

# Recourse-based Stochastic Nonlinear Programming: Properties and Benders-SQP Algorithms

Ankur A. Kulkarni

Uday V. Shanbhag\*

August 8, 2009

## Abstract

In this paper, we study recourse-based stochastic nonlinear programs and make two sets of contributions. The first set assumes general probability spaces and provides a deeper understanding of feasibility and recourse in stochastic nonlinear programs. A sufficient condition, for equality between the sets of feasible first-stage decisions arising from two different interpretations of almost sure feasibility, is provided. This condition is an extension to nonlinear settings of the “W-condition,” first suggested by Walkup and Wets [65]. Notions of complete and relatively-complete recourse for nonlinear stochastic programs are defined and simple sufficient conditions for these to hold are given. Implications of these results on the L-shaped method are discussed. Our second set of contributions lies in the construction of a scalable, superlinearly convergent method for solving this class of problems, under the setting of a finite sample-space. We present a novel hybrid algorithm that combines sequential quadratic programming (SQP) and Benders decomposition. In this framework, the resulting quadratic programming approximations while arbitrarily large, are observed to be two-period stochastic quadratic programs (QPs) and are solved through two variants of Benders decomposition. The first is based on an inexact-cut L-shaped method for stochastic quadratic programming [55, 57] while the second is a quadratic extension to a trust-region method suggested by Linderoth and Wright in [42]. Obtaining Lagrange multiplier estimates in this framework poses a unique challenge and are shown to be cheaply obtainable through the solution of a single low-dimensional QP. Globalization of the method is achieved through a parallelizable linesearch procedure. Finally, the efficiency and scalability of the algorithm are demonstrated on a set of stochastic nonlinear programming test problems.

## 1 Introduction

This paper concerns stochastic nonlinear programs: optimization problems in which the objective function and constraints are nonlinear and possibly nonconvex functions of the decision variables, parameterized by random variables. The randomness is characterized by a probability space  $(\Upsilon, \mathcal{F}, \mu)$  wherein by  $\omega$  we denote any point in  $\Upsilon$ . We specifically consider recourse based stochastic nonlinear programs such as (SNLP):

$$\begin{array}{ll} \text{SNLP} & \text{minimize} \quad f(x) + Q(x) \\ & \text{subject to} \quad \begin{array}{l} u(x) = 0, \\ x \geq 0, \end{array} \end{array}$$

where  $Q(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $Q(x) \equiv \mathbb{E}_\mu [Q(x; \omega)]$ ,  $\mathbb{E}_\mu[\cdot]$  is the expectation operator on  $(\Upsilon, \mathcal{F}, \mu)$  and  $Q(x; \omega)$  is the optimal value of the problem (RNLP( $x; \omega$ )):

$$\text{RNLP}(x; \omega) \quad \min_y \{h(y; \omega) : a(x, \omega) + d(y, \omega) = 0, y \geq 0\},$$

where  $a : \mathbb{R}^n \times \Upsilon \rightarrow \mathbb{R}^p$  and  $d : \mathbb{R}^N \times \Upsilon \rightarrow \mathbb{R}^p$ . This paper pertains to two important challenges regarding (SNLP): (i) a deeper understanding of feasibility and recourse in nonlinear (and hence analytically nontrivial) settings and (ii) the construction of a scalable convergent algorithm for solving (SNLP) under a finite  $\Upsilon$ . Our key contributions may be summed up as follows:

---

\*Both authors are in the Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL-61801 and are reachable at (akulkar3, udaybag@illinois.edu). This research has been supported by an NSF research grant (CCF-0728863)

**1. Feasibility and recourse:** Throughout we shall be concerned with the “almost sure” notion of feasibility and optimality of (SNLP), as defined below.

**Definition 1.1 (Almost sure feasible and optimal solutions)** *Let  $Y$  be a space of  $\Upsilon \rightarrow \mathbb{R}^N$  continuous functions (random variables). A vector  $x^* \in \mathbb{R}^m$  is called almost sure feasible for (SNLP) if  $u(x^*) = 0, x^* \geq 0$  and there exists  $y^* \in Y$  such that*

$$a(x^*, \omega) + d(y^*(\omega), \omega) = 0, y^*(\omega) \geq 0, \text{ almost surely.}$$

Furthermore, if the set of almost sure-feasible solutions is denoted by  $\mathcal{N}_F$  and there exists a neighborhood  $\mathcal{N}$  of  $x^*$  such that for all  $x \in \mathcal{N} \cap \mathcal{N}_F$ ,

$$f(x^*) + \mathcal{Q}(x^*) \leq f(x) + \mathcal{Q}(x),$$

then  $x^*$  is called an almost sure optimal solution of (SNLP).

- (i) When  $\Upsilon$  is infinite, there are many interpretations of “feasibility” for stochastic programs, as first noted by Walkup and Wets [65, 1967] in the case of stochastic linear programs. We consider the two interpretations feasibility studied by them and derive a simple condition under which they are equivalent. Specifically, if  $\hat{\Upsilon}$  is a closed subset of measure one of  $\Upsilon$ , then under a suitable sufficiency condition the set of first-stage feasible decisions was invariant under the choice of  $\hat{\Upsilon}$ . Importantly, we derive the W-condition for stochastic convex programs with (i) linear and (ii) nonlinear constraints. Logically, this sufficiency condition is an extension of the W-condition from [65] to stochastic nonlinear programs, but it is analytically more difficult to obtain.
- (ii) Furthermore, the notion of complete and relatively complete recourse, while well understood in linear settings, is made rigorous in nonlinear settings.
- (iii) Finally, we highlight some interesting implications of our findings on the workings of L-shaped methods.

Since this body of questions has relevance to L-shaped methods, we thought it fitting to present them in connection with the algorithmic contributions.

**2. A scalable convergent algorithmic scheme:** A novel scalable framework for (SNLP) is designed under the assumption that  $\Upsilon$  is finite. This framework combines the convergence properties of sequential quadratic programming (SQP) with the scalability afforded by Benders decomposition. However, our algorithm, while inspired by SQP methods, has several crucial differences:

- (i) The quadratic programming approximations generated, are solved not by an active-set scheme, as is often the case [46], but by a quadratic generalization of the L-shaped method. We provide two generalizations for convex stochastic QPs: the first uses inexact cuts (due to Shanbhag et al. [55, 57]) while the second employs a trust-region (TR) (extended from TR-based algorithm of Linderoth and Wright [42]). We present extensions of the convergence theory provided in [42] to allow for solving stochastic QPs.
- (ii) The use of Benders-type QP solvers necessitates the use of a *sparse* quasi-Newton update. We develop an update that generates QP approximations of a desired structure while also allowing superlinear convergence.
- (iii) It is seen that obtaining Lagrange multiplier estimates poses a unique challenge in our method and we show that these estimates are cheaply obtainable through the solution of a single low-dimensional quadratic program. This is essential to the scalability of the scheme.

Numerical simulations show that the method scales linearly with  $|\Upsilon|$  and the framework can address problems whose deterministic equivalents are well over *a few hundred thousand variables and constraints*.

Stochastic programming formulations were separately suggested by Dantzig [16] and Beale [3] for planning under uncertainty. In 1969, based on a decomposition scheme suggested by Benders [4], the L-shaped method was presented for solving two-stage stochastic linear programs [63]. In settings where  $|\Upsilon|$  is infinite, sampling methods are often employed. In the 90s, work by Dantzig, Glynn, Infanger, Hiple and Sen [17, 37, 31, 18, 17, 36, 57] concentrated on integrating Monte-Carlo sampling schemes within the L-shaped method [63]. Subsequent efforts by Shapiro, Robinson, Linderoth, Wright and others aimed at using sample-average approximation (SAA) methods

and their variants [59, 60, 41, 50] for solving a broad class of optimization problems. While much of the sample-average approximation framework focuses on using a fixed-sample size, recent efforts by Deng and Ferris, Homem-De-Mello and others [19, 35] have concentrated on developing iterative schemes with variable sample sizes. Convex stochastic programs have recently seen much study through the use of both primal-dual and dual decomposition methods [5, 54]. Recent work by Shanbhag et al. [57] examined stochastic convex programming through the use of an L-shaped method. The crux of the work focused on providing convergence theory for inexact cut L-shaped methods in settings with a finite sample-space. Furthermore, when the sample-space is infinite, the authors provide convergence and confidence statements for the estimators of the value function. However, there are a multitude of practical problems where neither linearity nor convexity can be assumed (cf. [27, 6, 46]).

In spite of their importance, stochastic nonlinear programs like (SNLP) have been relatively under studied. Theoretical work in explicitly stating the Karush-Kuhn-Tucker (KKT) conditions for the stochastic convex programs is found in the work by Rockafellar and Wets [53]. More recent algorithmic work can be found in [9, 10, 69, 68]. In a recent paper by Liu and Zhao [43] multistage stochastic nonlinear programs were addressed by using an SQP method with the scenario analysis technique of Rockafellar and Wets [52]. Their algorithm reformulates the objective of (SNLP) into a sum of scenario based objectives each with a different first-stage variable. The nonanticipativity condition, which requires that all states which are observationally indistinguishable should have identical optimal value for the decision variable, is imposed explicitly by adding a set of linear equality constraints. It is notable that ours is a significantly different approach.

Also under studied is (SNLP) with  $\Upsilon$  infinite, and the various issues of feasibility and optimality that arise in this case. The work by Walkup and Wets [65] appears to be one of the few in this direction and their contributions are surveyed in section 2.

The remainder of the paper is organized into six sections. We begin by considering a general  $\Upsilon$  in section 2 wherein we study feasibility and recourse in stochastic nonlinear programs with a view towards extending the W-condition to nonlinear settings. Furthermore, we define complete and relatively complete recourse in nonlinear settings and we give sufficient conditions for these to hold. In section 3, we present an outline of our hybrid SQP framework, focusing on the quasi-Newton update and the globalization scheme. Section 4 is organized around the two extensions of the L-shaped method for the stochastic QP subproblems. We show why the Lagrange multiplier estimates are not directly obtained and how they can be extracted through the solution of a small quadratic program. In section 5, we demonstrate the scalability and efficiency of the algorithm on a test problem set of stochastic NLPs. We conclude in section 6.

