	105722	13015700	10411200	42289	2603140	37	8.370	30.953
0.3	13078000	13014000	107695	13011700	9109830	64617	3903520	37	8.405	30.860
0.4	13099300	13014000	105760	13015600	7808430	84608	5206240	44	9.904	36.485
0.5	13120600	13014100	108638	13009900	6507030	108638	6504940	41	8.875	33.860
0.6	13141900	13014100	106198	13014700	5205620	127438	7808840	41	9.261	34.172
0.7	13163200	13014000	106795	13013500	3904210	149513	9109470	44	9.559	36.516
0.8	13184500	13014000	106465	13014200	2602810	170344	10411400	45	9.741	37.282
0.9	13205800	13014000	106156	13014800	1301400	191081	11713300	34	7.540	28.407
1	13227100	13120300	106797	13013500	0	213594	13013500	42	8.570	33.688

Table A.76: Results of QDEV-SEP for stormTr_l

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	13014000	13014000	13014000	0	13014000	0	0	16	2.488	3.110
0.1	13035300	13014000	106540	13014000	11712600	21308.1	1301400	17	2.638	3.109
0.2	13056700	13014000	105415	13016300	10411200	42165.8	2603260	18	2.770	3.313
0.3	13078000	13014000	106984	13013200	9109830	64190.2	3903950	20	3.141	3.625
0.4	13099300	13014000	105415	13016300	7808430	84331.6	5206520	24	3.721	4.328
0.5	13120600	13014000	106738	13013700	6507020	106738	6506830	27	3.922	4.844
0.6	13141900	13014000	106540	13014000	5205620	127848	7808420	29	4.276	5.14
0.7	13163200	13014000	105759	13015600	3904210	148062	9110930	30	4.267	5.25
0.8	13184500	13014000	106465	13014200	2602810	170344	10411400	32	4.566	5.61
0.9	13205800	13014000	106514	13014100	1301400	191725	11712700	36	5.313	6.328
1.0	13227100	13014100	106797	13013500	0	213594	13013500	42	6.109	7.375

Table A.77: Results of ASD-AGG for stormTr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	13696600	13696600	13813000	13696600	0	13	47.009	57.375
0.1	13708300	13696600	13813000	12327000	1381300	13	47.554	57.188
0.2	13719900	13696600	13813000	10957300	2762600	13	47.797	57.484
0.3	13731500	13696600	13813000	9587650	4143900	13	47.275	57.125
0.4	13743200	13696600	13813000	8217980	5525200	13	48.044	57.531
0.5	13754800	13696600	13813000	6848320	6906500	13	47.851	57.765
0.6	13766500	13696600	13813000	5478660	8287800	13	47.315	57.125
0.7	13778100	13696600	13813000	4108990	9669110	13	47.929	57.391
0.8	13789700	13696600	13813000	2739330	11050400	13	48.207	57.656
0.9	13801400	13696600	13813000	1369660	12431700	13	47.512	57.578
1	13813000	13696600	13813000	0	13813000	13	47.945	57.469

Table A.78: Results of ASD-SEP for stormTr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	13696600	13696600	13813000	13696600	0	13	48.151	58.422
0.1	13708300	13696600	13813000	12327000	1381300	13	47.633	58.125
0.2	13719900	13696600	13813000	10957300	2762600	13	48.703	58.812
0.3	13731500	13696600	13813000	9587650	4143900	13	48.798	58.984
0.4	13743200	13696600	13813000	8217980	5525200	13	48.191	58.125
0.5	13754800	13696600	13813000	6848320	6906500	13	47.374	58.25
0.6	13766500	13696600	13813000	5478660	8287800	13	48.151	58.578
0.7	13778100	13696600	13813000	4108990	9669110	13	48.072	58.422
0.8	13789700	13696600	13813000	2739330	11050400	13	49.013	59.297
0.9	13801400	13696600	13813000	1369660	12431700	13	48.043	57.828
1	13813000	13696600	13813000	0	13813000	13	48.614	57.797