## 2 Feasibility and recourse in stochastic nonlinear programs

We now begin the first set of contributions of this paper. Consider the stochastic nonlinear program (SNLP) in which  $\Upsilon$  may be an infinite set. Recall from Definition 1.1 that  $Y$  was defined as the space of  $\Upsilon \rightarrow \mathbb{R}^N$  continuous functions.

**Assumption 2.1** *We assume that for every almost-sure feasible  $x$ , there exists  $y \in Y$  such that  $y(\omega)$  is a solution to  $(RNLP(x; \omega))$ .*

Notice that while random variables in general need only be measurable, we additionally require that the optimal second stage decision be a continuous  $\Upsilon \rightarrow \mathbb{R}^N$  function. This will particularly be useful in the proof of Theorem 2.13. From a modeling standpoint, by allowing only continuous second stage decisions, we are implicitly assuming that the decision maker does not gain from considering any discontinuous recourse decisions as a possible response to the uncertain future. Indeed for practical problems of interest being captured by (SNLP), we believe that the class of measurable functions may be too broad for second stage decisions. We expect that, in problems of our interest, where the other problem data is smoothly varying with respect to  $\omega$ , the second stage optimal decisions are also well-behaved (in particular, that they are continuous with respect to  $\omega$ ). Furthermore, when  $\Upsilon$  is finite,  $Y$  contains all random variables, whereby in the more conventional case of finite  $\Upsilon$ , no generality is lost.

When the constraints of an optimization problem are stochastic there are several interpretations of “feasibility”. We consider two notions of feasibility introduced by Walkup and Wets in [65] and study how they affect the first-stage feasibility set of (SNLP), the boundedness of  $\mathcal{Q}(x; \omega)$  and the optimal value of  $(RNLP(x; \omega))$ . We then derive a sufficiency condition under which these interpretations are “equivalent”, in a sense made precise below. Admittedly, the questions we deal with here are subtle and perhaps of only academic interest. But they are

seen to have important and interesting implications on L-shaped algorithms and sampling-based methods, as mentioned in section 2.4. We thus find it apt to present them along with the Benders-SQP algorithm that will follow in section 3.

Let  $K_1 = \{x \mid u(x) = 0, x \geq 0\}$  be the first stage feasibility set and  $\tilde{\Upsilon}$  be defined as

$$\tilde{\Upsilon} = \bigcap_{U \in \mathcal{F}, U \text{ closed}, \mu(U)=1} U.$$

First note that  $\tilde{\Upsilon} \neq \emptyset$ : this follows from observing that for any  $U, V \in \mathcal{F}$ , with  $\mu(U) = \mu(V) = 1$ , we must have  $U \cap V \neq \emptyset$ , since otherwise we would get  $\mu(\Upsilon) \geq \mu(U \cup V) = 2$ . Furthermore,  $\tilde{\Upsilon}$  is closed, hence measurable with  $\mu(\tilde{\Upsilon}) = 1$  and is in fact the smallest closed set of measure 1.  $\tilde{\Upsilon}$  is often called the *support* of  $\mu$ . The two notions of feasibility we consider can be articulated via the sets  $\mathcal{C}_1, \mathcal{C}_2$ , defined as

$$\begin{aligned} \mathcal{C}_1 &= \{x \in K_1 \mid \exists y \in Y : a(x, \omega) + d(y(\omega), \omega) = 0, y(\omega) \geq 0 \ \forall \omega \in \tilde{\Upsilon}\}, \\ \mathcal{C}_2 &= \{x \in K_1 \mid \exists y \in Y : a(x, \omega) + d(y(\omega), \omega) = 0, y(\omega) \geq 0 \ \text{with probability one}\}. \end{aligned}$$

Both  $\mathcal{C}_1$  and  $\mathcal{C}_2$  are reasonable choices for the set of feasible  $x$  and result from two different interpretations of feasibility for stochastic constraints: (1)  $\mathcal{C}_1$  arises from a requirement that  $x \in K_1$  be deemed feasible for (SNLP) if there exists a random variable  $\bar{y} \in Y$  such that  $\bar{y}(\omega)$  is feasible for (RNLP( $x; \omega$ )) for all  $\omega$  lying in  $\tilde{\Upsilon}$ , the support of  $\mu$ ; whereas (2)  $\mathcal{C}_2$  requires that a  $\bar{y}$  exists that satisfies the stochastic constraints almost surely. Clearly,  $\mathcal{C}_1 \subseteq \mathcal{C}_2$ , but in general  $\mathcal{C}_1$  and  $\mathcal{C}_2$  can differ vastly, as can the set over which the function  $\mathcal{Q}(\cdot; \omega)$  for a given  $\omega$  is bounded. Ostensibly, the optimal value of (SNLP) can depend significantly on whether (1) or (2) is the chosen formulation.

Interpretation (1) is more suited to algorithmic implementation, since it involves checking for feasibility on a case-by-case basis for each  $\omega \in \tilde{\Upsilon}$ , while (2) is more conventional from the standpoint of stochastic equations or inclusions. [65] studied these interpretations in the context of the stochastic linear program (StLP):

|      |   |                    |  |
|------|---|--------------------|--|
| StLP | $\begin{aligned} &\underset{x}{\text{minimize}} && c^T x + \mathcal{Q}_L(x) \\ &\text{subject to} && Ax = b \\ &&& x \geq 0, \end{aligned}$ | RLP( $x, \omega$ ) | $\begin{aligned} &\underset{y}{\text{minimize}} && q(\omega)^T y \\ &\text{subject to} && A_\omega x + B_\omega y = b_\omega \\ &&& y \geq 0, \end{aligned}$ |
|------|---|--------------------|--|

where  $\mathcal{Q}_L(x; \omega)$  is the optimal value of (RLP( $x; \omega$ )) and  $\mathcal{Q}_L(x) = \mathbb{E}_\mu[\mathcal{Q}_L(x; \omega)]$ . It was shown in [65], that for such a program, if the ‘‘W-condition’’ below is satisfied, then the above interpretations are *equivalent* in that they result in the same set of feasible first-stage decisions. i.e. if (SNLP) was linear and the W-condition holds, we get  $\mathcal{C}_1 = \mathcal{C}_2$ . Let  $\mathcal{B}$  denote the collection  $\{B_\omega : \omega \in \tilde{\Upsilon}\}$  and for any matrix  $A \in \mathbb{R}^{p \times N}$  define the set-valued map  $\text{pos}(\cdot) : \mathbb{R}^{p \times N} \rightarrow 2^{\mathbb{R}^p}$ , as  $\text{pos}(A) = \{Ay \mid y \geq 0\}$ . Recapitulated below are the relevant results from [65], beginning with the W-condition.

**Definition 2.2 (W-condition for stochastic linear programs)** *The stochastic linear program (StLP) is said to satisfy the W-condition if the restrictions of set-valued maps  $\text{pos}(B_\omega)$  and  $\text{pos}([B_\omega^T, -B_\omega^T, I])$  to  $\mathcal{B}$  and  $\{[B^T, -B^T, I] : B \in \mathcal{B}\}$ , respectively are continuous.*

The main result from [65], articulating equivalence between the first-stage feasible sets, is reproduced below.

**Theorem 2.3** *Consider (StLP) and define*

$$\mathcal{Q}_L(x; \omega) = \begin{cases} \text{optimal value of } (sub_\omega), & \text{if } (sub_\omega) \text{ is feasible,} \\ +\infty, & \text{if } (sub_\omega) \text{ is infeasible.} \end{cases}$$

*If (StLP) satisfies the W-condition, then*

$$\{x \mid -\infty < \mathcal{Q}_L(x; \omega) < \infty \ \text{with probability 1}\} = \bigcap_{\omega \in \tilde{\Upsilon}} \{x \mid -\infty < \mathcal{Q}_L(x; \omega) < \infty\}.$$

Walkup and Wets [65] also observed that if (a)  $B_\omega \in \mathbb{R}^{p \times N}$  is constant for all  $\omega \in \Upsilon$ , or (b)  $\text{pos}(B_\omega) = \mathbb{R}^p$  for all  $\omega \in \Upsilon$ , then the W-condition holds. (a) is called the condition of *fixed recourse*, and (b) is called *complete recourse* – conditions which since have proliferated in stochastic programming literature, including that which assumed  $|\Upsilon| < \infty$ . The case where  $B_\omega$  is *not* a constant with respect to  $\omega$  is simply called *random recourse*. Given below is an illuminating example extended to the quadratic setting from [8, page 87], that illustrates the complications that result from random recourse.

**Example 2.4** Suppose  $\Upsilon = [0, 1]$ ,  $\mu$  satisfies  $\mu(\omega : \omega \leq u) = u^3$ , and

$$\mathcal{Q}_Q(x; \omega) = \min_y \{y^2 \mid \omega y = 1 - x, y \geq 0\},$$

with the convention that if  $\{y \geq 0 \mid \omega y = 1 - x\} = \emptyset$ , then  $\mathcal{Q}_Q(x; \omega) = +\infty$ . It is clear that for all  $\omega \in (0, 1]$ , the optimal  $y$  is  $\frac{1-x}{\omega}$ , and is required to be nonnegative. Therefore

$$\{x \mid -\infty < \mathcal{Q}_Q(x; \omega) < \infty, \quad \forall \omega \in (0, 1]\} = \{x \mid x \leq 1\}.$$

For  $\omega = 0$ , the problem is feasible only for  $x = 1$ . Thus

$$\{x \mid -\infty < \mathcal{Q}_Q(x; 0) < \infty\} = \{1\} \quad \text{and} \quad \mathcal{Q}_Q(x; 0) = 0.$$

Observe that  $\tilde{\Upsilon} = \Upsilon$ . Furthermore, since  $\mu(\omega : \omega = 0) = 0$  we get

$$\begin{aligned} \mathcal{C}_1 &= \{x \mid -\infty < \mathcal{Q}_Q(x; \omega) < \infty \quad \forall \omega \in \tilde{\Upsilon}\} = \{1\}, \\ \mathcal{C}_2 &= \{x \mid \mu(\omega : -\infty < \mathcal{Q}_Q(x; \omega) < \infty) = 1\} = \{x \mid x \leq 1\}. \end{aligned}$$

So if  $\mathcal{C}_1$  is the feasible set,

$$\mathcal{Q}_Q(x) = \int_0^1 \frac{(1-x)^2}{\omega^2} 3\omega^2 d\omega = 3(1-x)^2.$$

However, if  $\mathcal{C}_2$  is the feasible set,  $\mathcal{Q}_Q(x) = 0$ . Observe that if one were to consider the same problem as above but with  $\Upsilon = [\epsilon, 1 + \epsilon]$ ,  $\epsilon > 0$  and proceed exactly as above, one would obtain  $\mathcal{C}_1 = \mathcal{C}_2 = \{x \mid x \leq 1\}$ . Furthermore, if one assumed fixed recourse by replacing the constraint  $\omega y = 1 - x$  by  $\gamma y = 1 - x$  for some constant  $\gamma > 0$  and proceeded as above, one would get  $\mathcal{C}_1 = \mathcal{C}_2$ . For the  $1 \times 1$  matrix  $\omega$ ,  $\text{pos}(\omega)$  is not upper semicontinuous w.r.t  $\omega$  at  $\omega = 0$ , since  $\text{pos}(0) = 0$  and  $\text{pos}(\omega) = \mathbb{R}_+$  for  $\omega > 0$ . In the following section, we show that this continuity requirement is precisely what suffices to ensure  $\mathcal{C}_1 = \mathcal{C}_2$  and constitutes the “W-condition” for convex programs with polyhedral constraints.

The goal of this section is to obtain a condition similar to Definition 2.2, for stochastic convex programs, to allow for an equality between appropriately defined sets corresponding to  $\mathcal{C}_1$  and  $\mathcal{C}_2$ . Section 2.1 derives such a condition for stochastic convex programs with linear constraints, while in 2.2 we extend the condition to the case of general nonlinear constraints. In section 2.3, we define complete and relatively complete recourse in the nonlinear context and examine what sufficient conditions lead to these recourse properties. Finally, in section 2.4, we conclude with an examination of the implications of our findings on L-shaped methods.

## 2.1 A W-condition for stochastic convex programs with linear constraints

We show that for stochastic convex programs with linear constraints, the W-condition can be directly extracted from some results in [65]. Consider the problem which has the same constraints as (StLP):

|     |  |
|-----|--|
| StP | $\begin{aligned} &\underset{x}{\text{minimize}} && f(x) + \mathcal{Q}(x) \\ &\text{subject to} && Ax = b \\ &&& x \geq 0, \end{aligned}$ |
|-----|--|

where we define  $\mathcal{Q}(x; \omega)$  as the optimal value of

|                        |   |
|------------------------|---|
| $\text{P2}(x; \omega)$ | $\min_y \{h(y, \omega) : A_\omega x + B_\omega y = b_\omega, y \geq 0\},$ |
|------------------------|---|

if  $(P2(x; \omega))$  is feasible and  $Q(x; \omega) = +\infty$ , if it is infeasible. Additionally, we assume that  $h(\cdot, \omega)$  is a convex  $C^1$  function with *bounded level sets*, for each  $\omega \in \Upsilon$ . and  $f(\cdot)$  is a real-valued function, is not necessarily convex. We further assume that  $A_\omega, B_\omega$  and  $h(\cdot, \omega)$  are continuous with respect to  $\omega$ . For a fixed  $x$  the optimality conditions of  $P2(x, \omega)$  are finitely many algebraic equations (and inequalities), namely the KKT conditions, implying that  $Q(x; \omega)$  is a measurable function of  $\omega$ . The set of  $x$  that is a.s. permissible is denoted by  $\mathbf{K}$  and can be written using only  $Q(x; \omega)$  in the following way:

$$\mathbf{K} = \{x \mid (P2(x; \omega)) \text{ is feasible and bounded a.s.}\} = \{x \mid -\infty < Q(x; \omega) < \infty \text{ a.s.}\}.$$

Following [65], we consider the sets

$$\begin{array}{ll} \text{feasibility set} & K_2 := \{x \mid Q(x; \omega) < +\infty \text{ a.s.}\}, \\ \text{elementary feasibility sets} & K_2(\omega) := \{x \mid Q(x; \omega) < +\infty\}, \\ \text{dual feasibility set} & K_2^* := \{x \mid Q(x; \omega) > -\infty \text{ a.s.}\}, \\ \text{elementary dual feasibility sets} & K_2^*(\omega) := \{x \mid Q(x; \omega) > -\infty\}. \end{array} \quad (1)$$

It follows that  $\mathbf{K} = K_2 \cap K_2^*$ . Consequently,

$$\bigcap_{\omega \in \tilde{\Upsilon}} K_2(\omega) \subseteq K_2 \quad \text{and} \quad \bigcap_{\omega \in \tilde{\Upsilon}} K_2^*(\omega) \subseteq K_2^* \implies \bigcap_{\omega \in \tilde{\Upsilon}} (K_2(\omega) \cap K_2^*(\omega)) \subseteq \mathbf{K}. \quad (2)$$

We are now prepared to identify a W-condition analogous to Definition 2.2 for stochastic convex programs so that the last inclusion in (2) holds with equality. Observe that if  $(P2(x; \omega))$  is feasible, it is always bounded below (by its unconstrained minimum). This is a point of departure from LPs where feasibility does not necessarily imply boundedness. Consequently,

$$K_2(\omega) \subseteq K_2^*(\omega) \quad \forall \omega \in \Upsilon.$$

The next result provides a sufficiency condition for drawing an equivalence between the feasibility sets arising from the two different interpretations of feasibility.

**Theorem 2.5** *Consider the stochastic convex program (StP). If the restriction of  $pos(\cdot)$  to  $\mathcal{B}$  is continuous, then*

$$\{x \mid Q(x; \omega) < +\infty \quad \forall \omega \in \Upsilon\} = \{x \mid Q(x; \omega) < +\infty, \quad \forall \omega \in \tilde{\Upsilon}\}.$$

**Proof :** Since  $(P2(x; \omega))$  has the same constraints as  $(\text{sub}_\omega)$ , the result follows directly from [65] (Th. 3.7).  $\blacksquare$   
This gives us the W-condition for convex stochastic programs.

**Definition 2.6 (W-condition for convex stochastic programs with bounded level sets)** *The problem (StP) is said to satisfy the W-condition if the restriction of  $pos(\cdot)$  to  $\mathcal{B}$  is continuous.*

Using the W-condition and a bounded level sets assumption, we may provide a simpler characterization for  $\mathbf{K}$ .

**Theorem 2.7** *Consider the stochastic convex program (StP) where the level sets of the second-stage recourse problems are assumed to be bounded for all  $\omega \in \Upsilon$  and the W-condition holds. For such a program we have that*

$$\mathbf{K} = K_2.$$

**Proof :** By Theorem 2.5,  $K_2 = \bigcap_{\omega \in \tilde{\Upsilon}} K_2(\omega)$ . Combining with Eq (2),

$$K_2 = \bigcap_{\omega \in \tilde{\Upsilon}} K_2(\omega) \subseteq \bigcap_{\omega \in \tilde{\Upsilon}} K_2^*(\omega) \subseteq K_2^*.$$

Hence  $\mathbf{K} = K_2 \cap K_2^* = K_2$ .  $\blacksquare$

Next we show that for  $|\Upsilon| < \infty$ , condition in Definition 2.6 can be ensured simply, by proving that the restriction of  $pos(\cdot)$  to  $\mathcal{B}$  is continuous if  $\Upsilon$  is a finite set. Recall the definition of upper semicontinuity of set-valued maps.

**Definition 2.8 (Upper semicontinuity of set-valued maps [2])** Let  $X$  and  $Y$  be metric spaces. A set-valued map  $F : X \mapsto Y$  is called upper semicontinuous at  $x \in \text{dom}(F)$  if and only if for any neighborhood  $\mathcal{U}$  of  $F(x)$ ,

$$\exists \eta > 0 \text{ such that } \forall x' \in B_X(x, \eta), \quad F(x') \subset \mathcal{U},$$

where  $B_X(x, \eta)$  is a ball defined on  $X$  centered at  $x$  and with radius  $\eta$ . A map is said to be upper semicontinuous if it is upper semicontinuous for all  $x \in X$ .

**Proposition 2.9** For a finite set  $\Upsilon$ , the restriction of  $\text{pos}(\cdot)$  to  $\mathcal{B}$  is continuous.

**Proof :** Proving such a result requires proving the upper and lower semicontinuity of  $\text{pos}(\cdot)$ . It is known that  $\text{pos}(\cdot)$  is lower semicontinuous [64]. Hence we only need to prove upper semicontinuity.

Denote the restriction of  $\text{pos}$  to  $\mathcal{B}$  by  $\text{pos}|_{\mathcal{B}}$ . Let  $B_\omega \in \text{dom}(\text{pos}|_{\mathcal{B}})$ , pick an arbitrary  $\omega \in \tilde{\Upsilon}$  and consider a neighborhood  $\mathcal{U}$  around  $\text{pos}(B_\omega)$ . Since  $\Upsilon$  is a finite set,  $\tilde{\Upsilon}$  and  $\mathcal{B}$  are also finite. There exists a neighborhood  $\mathcal{N} \subset \mathbb{R}^{p \times N}$  of  $B_\omega$  such that  $\mathcal{N} \cap \mathcal{B} = \{B_\omega\}$ . Furthermore, for all  $B \in \mathcal{N} \setminus B_\omega$ ,  $\text{pos}|_{\mathcal{B}}(B) = \emptyset$ . So for all  $B \in \mathcal{N}$ ,  $\text{pos}(B) \subset \mathcal{U}$ . It follows that  $\text{pos}|_{\mathcal{B}}$  is continuous. ■

We point out some insights obtained from these results that apply to the formulations of stochastic programs. Note that the W-condition does not eliminate the possibility that there exists  $\tilde{x}$ , with  $\mathcal{Q}(\tilde{x}) < \infty$  but  $\mathcal{Q}(\tilde{x}; \tilde{\omega}) = +\infty$  for some  $\tilde{\omega} \in \Upsilon$ . Traditional stochastic programs were formulated to require feasibility of  $(\text{RNLP}(\tilde{x}; \omega))$  for all  $\omega \in \Upsilon$ . For such a program  $\tilde{x}$  could qualify as infeasible. Scenarios like  $\tilde{\omega}$  introduce constraints that restrict the feasible region but leave the objective unaffected. Thus a stochastic program requiring feasibility of  $(\text{RNLP}(\tilde{x}; \omega))$  for all  $\omega \in \Upsilon$  may have *no feasible first-stage solution* if one added appropriately chosen constraints corresponding to measure zero events. Note however that  $\tilde{x}$  is feasible under the larger ‘‘almost surely’’ sense of Definition 1.1. To obtain the almost sure optimum, in some settings, it may be possible to prune  $\Upsilon$  and perform stochastic programming on a smaller set of scenarios (but of measure one). If the smaller set is  $\tilde{\Upsilon}$ , the W-condition ensures an equivalence to the almost sure case from the standpoint of feasible first-stage decisions. If  $\Upsilon$  is finite, this pruning can be done trivially since scenarios  $\tilde{\omega}$  as above have  $\mu(\tilde{\omega}) = 0$  and can be eliminated at the outset. This also leads us to an important insight into decision making under uncertainty. Often in real life problems, scenarios that lead to infinite returns also have an infinitesimal likelihood of occurrence. What we have presented above is a mathematical justification of why, if almost sure feasibility/optimality is all one desires and the W-condition holds, making decisions by considering the smallest closed scenario set of measure 1 (i.e.  $\tilde{\Upsilon}$ ) suffices.