Table A.79: Results of QDEV-AGG for stormTr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	13696600	13696600	13696600	0	13696600	0.0	0	13	54.782	259.751
0.1	13719900	13696600	117050	13695300	12327000	23409.9	1369530	40	220.558	824.662
0.2	13743200	13696600	115669	13698000	10957300	46267.7	2739610	45	248.882	924.334
0.3	13766500	13696600	117746	13693900	9587650	70647.4	4108170	33	180.356	680.098
0.4	13789700	13696600	116404	13696600	8217990	93123.2	5478630	42	230.134	863.364
0.5	13813000	13696600	116079	13697200	6848320	116079.0	6848610	46	249.414	941.35
0.6	13836300	13696600	116616	13696200	5478660	139939.0	8217690	44	237.723	901.521
0.7	13859600	13696600	117527	13694300	4108990	164537.0	9586030	42	227.544	860.193
0.8	13882800	13696600	117061	13695300	2739330	187297.0	10956200	46	251.611	947.647
0.9	13906100	13696600	117270	13694800	1369660	211086.0	12325400	41	228.906	851.928
1	13929400	13813500	115844	13697700	0	231687.0	13697700	43	223.475	881.506

Table A.80: Results of QDEV-SEP for stormTr_m

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	13696600	13696600	13696600	0	13696600	0	0	13	48.093	57.798
0.1	13719900	13696600	117936	13693500	12327000	23587.2	1369350	13	47.344	57.235
0.2	13743200	13696600	118016	13693400	10957300	47206.3	2738670	16	60.102	71.469
0.3	13766500	13696600	117795	13693800	9587650	70677.3	4108140	20	72.943	87.173
0.4	13789700	13696700	116816	13695800	8217990	93452.6	5478300	19	68.518	82.314
0.5	13813000	13696600	116688	13696000	6848320	116688	6848000	23	83.132	99.783
0.6	13836300	13696600	115902	13697600	5478660	139083	8218540	25	89.332	108.767
0.7	13859600	13696600	116524	13696300	4108990	163133	9587430	31	112.563	134.798
0.8	13882800	13696600	116310	13696800	2739330	186095	10957400	31	113.181	135.47
0.9	13906100	13696700	116486	13696400	1369670	209675	12326800	37	131.510	158.424
1.0	13929400	13696600	115844	13697700	0	231687	13697700	43	152.461	183.393

Table A.81: Results of ASD-AGG for stormTr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	13918300	13918300	14039700	13918300	0	13	1194.47	1437.20
0.1	13930400	13918300	14039700	12526400	1403970	13	1193.47	1436.16
0.2	13942600	13918300	14039700	11134600	2807940	13	1182.62	1425.38
0.3	13954700	13918300	14039700	9742790	4211910	13	1190.27	1433.64
0.4	13966800	13918300	14039700	8350970	5615870	13	1194.03	1437.33
0.5	13979000	13918300	14039700	6959140	7019840	13	1196.43	1441.03
0.6	13991100	13918300	14039700	5567310	8423810	13	1190.89	1442.59
0.7	14003300	13918300	14039700	4175480	9827780	13	1193.94	1439.16
0.8	14015400	13918300	14039700	2783660	11231700	13	1204.56	1509.36
0.9	14027500	13918300	14039700	1391830	12635700	13	1193.69	1491.09
1	14039700	13918300	14039700	0	14039700	13	1197.54	1469.31