## 2.2 Extension to stochastic convex programs with nonlinear constraints

We consider (SNLP) with the property that  $h(\cdot, \omega)$  is convex with bounded level sets to articulate a ‘‘W-condition’’ for such a problem class. Using the arguments used in the previous section, we see that if  $(\text{RNLP}(x; \omega))$  is feasible, it is bounded. Thus we only concern ourselves with the question of feasibility and the interpretations at the beginning of section 2. Clearly, when the constraints of (SNLP) are nonlinear a W-condition that is structurally similar to Definition 2.2 can be difficult to obtain. We observe that a more elegant condition can be obtained by making a simple (and not very restrictive) assumption about the probability measure.

**Assumption 2.10** For all  $U \in \mathcal{F}$ ,  $U$  open,  $\mu(U) > 0$ .

The Lebesgue measure has this property, implying  $\mu$  is well defined. Recall the sets  $\mathcal{C}_1$  and  $\mathcal{C}_2$  from section 2. We first define the W-condition and then prove that it ensures  $\mathcal{C}_1 = \mathcal{C}_2$ .

**Definition 2.11 (W-condition for stochastic nonlinear programs)** The problem (SNLP) is said to satisfy the W-condition if  $a(x, \omega)$  and  $d(z, \omega)$  are continuous with respect to  $\omega$  for any  $x, z$ .

We begin by proving a technical result that shows that any set of measure 1 is dense in  $\tilde{\Upsilon}$ .

**Lemma 2.12** Let  $U \in \mathcal{F}$  be any set of measure 1.  $U$  is dense in  $\tilde{\Upsilon}$ .

**Proof :** Assume the contrary. There exists  $U \in \mathcal{F}$  of measure 1, a point  $\omega \in \tilde{\Upsilon}$  and an open neighborhood  $V \in \mathcal{F}$  of  $\omega$  such that  $V \cap U = \emptyset$ . But by Assumption 2.10,  $\mu(V) > 0$ , implying that  $\mu(\Upsilon) > 1$ . Through such a contradiction, the required result follows. ■

The theorem below is our central equivalence result. Recall that  $Y$  is the space of functions that are continuous on  $\Upsilon$ .

**Theorem 2.13** *If (SNLP) satisfies the W-condition of Definition 2.11,  $\mathcal{C}_2 = \mathcal{C}_1$ .*

**Proof :** Since  $\mathcal{C}_1 \subseteq \mathcal{C}_2$ , we concentrate on showing the reverse inclusion. Let  $x \in \mathcal{C}_2$  and let  $y \in Y, U \in \mathcal{F}$ ,  $\mu(U) = 1$  such that

$$a(x, \omega) + d(y(\omega), \omega) = 0, \quad y(\omega) \geq 0 \quad (3)$$

for all  $\omega \in U$ . Let  $\omega' \in \tilde{\Upsilon}$  be arbitrary. By Lemma 2.12, for each open neighborhood  $V$  of  $\omega'$ ,  $V \cap U \neq \emptyset$ . Thus there exists  $\omega_k, k = 1, 2, \dots$  each of which satisfies (3) and  $\omega_k \rightarrow \omega'$ . Recall from the definition of  $Y$  from section 1, that if  $y \in Y$  then  $y(\omega)$  is a continuous function of  $\omega$ . So  $a(x, \omega) + d(y(\omega), \omega)$  is continuous in  $\omega$ , and we conclude that  $\omega'$  satisfies (3). Thus

$$a(x, \omega) + d(y_{|\tilde{\Upsilon}}(\omega), \omega) = 0 \quad y_{|\tilde{\Upsilon}}(\omega) \geq 0 \quad \forall \omega \in \tilde{\Upsilon},$$

implying that  $x \in \mathcal{C}_1$ . This completes the proof.  $\blacksquare$

Observe that we have been able to obtain a W-condition in Definition 2.11 that is weaker than the one obtained by Walkup and Wets in Definition 2.2 (since continuity of  $B_\omega, A_\omega$  with respect to  $\omega$  only implies lower semicontinuity of  $\text{pos}(B_\omega)$ ). This has been facilitated by Assumption 2.10.

## 2.3 Nonlinear complete and relatively complete recourse

When working with polyhedral constraints, ensuring complete or relatively-complete recourse requires that the recourse matrices satisfy certain properties. The analogues of these sufficiency conditions in nonlinear settings forms the subject of this subsection. Throughout this subsection we consider a slight modification of (SNLP), shown below

|  |
|--|
| $\begin{array}{ll} \text{SNLP}' & \begin{array}{l} \underset{x}{\text{minimize}} \quad f(x) + \mathcal{Q}(x) \\ \text{subject to} \quad \begin{array}{l} u(x) \geq 0 \\ x \geq 0, \end{array} \end{array} \end{array}$ |
|--|

where  $\mathcal{Q}(x) = \mathbb{E}_\mu [\mathcal{Q}(x; \omega)]$  and  $\mathcal{Q}(x; \omega)$  is the optimal value of

|   |
|---|
| $\text{RNLP}'(x; \omega) \quad \min_y \{h(y, \omega) : a(x, \omega) + d(y, \omega) \geq 0, y \geq 0\}.$ |
|---|

Furthermore,  $h(\cdot, \omega) \in C^1$  is convex with bounded level sets,  $a(\cdot, \omega), d(\cdot, \omega)$  are  $C^1$  functions and  $d(\cdot, \omega)$  is concave for all  $\omega \in \Upsilon$ . A consequence of complete recourse in (StLP), apart from the satisfaction of the W-condition, is that it ensures for any  $x$  the existence of a recourse decision  $y \in Y$  satisfying  $A_\omega x + B_\omega y(\omega) = b_\omega \quad \forall \omega \in \Upsilon$ . A concept closely related to complete recourse is *relatively complete recourse*. (StLP) has relatively complete recourse if for all  $x$  satisfying  $Ax = b, x \geq 0$ , there exists  $y \in Y$  satisfying  $A_\omega x + B_\omega y(\omega) = b_\omega, y(\omega) \geq 0$  for all  $\omega \in \Upsilon$ . We extend these notions to nonlinear recourse.

**Definition 2.14 (Nonlinear complete recourse)** *The stochastic nonlinear program (SNLP') has complete recourse if for any  $x$ , there exists  $\bar{y} \in Y$  such that  $\bar{y}(\omega)$  solves (RNLP')(x;  $\omega$ ) for all  $\omega \in \Upsilon$ .*

**Definition 2.15 (Nonlinear relatively complete recourse)** *The stochastic nonlinear program (SNLP') has relatively complete recourse if for any  $x$  in  $\{x : u(x) \geq 0, x \geq 0\}$ , there exists  $\bar{y} \in Y$  such that  $\bar{y}(\omega)$  solves (RNLP')(x,  $\omega$ ) for all  $\omega \in \Upsilon$ .*

We now provide simple sufficiency conditions for complete (and relatively complete) recourse using the following proposition from [23], Vol I.

**Proposition 2.16 (Prop. 2.2.7 [23])** *Let  $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$  be continuous and  $K \subseteq \mathbb{R}^N$  be a closed convex set. If there exists a vector  $z^{\text{ref}} \in K$  such that*

$$F(z)^T (z - z^{\text{ref}}) \geq 0, \quad \forall z \in K,$$

*then  $VI(K, F)$  has a nonempty compact solution set.*

It is easy to see that  $K(x, \omega) = \{y | a(x, \omega) + d(y, \omega) \geq 0, y \geq 0\}$  is a closed convex set. By convexity of  $h(\cdot, \omega)$ ,  $(\text{RNLP}'(x; \omega))$  has a solution  $y^*$  if and only if

$$\nabla h(y^*, \omega)^T (y - y^*) \geq 0 \quad \forall y \in K(x, \omega).$$

We now state a result that provides a sufficient condition for being able to take complete and relatively complete recourse in nonlinear settings.

**Theorem 2.17** *For any  $x$  and  $\omega \in \Upsilon$ , if there exists a  $y^{ref} \in Y$  such that  $y^{ref}(\omega) \in K(x, \omega)$ , such that*

$$\nabla h(y, \omega)^T (y - y^{ref}(\omega)) \geq 0 \quad \forall y \in K(x, \omega),$$

*then (SNLP') has complete recourse. Furthermore, if such a  $y^{ref}$  exist for all  $x \geq 0$  such that  $u(x) \geq 0$ , the problem has relatively complete recourse.*

## 2.4 Implications on the L-shaped method

We consider (SNLP) where  $f(\cdot)$  and  $h(\cdot, \omega)$  are convex with bounded level sets for each  $\omega \in \Upsilon$  to highlight some applications of the preceding theory to L-shaped methods for solving such programs. Recall from Definition 1.1 the definition of almost sure feasible  $x$ . Using this we write (SNLP) as

$$\min \left\{ f(x) + \int_{\Upsilon} \mathcal{Q}(x; \omega) d\mu \quad : \quad x \in K_1 \cap \mathcal{C}_2 \right\} \equiv \min \left\{ f(x) + \int_{\tilde{\Upsilon}} \mathcal{Q}(x; \omega) d\mu \quad : \quad x \in K_1 \cap \mathcal{C}_2 \right\} \quad (4)$$

Assume that the the W-condition is satisfied. Then (4) is equivalent to

$$\min \left\{ f(x) + \int_{\tilde{\Upsilon}} \mathcal{Q}(x; \omega) d\mu \quad : \quad x \in K_1 \cap \mathcal{C}_1 \right\}. \quad (5)$$

We now argue that L-shaped methods that are applied to the case of infinite  $\Upsilon$  via sampling from  $\Upsilon$ , may converge to a suboptimal solution of (SNLP) if a certain precaution is not taken. Such L-shaped methods proceed iteratively by solving the first-stage problem to get  $\hat{x} \in K_1$  and then solving  $(\text{RNLP}(\hat{x}; \omega))$  to find a solution  $\hat{y}(\omega)$  for each  $\omega$  in a sample set  $\hat{\Upsilon} \subseteq \Upsilon$ . If all  $(\text{RNLP}(\hat{x}, \omega))$  are feasible for *all*  $\omega \in \hat{\Upsilon}$ , they constrain the first-stage problem with an optimality cut. If  $(\text{RNLP}(\hat{x}; \tilde{\omega}))$  is infeasible for some  $\tilde{\omega} \in \hat{\Upsilon}$ , the method adds a feasibility cut. Note, however, that scenarios in  $\Upsilon \setminus \tilde{\Upsilon}$  do not appear in (5) and such scenarios are inconsequential to the objective, the feasible region and indeed the optimal solution of (SNLP). Thus, in the L-shaped method, if  $(\text{RNLP}(\hat{x}; \tilde{\omega}))$  is infeasible for some  $\tilde{\omega} \in \hat{\Upsilon}$ , one need add a feasibility cut *if and only if*  $\tilde{\omega} \in \tilde{\Upsilon}$ . One should not add feasibility cuts for infeasibility arising from scenarios in  $\Upsilon \setminus \tilde{\Upsilon}$ . If the precaution of not adding feasibility cuts for samples in  $\Upsilon \setminus \tilde{\Upsilon}$  is not taken, feasibility cuts arising out from these scenarios would unduly constrain the first-stage problem, and may eventually result in convergence of the L-shaped method at points that are suboptimal for (4). Alternatively one may employ the L-shaped method with samples  $\hat{\Upsilon}$  chosen only from  $\tilde{\Upsilon}$ . To summarize, if one were to analytically solve (4), none of the above exemptions would be necessary. But in the algorithmic computation of the solution via the L-shaped method feasibility cuts need to be applied with the above caveat in mind in order to obtain the same solution as the one obtained analytically.

## 3 A sequential quadratic programming framework

The rest of this paper is devoted to our second set of contributions, namely development of an SQP method for solving (SNLP). The SQP approach for the solution of nonlinear optimization problems found inception in the Ph.D. thesis of Wilson in 1963 [66]. But the method became truly popular after Hessian-update methods for problems with indefinite Lagrangian Hessians were studied by Mangasarian and Han [11, 29, 30]. Through several papers by Powell in the late 1970s [49, 47, 48], Han's theorems on the convergence of SQP methods became widely known in the optimization community.

The SQP method is an iterative method with its basic idea being to model the nonlinear program at a given iterate  $x_k$  by an approximate quadratic program (QP). The solution of this QP provides a direction  $p_k$  and

an acceptable stepsize  $\alpha_k$  is obtained by conducting a linesearch on an appropriately defined merit function. The new iterate  $x_{k+1}$  is defined as  $x_k + \alpha_k p_k$ . Under some assumptions, it can be shown that the sequence of iterates converges to the KKT point of the original nonlinear program. SQP methods offer great flexibility in the construction of the QP approximations in order to be able to deal with regions with an indefinite Hessian, with the overarching goal of driving the sequence of solutions to a KKT point. With an appropriate choice of Hessian matrices, SQP methods can be shown to be equivalent to a Newton or quasi-Newton method applied to solving the KKT system of the original nonlinear problem [46]. As a result of this equivalence, in regions close to the KKT point of the original problem, with exact Hessians, SQP methods show similar behaviour to Newton methods – that of quadratic local convergence.

If the SQP method is augmented with a merit function, a reduction in which directs the sequence  $\{x_k\}$  toward a KKT point, the SQP method can be proved to be globally convergent [11]. There are several practical SQP methods tailored to various kinds of applications; NPSOL [26] for dense problems, SNOPT [25] for sparse problems and `filtersQP` – a filter method [24] – being some of them. The method that follows is the one employed in the solver SNOPT by Gill, Murray and Saunders [25], which we shall apply to (SNLP). It must be emphasized that, while the outer SQP structure is similar to SNOPT, our framework has a multitude of distinctions from a standard linesearch SQP methods, principal amongst these being the nature of the Hessian update and the subproblem solvers.

In the next two sections, we present the second set of contributions of this paper, namely the Benders-SQP algorithm for solving (SNLP). This section, in particular, discusses the outer SQP framework, assuming that the resulting quadratic programming subproblems can be solved. Special emphasis will be given to making the algorithm *scalable*, *parallelizable* and *fast locally convergent*. We assume that  $\omega$  takes values in a *finite* set  $\Omega$  (may be thought of as a set of samples of  $\Upsilon$ ) and with a probability given by the counting measure  $\mathbb{P}$ . Let  $\mathbb{E}$  be the expectation with respect to  $\mathbb{P}$ .

The remainder of this section is organized as follows. In section 3.1, we outline our assumptions and provide the notation for our SQP method. The SQP method applied to (SNLP) is described in section 3.2. In section 3.3 we show that one can always get descent in the merit function, a property central to establishing convergence of SQP methods. In 3.4 we outline our sparse Hessian update, termination criteria and prove that the algorithm is superlinearly convergent.

### 3.1 A linesearch-based SQP method

Following are some notational rules we follow during our description of the SQP algorithm. Some of these notations might change meaning later in the paper; the new meaning will be made clear as and when necessary.

**Variables:** We denote  $y, z, \lambda, \mu, \nu$  and  $c(z)$  as

$$y := \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ y^{|\Omega|} \end{pmatrix}, z := \begin{pmatrix} x \\ y \end{pmatrix}, \lambda := \begin{pmatrix} \lambda^x \\ p_1 \lambda^1 \\ p_2 \lambda^2 \\ \vdots \\ p_{|\Omega|} \lambda^{|\Omega|} \end{pmatrix}, \mu := \begin{pmatrix} \mu^x \\ p_1 \mu^1 \\ p_2 \mu^2 \\ \vdots \\ p_{|\Omega|} \mu^{|\Omega|} \end{pmatrix}, \nu := \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \text{ and } c(z) := \begin{pmatrix} u(x) \\ a_1(x) + d_1(y^1) \\ a_2(x) + d_2(y^2) \\ \vdots \\ a_{|\Omega|}(x) + d_{|\Omega|}(y^{|\Omega|}) \end{pmatrix}.$$

#### Subscripts and superscripts:

- We use a superscript  $x$  to denote terms associated with  $f(\cdot)$  or  $u(\cdot)$  or the multipliers. Second-stage variables, constraints or objectives will be denoted by  $\omega \in \{1, \dots, |\Omega|\}$  as superscript. All quantities that are *given* as a part of the definition of SNLP – viz.,  $a_\omega(\cdot), h_\omega(\cdot)$  etc. use the scenario number as subscript and thus are distinguished from quantities that we introduce into the problem – viz.  $y^\omega, \lambda^\omega$  etc.
- We reserve the subscript  $k$  to denote the SQP iteration to which the subscripted variables belong: e.g.  $x_k, y_k^\omega, \lambda_k^x$  etc. For a function  $F(x)$ ,  $F_k$  will denote its value at  $x_k$ , unless stated otherwise.
- The superscripted asterisk denotes the optimal for (SNLP):  $(z^*, \nu^*)$  is the KKT point of (SNLP).
- We use the subscript  $L$  and  $Q$  along with the function to depict its linearization and its quadratic approximation respectively.

**Jacobians and Hessians:** Jacobians and Hessians (or approximations to the Hessian, whichever is applicable) will be denoted by  $J$  and  $H$  respectively. They will be affixed with suitable subscripts and superscripts to denote which SQP iteration and functions they pertain to.

**Miscellaneous:** Let  $\delta_L$  denote the departure from linearity of  $c(z)$  at the current iterate  $z_k$  while  $c_L$  denote its linearization.

$$c_L(z; z_k) := c(z_k) + J_k^c \Delta z_k. \quad \text{and} \quad \delta_L(z; z_k) := c(z) - c_L(z; z_k)$$

Lastly, for any variable  $x$ , we use  $\Delta x_k$  to denote the difference  $(x - x_k)$ .

We make the following assumptions about (SNLP):

### Assumption 3.1

- (a.) *The optimal value of (SNLP) is bounded and denoted by  $v^*$ .*
- (b.) *The functions  $f(\cdot)$ ,  $h_\omega(\cdot)$ ,  $u(\cdot)$ ,  $a_\omega(\cdot)$  and  $d_\omega(\cdot)$  are continuously differentiable functions for all  $\omega \in \Omega$ .*
- (c.) *The Jacobian corresponding to the active constraints at any first-order KKT point has full rank. Strict complementarity holds at all stationary points of (SNLP).*
- (d.) *For all  $\omega \in \Omega$ ,  $\mathbb{P}(\omega) > 0$ .*

Assumption (d) is motivated by the insights listed in section 2.4. It implies that  $\Omega$  is the smallest relatively closed subset of  $\Omega$  with measure 1.

## 3.2 Constructing QP approximations

SQP literature commonly refers to quadratic programs that approximate the original problem as QP *subproblems* [25]. We will continue to refer to them as approximations and instead reserve the term *subproblem* for scenario-wise problems obtained from decomposing the stochastic program. Following [25], the modified Lagrangian is defined as

$$\mathcal{L}(z; \nu_k, z_k) := f(x) + \mathbb{E}[h_\omega(y^\omega)] - \lambda_k^T \delta_L(z; z_k).$$

We distinguish it from the *true* Lagrangian  $\widehat{\mathcal{L}}(z; \nu_k) := f(x) + \mathbb{E}[h_\omega(y^\omega)] - \lambda_k^T c(z) - \mu_k^T z$ . The modified Lagrangian is similar to the true Lagrangian with one crucial difference: instead of the constraint function, the modified Lagrangian uses the *departure* from linearity. Let  $g_k := \nabla \mathcal{L}_k$ . By the procedure described in the preceding section, the QP approximation is

|                 |   |
|-----------------|---|
| $\text{StQP}_k$ | $\begin{aligned} & \underset{z}{\text{minimize}} && \mathcal{L}_k + g_k^T \Delta z_k + \frac{1}{2} \Delta z_k^T H_k \Delta z_k \\ & \text{subject to} && c_L(z; z_k) = 0 \\ & && z \geq 0, \end{aligned}$ |
|-----------------|---|

where  $H_k \succ 0$  is a positive definite approximation to  $\nabla^2 \mathcal{L}_k$ . In constructing the QP approximations using the modified Lagrangian, we maintain feasibility with respect to the linearization. Continuing the same notation as for (SNLP), let  $(\bar{z}_k, \bar{\nu}_k)$  be the KKT point of (StQP<sub>k</sub>). We assume the following about the nature of the solution of (StQP<sub>k</sub>).