Table A.82: Results of ASD-SEP for stormTr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	ϕ_{ASD}	$(1-\lambda)*$ $\mathbb{E}[f(\cdot)]$	$\lambda*$ ϕ_{ASD}	Itr	CPU (sec)	Run (sec)
0	13918300	13918300	14039700	13918300	0	13	1189.26	1432.13
0.1	13930400	13918300	14039700	12526400	1403970	13	1189.55	1437.94
0.2	13942600	13918300	14039700	11134600	2807940	13	1189.36	1431.58
0.3	13954700	13918300	14039700	9742790	4211910	13	1181.53	1421.13
0.4	13966800	13918300	14039700	8350970	5615870	13	1188.62	1429.45
0.5	13979000	13918300	14039700	6959140	7019840	13	1194.84	1439.97
0.6	13991100	13918300	14039700	5567310	8423810	13	1204.05	1452.78
0.7	14003300	13918300	14039700	4175480	9827780	13	1194.13	1438.72
0.8	14015400	13918300	14039700	2783660	11231700	13	1211.46	1465.26
0.9	14027500	13918300	14039700	1391830	12635700	13	1214.13	1460.52
1	14039700	13918300	14039700	0	14039700	13	1191.29	1440.16

Table A.83: Results of QDEV-AGG for stormTr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ $*\ \mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ $*\ \mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0	13918300	13918300	13918300	0	13918300	0	0	13	1381.96	6597.06
0.1	13942600	13918300	122089	13916900	12526500	24417.7	1391690	35	4261.31	17674.6
0.2	13966800	13918300	122809	13915500	11134600	49123.5	2783100	42	5072.90	20886
0.3	13991100	13918300	120667	13919800	9742800	72400.4	4175930	37	4442.75	18434.4
0.4	14015400	13918300	121360	13918400	8350970	97088.1	5567350	42	5031.53	20856.5
0.5	14039700	13918300	121259	13918600	6959140	121259.0	6959290	44	5144.81	21706
0.6	14064000	13918300	121339	13918400	5567310	145607.0	8351050	43	5096.44	21246.7
0.7	14088300	13918300	122071	13917000	4175490	170899.0	9741890	42	5033.82	20824.9
0.8	14112500	13918300	121710	13917700	2783660	194736.0	11134100	46	5511.31	22830.6
0.9	14136800	13918300	121598	13917900	1391830	218877.0	12526100	42	5092.28	20894.9
1	14161100	14040100	121051	13919000	0	242102.0	13919000	41	4452.99	19932

Table A.84: Results of QDEV-SEP for stormTr_h

λ	Obj.	$\mathbb{E}[f(\cdot)]$	$\mathbb{E}[\nu(\omega)]$	z	$(1-\lambda\varepsilon_1)$ * $\mathbb{E}[f(\cdot)]$	$\lambda(\varepsilon_1+\varepsilon_2)$ * $\mathbb{E}[\nu(\omega)]$	$\lambda\varepsilon_1 z$	Itr	CPU (sec)	Run (sec)
0.0	13918300	13918300	13918300	0	13918300	0	0	13	1197.62	1445.47
0.1	13942600	13918300	120836	13919400	12526400	24167.1	1391940	14	1270.23	1532.09
0.2	13966800	13918300	120843	13919400	11134600	48337.4	2783880	17	1545.89	1857.82
0.3	13991100	13918300	120656	13919800	9742790	72393.4	4175940	20	1788.71	2153.4
0.4	14015400	13918300	120916	13919300	8350970	96732.9	5567700	18	1617.34	1945.57
0.5	14039700	13918300	122703	13915700	6959140	122703	6957850	24	2162.52	2614.27
0.6	14064000	13918300	122241	13916600	5567320	146690	8349990	26	2358.67	2830.58
0.7	14088300	13918300	121757	13917600	4175480	170459	9742310	26	2314.79	2782.43
0.8	14112500	13918300	122015	13917100	2783660	195223	11133700	30	2650.23	3189.04
0.9	14136800	13918300	120936	13919300	1391830	217684	12527300	39	3432.96	4125.43
1.0	14161100	13918300	121051	13919000	0	242102	13919000	41	3590.21	4305.12

A.2 Parameters for QDEV-AGG and QDEV-SEP Algorithm by Test Instance

This section lists the values to which the parameters are set for each instance. In our implementation of the QDEV-AGG and QDEV-SEP algorithms. The quantile value ψ was set to 0.5 to allow accurate comparison with the algorithms for ASD that have target value of $\mathbb{E}[f(x,\tilde{\omega})]$. In addition the upper and lower bounds of target z, δ_l and δ_h were set to fixed values that are summarized below.