**Assumption 3.2** *At the solution of (StQP<sub>k</sub>),  $(\bar{z}_k, \bar{\nu}_k)$ , the constraint Jacobian  $J_k^c$  is full row rank and the Hessian of the Lagrangian  $\nabla^2 \mathcal{L}(\bar{z}_k, \bar{\nu}_k)$  is positive definite in the null-space of the constraints.*

The modified Lagrangian function  $\mathcal{L}$  may be written as  $\mathcal{L} = \mathcal{L}^x + \mathbb{E}[\mathcal{L}^\omega]$ , where

$$\begin{aligned} \mathcal{L}^x(x; x_k, \nu_k) &= f(x) - \lambda_k^{xT} (u(x) - u_L(x; x_k)) - \mathbb{E}[\lambda_k^{\omega T} (a_\omega(x) - a_{\omega L}(x; x_k))] \\ \mathcal{L}^\omega(y^\omega; y_k^\omega, \lambda_k^\omega) &= h_\omega(y^\omega) - \lambda_k^{\omega T} (d_\omega(y^\omega) - d_{\omega L}(y^\omega; y_k^\omega)). \end{aligned}$$

The value and the gradient of the modified Lagrangian at the point of linearization is the same as that of the objective function at that point, while the Hessian of the modified Lagrangian equals the Hessian of the true Lagrangian at all points. i.e.

$$\mathcal{L}(z_k; z_k, \nu_k) = f(x_k) + \mathbb{E}[h_\omega(y_k^\omega)], \quad \nabla_z \mathcal{L}(z_k; z_k, \nu_k) = \begin{pmatrix} \nabla f_k \\ p_1 \nabla h_{1k} \\ \vdots \\ p_{|\Omega|} \nabla h_{|\Omega|k} \end{pmatrix}, \quad \text{and} \quad \nabla_z^2 \mathcal{L}(z; z_k, \nu_k) = \nabla_z^2 \widehat{\mathcal{L}}(z; \nu_k).$$

In particular,  $\nabla^2 \mathcal{L}$  is independent of  $z_k$ . See [25, 51, 34] for more on the modified Lagrangian.

From a structural standpoint, the matrices  $H_k$  need to satisfy the following conditions:

1. It must have a block diagonal structure with blocks corresponding to  $x$  and each  $y^\omega$ ,  $\omega \in \Omega$ .
2. Each block should be positive definite.

We will return to the question of constructing such Hessians, with some added requirements in section 3.4. Given Hessians of such structure, the quadratic approximations of the Lagrangian functions may then be stated as

$$\begin{aligned} \mathcal{L}_Q^x(x; x_k, \nu_k) &= \mathcal{L}_k^x + \Delta x_k^T \nabla \mathcal{L}_k^x + \frac{1}{2} \Delta x_k^T H_k^x \Delta x_k \\ \mathcal{L}_Q^\omega(y^\omega; y_k^\omega, \lambda_k^\omega) &= \mathcal{L}_k^\omega + (\Delta y_k^\omega)^T \nabla \mathcal{L}_k^\omega + \frac{1}{2} (\Delta y_k^\omega)^T H_k^\omega \Delta y_k^\omega. \end{aligned}$$

With such a structure, the QP subproblem (StQP<sub>k</sub>) can now be rewritten as a two-stage stochastic program

|                   |            |   |   |
|-------------------|------------|---|---|
| StQP <sub>k</sub> | minimize   | $\mathcal{L}_Q^x(x; x_k, \nu_k) + \mathbb{E}[\mathcal{L}_Q^\omega(y^\omega; y_k^\omega, \lambda_k^\omega)]$ |   |
|                   |            | $J_k^u \Delta x_k = -u_k$   | : $\bar{\lambda}^x$   |
|                   | subject to | $J_k^{d_\omega} \Delta y_k^\omega + J_k^{a_\omega} \Delta x_k = -a_{\omega,k} - d_\omega(y_k^\omega)$       | : $p_\omega \bar{\lambda}^\omega$   |
|                   |            | $x, y^\omega \geq 0,$   | : $\bar{\mu}^x, p_\omega \bar{\mu}^\omega \quad \forall \omega \in \Omega.$ |

In the following sections, we will assume that (StQP<sub>k</sub>) is solvable and that its KKT point can be obtained. Scalable convergent schemes for solving (StQP<sub>k</sub>) are presented in section 4.

### 3.3 A parallelizable linesearch

The augmented Lagrangian merit function for an SQP method on (SNLP) is

$$\mathcal{M}(z, \nu, s; \rho) = f(x) + \mathbb{E}[h_\omega(y^\omega)] - \lambda^T c(z) - \mu^T (z - s) + \frac{1}{2} \rho \sum \|c_i(z)\|^2 + \frac{1}{2} \rho \|z - s\|^2. \quad (6)$$

The linesearch procedure is as follows. Having obtained the KKT point  $(\bar{z}_k, \bar{\nu}_k)$  of (StQP<sub>k</sub>), we determine the slack variables  $\bar{s}_k$  corresponding to  $\bar{z}_k$ . The search direction  $(p_k, \xi_k, q_k)$  in the primal variables, the multipliers and slacks is constructed as  $((\bar{z}_k - z_k), (\bar{\nu}_k - \nu_k), (\bar{s}_k - s_k))$ . For a given  $\rho$ ,  $\alpha_k$  denotes the steplength satisfying the Armijo conditions for  $\mathcal{M}(z_k + \alpha_k p_k, \nu_k + \alpha_k \xi_k, s_k + \alpha_k q_k; \rho)$ . The condition below ensures sufficient descent [28]:

$$\begin{pmatrix} p_k \\ \xi_k \\ q_k \end{pmatrix}^T \nabla \mathcal{M}_k < -\frac{1}{2} p_k^T H_k p_k \leq 0. \quad (7)$$

If (7) is not met, we increase the parameter  $\rho$  so as to satisfy it. The following lemma shows that this can indeed be done.

**Lemma 3.3** *At a given iterate  $(z_k, \nu_k)$  there exists a finite  $\bar{\rho}$  such that for all  $\rho \geq \bar{\rho}$ , the sufficient descent condition (7) is satisfied.*

**Proof :** The derivative of  $\mathcal{M}$  from Eq (6) at  $(z_k, \nu_k, s_k)$  is given by

$$\nabla \mathcal{M}_k = \begin{pmatrix} g_k - (J_k^c)^T \lambda_k - \mu_k + \rho (J_k^c)^T c_k + \rho (z_k - s_k) \\ -c_k \\ -(z_k - s_k) \\ \mu - \rho (z_k - s_k) \end{pmatrix}.$$

Therefore, we have that

$$\begin{pmatrix} p_k \\ \xi_k \\ q_k \end{pmatrix}^T \nabla \mathcal{M}_k = g_k^T p_k + p_k^T J_k^{cT} (\rho c_k - \lambda_k) - \xi_k^T \begin{pmatrix} c_k \\ z_k - s_k \end{pmatrix} + (q_k - p_k)^T (\mu_k - \rho (z_k - s_k)). \quad (8)$$

Observe that since  $\bar{s}_k = \bar{z}_k$ ,  $(q_k - p_k) = (z_k - s_k)$ . And since  $\bar{z}_k$  is feasible for (StQP<sub>k</sub>),  $J_k^c p_k = -c_k$ . Two cases arise.

Case 1: The current iterate is infeasible or  $\|z_k - s_k\|^2 + \|c_k\|^2 > 0$ . Using Eq (8), condition (7) becomes

$$\rho > \frac{-\frac{1}{2} p_k^T H p_k - g_k^T p_k - c_k^T \lambda_k + \begin{pmatrix} \bar{\lambda}_k - 2\lambda_k \\ \bar{\mu}_k - 2\mu_k \end{pmatrix}^T \begin{pmatrix} c_k \\ z_k - s_k \end{pmatrix}}{\|z_k - s_k\|^2 + \|c_k\|^2}. \quad (9)$$

Since  $(z_k, \nu_k) \neq (z^*, \nu^*)$ , the right hand side of (9) is well defined. Hence there exists a  $\rho < \infty$  such that condition (7) is satisfied.

Case 2: The current iterate is feasible or  $\|z_k - s_k\|^2 + \|c_k\|^2 = 0$ . Therefore, from (8), we have that

$$\begin{aligned} \begin{pmatrix} p_k \\ \xi_k \\ q_k \end{pmatrix}^T \nabla \mathcal{M}_k &= g_k^T p_k + p_k^T J_k^{cT} (\rho c_k - \lambda_k) - \xi_k^T \begin{pmatrix} c_k \\ z_k - s_k \end{pmatrix} + (q_k - p_k)^T (\mu_k - \rho (z_k - s_k)) \\ &= g_k^T p_k + p_k^T J_k^{cT} (-\lambda_k) + (q_k - p_k)^T (\mu_k). \end{aligned}$$

But since  $c_k = 0$ , we have that  $J_k^c p_k = 0$ . Also  $(q_k - p_k) = (z_k - s_k) = 0$ , leaving us to prove that

$$g_k^T p_k < -\frac{1}{2} p_k^T H_k p_k.$$

To prove this, observe that  $z = z_k$  is feasible for (StQP<sub>k</sub>), and at the solution of (StQP<sub>k</sub>),  $\Delta z_k = p_k$ . Denote the objective of (StQP<sub>k</sub>) by  $F(z)$ . Hence  $F(z_k + p_k) < F(z_k)$ , implying that  $g_k^T p_k < -\frac{1}{2} p_k^T H_k p_k$ . ■

Given an  $\alpha_k$ , the  $(k+1)^{th}$  iterate is computed as

$$\begin{pmatrix} z_{k+1} \\ \nu_{k+1} \end{pmatrix} = \begin{pmatrix} z_k \\ \nu_k \end{pmatrix} + \alpha_k \begin{pmatrix} \bar{z}_k - z_k \\ \bar{\nu}_k - \nu_k \end{pmatrix}.$$

Observe that the merit function can be written as

$$\begin{aligned} \mathcal{M}(z, \nu, s; \rho) &:= f(x) - (\lambda^x)^T u(x) - (\mu^x)^T (x - s^x) + \frac{1}{2} \rho \sum_i \|u_i(x)\|^2 + \frac{1}{2} \rho \|x - s^x\|^2 \\ &\quad + \mathbb{E} [h_\omega(y^\omega) - (\lambda^\omega)^T (a_\omega(x) + d_\omega(y^\omega)) - (\mu^\omega)^T (y^\omega - s^\omega)] \\ &\quad + \mathbb{E} \left[ \frac{1}{2} \rho \sum_i \|a_{\omega,i}(x) + d_{\omega,i}(y^\omega)\|^2 + \frac{1}{2} \rho \|y^\omega - s^\omega\|^2 \right]. \end{aligned}$$

Hence its evaluation can be parallelized to a large extent by decomposing into first-stage and scenario-wise second-stage evaluations.

### 3.4 A Sparse Quasi-Newton Update

The entire analysis that ensued in the previous section relied on the separability of the objective of (StQP<sub>k</sub>). Indeed, our solution method for (StQP<sub>k</sub>) is also incumbent on this property. The requirements from the Hessian

approximations are (1.) a block diagonal structure, (2.) block-wise positive definiteness, and (3.) to ensure local superlinear convergence and allow updating in parallel. In the SQP method used in SNOPT [25], the authors have employed a modified BFGS update which ensures superlinear convergence. However if we choose to apply this directly on  $H_k$  (the approximation to  $\nabla_z^2 \mathcal{L}$ ), it does not leave  $H_{k+1}$  with the block diagonal structure that we need. Several sparsity preserving updates exist in literature (see Shanno [58], Toint [62] and Lucia [44] for examples). But it is not obvious from the structure of these updates if they can be constructed in a parallelizable manner, so we do not employ any of them. We employ a BFGS update on the blocks, with the update rule defined below.

**Definition 3.4** *Suppose the Hessian of the Lagrangian function at the  $k$ th iterate is denoted by  $H_k$ . Then the updated Hessian, denoted by  $H_{k+1}$ , is given by a relationship, referred to as  $\mathcal{R}(H_k)$ , where  $H_{k+1}$  is defined as  $H_{k+1} = \mathcal{R}(H_k)$  and  $\mathcal{R}(H_k)$  is given by*

$$\begin{aligned} H_{k+1}^x &:= \text{BFGS}(H_k^x), \\ H_{k+1}^\omega &:= \text{BFGS}(H_k^\omega), \quad \forall \omega \in \Omega, \quad H_0 \succ 0, \end{aligned}$$

where  $\text{BFGS}(\cdot)$  is the BFGS update

$$B_{k+1} = B_k + \frac{r_k r_k^T}{r_k^T p_k} - \frac{(B_k p_k)(B_k p_k)^T}{p_k^T B_k p_k}, \quad (10)$$

with  $p_k = z_{k+1} - z_k$ , and

$$r_k = \nabla \mathcal{L}(z_{k+1}; z_k, \nu_{k+1}) - \nabla \mathcal{L}(z_k; z_k, \nu_{k+1}) = \nabla \widehat{\mathcal{L}}(z_{k+1}; z_k, \nu_{k+1}) - \nabla \widehat{\mathcal{L}}(z_k; z_k, \nu_{k+1}).$$

Clearly  $\mathcal{R}(H_k)$  preserves the block diagonal structure of  $H_k$ , but it is not guaranteed to retain the sparsity structure inside individual blocks. Also,  $\mathcal{R}$  can be applied in parallel on individual blocks of  $H_k$  without any information from the other blocks. It is well known that BFGS updates retain positive definiteness as long as  $p_k^T r_k > 0$  [46]. If  $p_k^T r_k$  is not sufficiently positive, we choose to simply skip the update<sup>1</sup>. So we conclude, under  $\mathcal{R}$ ,

$$H_0 \succ 0 \implies H_k \succ 0 \quad \forall k > 0.$$

The following is our Benders-SQP method for solving (SNLP).

---

**Algorithm 1:** Benders-SQP method for Stochastic NLPs

---

o initialization  $k = 1$ ;  
 choose constants  $\tau_P > 0$  and  $\tau_D > 0$ , initial first stage decision  $x_0$ , second stage decisions  $y_0^\omega$ , multipliers  $\lambda_0^x, \lambda_0^\omega, \mu_0^x, \mu_0^\omega \quad \forall \omega \in \Omega$  and initial (SNLP) Hessian  $H_0 \succ 0$ ;  
**while conditions (11) are not satisfied do**  
 1 | Construct QP approximation (StQP $_k$ ) at  $k^{\text{th}}$  iterate;  
 2 | Solve (StQP $_k$ ) using algorithm ILS or TR;  
 3 | Recover first-stage multipliers by solving problem (MULT) in section 4.3;  
 4 | Perform linesearch as shown in section 3.3 and get  $z_{k+1}$ ;  
 5 | Construct new Hessian approximation  $H_{k+1} = \mathcal{R}(H_k)$ ;  
**end**

---

For prespecified small constants  $\tau_P$  and  $\tau_D$  define  $\tau_z = \tau_P(1 + \|z\|)$  and  $\tau_\nu = \tau_D(1 + \|\nu\|)$ . A point  $(z^*, \nu^*)$  is taken to be a KKT point of (SNLP) if it satisfies

$$|c_i(z^*)| \leq \tau_{z^*}, \quad \mu_i^* \geq -\tau_{\nu^*}, \quad |c_i(z^*) \lambda_i^*| \leq \tau_{\nu^*}, \quad z_i^* \mu_i^* \leq \tau_{\nu^*}, \quad |[\nabla \widehat{\mathcal{L}}(z^*, \nu^*)]_i| \leq \tau_{\nu^*}, \quad \forall i. \quad (11)$$

The differences between Algorithm 1 and a standard linesearch SQP method lie in the computation of the search directions (i.e. the QP solver), in the computation of the Hessian update and finally in technique for obtaining multiplier estimates (discussed in section 4.3). The *global* convergence of our method can be directly concluded and is a consequence of prior work on SQP methods. A required condition for global convergence is that  $p_k$  results in descent with respect to a merit function (shown in Lemma 3.3). Additionally, the constructed sequences need

<sup>1</sup>Our numerical results show that this technique works well for our problem.

to satisfy set of properties such as boundedness of iterates, boundedness of the penalty parameters and that the sequence of steplengths is bounded away from zero (see sections 3.3, 3.4 and 3.5 from [45]). These follow immediately in the current setting, given that our problem is essentially a large structured nonlinear program. We reproduce and reference the main convergence result for the linesearch SQP method.

**Theorem 3.5 (Th. 3.15, 3.17 [45])** *The sequence of primal iterates and Lagrange multipliers produced by Algorithm 1 converges to a unique KKT point.*

We rely on this result for claiming convergence of our scheme and now proceed to discuss the question of the rate of convergence of iterates close to the solution.

Superlinear convergence is a different matter primarily because we do not have the same Hessian update as in deterministic NLPs. Our next goal is to show that  $\mathcal{R}$  provides us with superlinear convergence. We assume that  $r_k^T p_k > 0$  (cf. Definition 3.4) for all  $k$  such that  $z_k$  is a small neighborhood of  $z^*$ . This assumption is standard and it ensures that the update  $\mathcal{R}$  is not skipped. (If in the neighborhood of  $z^*$ ,  $r_k^T p_k$  becomes 0 for some  $k$ , superlinear convergence be known to degenerate for SQP methods with BFGS updates too.) Quasi-Newton methods for constrained optimization requires the satisfaction of the following test [12, 61, 46] for superlinear convergence:

**Theorem 3.6** *Suppose Assumption 3.2 holds and the iterates  $z_k$  obtained from Algorithm 1 converge to  $z^*$ , a solution of (SNLP). Then  $z_k \rightarrow z^*$  superlinearly if and only if*

$$\lim_{k \rightarrow \infty} \frac{\|P_k(B_k - \nabla_z^2 \widehat{\mathcal{L}}^*)(z_{k+1} - z_k)\|}{\|z_{k+1} - z_k\|} = 0, \quad (12)$$

where  $P_k = I - (J_k^c)^T [J_k^c J_k^{cT}]^{-1} J_k^c$  is the projection on the null-space of the constraints and  $\widehat{\mathcal{L}}$  denotes the true Lagrangian (not the modified Lagrangian).

The following theorem combines the Dennis-Moré characterization theorem for quasi-Newton methods [20, 46] with the fact that BFGS updates provide superlinear convergence (see [46] for a proof of this fact).

**Theorem 3.7** *For some function  $F$ , let  $z^*$  be a minimum. Suppose iterates  $z_k$  generated using a rule  $z_{k+1} = z_k - B_k^{-1} \nabla F_k$ , converge to  $z^*$ . Then  $z_k \rightarrow z^*$  superlinearly if and only if*

$$\lim_{k \rightarrow \infty} \frac{\|(B_k - \nabla^2 F(z^*))(z_{k+1} - z_k)\|}{\|z_{k+1} - z_k\|} = 0. \quad (13)$$

Furthermore, if  $B_k$  is the BFGS update applied to approximate  $\nabla^2 F_k$ , then (13) holds.

We are now ready to prove the required superlinear convergence result.

**Theorem 3.8** *Suppose the iterates  $z_k$  are obtained from Algorithm 1 converge to  $z^*$ , a solution of (SNLP). Then*

$$\lim_{k \rightarrow \infty} \frac{\|(H_k - \nabla^2 \widehat{\mathcal{L}}(z^*))(z_{k+1} - z_k)\|}{\|z_{k+1} - z_k\|} = 0$$

and  $z_k \xrightarrow{k} z^*$  superlinearly.