Note: For QDEV-SEP and QDEV-AGG Algorithms at $\lambda = 0$, the target parameter z is set to 0 for the first iteration and allowed to be free for the rest of the iterations. This allowed us to find the risk-neutral objective value ($\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$) or objective value at $\lambda = 0$. For $\lambda = 0.1$ to 1, z is bounded by a lower bound δ_l and a upper bound δ_h that are centered around the risk-neutral objective value for most instances.

QDEV-AGG Algorithm

Quantile deviation target's (z) bounds are listed below, along with ϵ (minimum percentage gap of upper and lower bounds for algorithm termination) if it is set to a different value other then 10^{-6}:

Instances: *cep1*, *cep1a*, *cep1sk*, *pgp2*, *pgp2e*, *pgp2f*, *gbd*, *gbd_sk3*, *LandS*, *stormTr_l*, *stormTr_m*, *stormTr_h*

$\delta_h = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} + 0.5\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$; $\delta_l = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} - 0.5\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$.

Instances: *ssnTr_l*, *ssnTr_m*, *ssnTr_h*

δ_h = 100, δ_l = -20;

Instances: *20Tr_l*, *20Tr_lsk1*

$\delta_h = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} + 0.05\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$; and $\delta_l = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} - 0.05\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$.

Instance: *20Tr_m*

$\delta_h = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} + 0.75\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$; and $\delta_l = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} - 0.75\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$.

Instances: *20Tr_msk1*

$\delta_h = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} + 0.50\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$; and $\delta_l = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} - 0.50\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$.

Instance: *20Tr_msk2* at λ = [0,0.7], [0.9, 1.0]

$\epsilon = 7 \times 10^{-6}$.

$\delta_h = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} + 0.75\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$; and $\delta_l = \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0} - 0.75\ \mathbb{E}[f(x, \tilde{\omega})]_{\lambda=0}$ at [0,0.7].

δ_h = $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ + 0.50 $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$; and δ_l= $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ + 0.50 $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ at [0.9, 1.0].

Instances: *20Tr_msk2* at $\lambda = 0.8$

δ_h = $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ + 0.50 $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$; and δ_l =$\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ − 0.50 $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$.

Instances: *20Tr_h* at λ = [0-0.4], 0.6, 0.7, 0.9

$\epsilon = 7 \times 10^{-6}$

δ_h = $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ + 0.05 $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$; and δ_l = $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$- 0.05 $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$.

Instances: *20Tr_h* at λ = 0.5, 0.8, 1.0

$\epsilon = 7 \times 10^{-6}$

δ_h = $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ + 0.75 $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$; and δ_l = $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ − 0.75 $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$.

QDEV-SEP Algorithm

Quantile deviation target's (z) bounds are listed below, along with ϵ if it is set to a different value other then 10^{-6}:

Instances: *cep1*, *cep1a*, *cep1sk*, *pgp2*, *pgp2e*, *pgp2f*, *gbd*, *gbd_sk3*, *LandS*, *stormTr_l*, *stormTr_m*, *stormTr_h*

δ_h = $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ + 0.5$\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$; and δ_l = $\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$ − 0.5$\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$.

Instances : *ssnTr_l*, *ssnTr_m*, *ssnTr_h*

δ_h= 100, and δ_l = -20.

Instances: *20Tr_l*, *20Tr_sk1*, *20Tr_m*, *20Tr_msk1*, *20Tr_msk2*, *20Tr_h*

$\delta_h = \mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0} + 0.05\,\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$; and $\delta_l = \mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0} - 0.05\,\mathbb{E}[f(x,\tilde{\omega})]_{\lambda=0}$.

Note: Parameter η that approximates quantile deviation in the master program is set to $\eta > 0$ for all 20 instances when using the QDEV-SEP algorithm.