**Proof :** Observe that the KKT conditions for (StQP<sub>k</sub>) are

$$H_k p_k + g_k - J_k^{cT} \lambda_k - \mu = 0 \implies p_k = -H_k^{-1} \nabla_z \widehat{\mathcal{L}},$$

since  $\widehat{\mathcal{L}}(z, \nu) = f(x) + \mathbb{E}_\omega h_\omega(y^\omega) - \lambda^T c(z) - \mu^T z$ . Since  $\mathcal{R}$  is block-wise separable,

$$x_{k+1} = x_k - (H_k^x)^{-1} \nabla_x \widehat{\mathcal{L}}_k \quad \text{and} \quad y_{k+1}^\omega = y_k^\omega - (H_k^\omega)^{-1} \nabla_{y^\omega} \widehat{\mathcal{L}}_k, \quad \forall \omega \in \Omega. \quad (14)$$

Let  $\nabla^2 \widehat{\mathcal{L}}^* := \nabla^2 \widehat{\mathcal{L}}(z^*)$ . Now

$$\lim_{k \rightarrow \infty} \frac{\|(H_k - \nabla^2 \widehat{\mathcal{L}}^*)(z_{k+1} - z_k)\|}{\|z_{k+1} - z_k\|} = \lim_{k \rightarrow \infty} \left[ \sum_{\omega \in \Omega} \left( \frac{\|(H_k^\omega - \nabla^2 \widehat{\mathcal{L}}^{\omega*})(y_{k+1}^\omega - y_k^\omega)\|}{\|p_k\|} \right) + \frac{\|(H_k^x - \nabla^2 \widehat{\mathcal{L}}^{\omega*})(x_{k+1} - x_k)\|}{\|p_k\|} \right].$$

Take a typical term from the right side like the one corresponding to  $\omega$ . The sequence of iterates  $\{y_k^\omega\}$  obtained by the SQP method and the Hessian  $H_k^\omega$  is a BFGS approximation to  $\nabla^2 \mathcal{L}_k^\omega = \nabla^2 \widehat{\mathcal{L}}_k^\omega$ . Using Theorem 3.7 we find that each such term in the right hand side of this equation is 0. A similar result holds for the term containing  $H_k^x$  and it follows that

$$\lim_{k \rightarrow \infty} \frac{\|(H_k - \nabla^2 \widehat{\mathcal{L}}(z^*)) (z_{k+1} - z_k)\|}{\|z_{k+1} - z_k\|} = 0$$

We need to show (12) holds with  $B_k = H_k$  obtained by rule  $\mathcal{R}$ .

$$0 \leq \lim_{k \rightarrow \infty} \frac{\|P_k (H_k - \nabla_z^2 \widehat{\mathcal{L}}^*) (z_{k+1} - z_k)\|}{\|z_{k+1} - z_k\|} \leq \lim_{k \rightarrow \infty} \|P_k\| \cdot \lim_{k \rightarrow \infty} \frac{\|(H_k - \nabla_z^2 \widehat{\mathcal{L}}^*) (z_{k+1} - z_k)\|}{\|z_{k+1} - z_k\|}.$$

Since  $\{z_k\}$  converges,  $\lim_{k \rightarrow \infty} \|P_k\|$  exists and is finite. We have shown above that the second limit in the right hand side is zero. This implies that rule  $\mathcal{R}$  satisfies the condition (12) and  $z_k \rightarrow z^*$  superlinearly. ■

## 4 Dual Decomposition Methods for the QP approximations

This section focuses primarily on the decomposition schemes for solving the QP approximations that appear within the SQP framework. After providing some preliminaries on the basic algorithmic framework, namely dual decomposition, we discuss two variants of this decomposition scheme. In both cases, we consider the following stochastic quadratic program which is in the form of (StQP $_k$ ):

|      |   |
|------|---|
| StQP | $\begin{aligned} & \underset{x, y}{\text{minimize}} && \frac{1}{2} x^T P x + c^T x + \mathbb{E} \left[ \frac{1}{2} (y^\omega)^T D_\omega y^\omega + d_\omega y^\omega \right] \\ & \text{subject to} && A x = b \\ & && A_\omega x + B_\omega y^\omega = b_\omega \\ & && x, y^\omega \geq 0, \quad \forall \omega \in \Omega. \end{aligned}$ |
|------|---|

Stochastic QPs, in an *inequality* form, such as when the deterministic equivalent is given by

$$\min_x \left\{ \frac{1}{2} x^T Q x + c^T x : A x \geq b, x \geq 0 \right\},$$

can be written as stochastic linear complementarity problems (LCPs). There have been several attempts made in constructing scalable schemes for solving such problems. In particular, work by Fukushima et al. [40] have considered smoothing schemes while Shanbhag et al. [56] have concentrated on scalable matrix-splitting methods. The latter, in particular, accommodates convex QPs through a Tikhonov regularization. A question worth discussing further is whether an LCP approach may prove more attractive. We contend that in the solution of stochastic QPs, an LCP approach may not prove favorable for at least two reasons. First, the convergence scheme is closely related to the spectral properties of the associated matrices [15] and may not follow from merely imposing strict convexity on the original QP. Second, the scenario subproblems are monotone LCPs over a larger space which are generally more challenging to solve than sparse convex QPs.

Matrices  $P$  and  $D_\omega$  are positive definite for all  $\omega \in \Omega$  while  $B_\omega$  is full row-rank for all  $\omega \in \Omega$ . In addition  $\mathbb{P}(\omega) > 0 \forall \omega \in \Omega$ . Using the decomposition idea of Van-Slyke and Wets for stochastic linear programs [63], we separate the stochastic QP into a master problem and a set of scenario-based subproblems defined as

|        |   |     |               |   |
|--------|---|-----|---------------|---|
| master | $\begin{aligned} & \underset{x, \theta}{\text{minimize}} && \frac{1}{2} x^T P x + c^T x + \theta \\ & \text{subject to} && A x = b \\ & && \theta \geq \mathcal{Q}(x) \\ & && x \geq 0 \end{aligned}$ | and | sub $_\omega$ | $\begin{aligned} & \underset{y^\omega}{\text{minimize}} && \frac{1}{2} (y^\omega)^T D_\omega y^\omega + d_\omega^T y^\omega \\ & \text{subject to} && B_\omega y^\omega = b_\omega - A_\omega x \quad : v^\omega \\ & && y^\omega \geq 0, \quad : u^\omega \end{aligned}$ |
|--------|---|-----|---------------|---|

respectively, where  $u^\omega, v^\omega$  are the Lagrange multipliers and  $\mathcal{Q}(x) = \mathbb{E}[\mathcal{Q}(x; \omega)]$  and  $\mathcal{Q}(x; \omega)$  is the optimal value of (sub $_\omega$ ). The Lagrange dual of (sub $_\omega$ ) is given by

|                |   |
|----------------|---|
| dsub $_\omega$ | $\begin{aligned} & \underset{u^\omega, v^\omega}{\text{maximize}} && -\frac{1}{2} \begin{pmatrix} u^\omega \\ v^\omega \end{pmatrix}^T Q_\omega \begin{pmatrix} u^\omega \\ v^\omega \end{pmatrix} + q_\omega^T \begin{pmatrix} u^\omega \\ v^\omega \end{pmatrix} + r_\omega \\ & \text{subject to} && u^\omega \geq 0, \end{aligned}$ |
|----------------|---|

where

$$Q_\omega = \begin{pmatrix} D_\omega^{-1} & D_\omega^{-1} B_\omega^T \\ B_\omega D_\omega^{-1} & B_\omega D_\omega^{-1} B_\omega^T \end{pmatrix}, \quad q_\omega = \begin{pmatrix} D_\omega^{-1} d_\omega \\ B_\omega D_\omega^{-1} d_\omega - (A_\omega x - b_\omega) \end{pmatrix} \text{ and } r_\omega = -\frac{1}{2} d_\omega^T D_\omega^{-1} d_\omega.$$

Observe that  $q_\omega$  depends on  $x$ , but  $Q_\omega, r_\omega$  do not. If  $(\text{sub}_\omega)$  is feasible then strong duality holds and the optimal values of  $(\text{sub}_\omega)$  and  $(\text{dsub}_\omega)$  are equal. This allows us to define  $\mathcal{Q}(x; \omega)$  (the random recourse function) in terms of the optimal point of  $(\text{dsub}_\omega)$ ,  $(u^{\omega*}, v^{\omega*})$ , using sets  $K_2$  and  $K_2(\omega)$  (see (1) in section 2.1).

**Definition 4.1** *The random recourse function is defined as*

$$\mathcal{Q}(x; \omega) := -\frac{1}{2} \begin{pmatrix} u^{\omega*} \\ v^{\omega*} \end{pmatrix}^T Q_\omega \begin{pmatrix} u^{\omega*} \\ v^{\omega*} \end{pmatrix} + q_\omega^T \begin{pmatrix} u^{\omega*} \\ v^{\omega*} \end{pmatrix} + r_\omega \quad \forall x \in K_2(\omega).$$

*The recourse function is defined as the expectation of the random recourse function or*

$$\mathcal{Q}(x) := \mathbb{E}[\mathcal{Q}(x; \omega)] \quad \forall x \in K_2.$$

Moreover let  $\mathcal{S}$  denote the solution set of (StQP):

$$\mathcal{S} := \arg \min_x \left\{ \frac{1}{2} x^T P x + c^T x + \mathcal{Q}(x) : A x = b, x \geq 0 \right\}$$

We state the following result and refer the reader to [55, 57] for proof. The proof in [55, 57] uses the recourse function defined in terms of the solution the Dorn dual [22] of  $(\text{sub}_\omega)$ , given the strong duality between  $(\text{sub}_\omega)$  and its Dorn dual. Although we have used the solution of the Lagrange dual to define  $\mathcal{Q}(x)$ , the result from [55] continues to hold since strong duality applies in our case too.

**Proposition 4.2** *The random recourse function  $\mathcal{Q}(x; \omega)$  is finite for all  $x \in K_2(\omega)$ . Moreover  $\mathcal{Q}(x; \omega)$  is convex and continuous for all  $x \in K_2(\omega)$  and for all  $\omega \in \Omega$ , and the recourse function  $\mathcal{Q}(x)$  is convex and continuous for all  $x \in K_2$ .*

## 4.1 L-shaped method with inexact cuts

The L-shaped method creates an outer-approximation of the recourse function by using a series of tangent hyperplanes, called optimality cuts [8, 54]. The use of *inexact optimality cuts*, i.e. hyperplanes that do not intersect the recourse function but are a small distance away from the tangent hyperplane, to model the recourse function for stochastic linear programs was discussed separately by Au et al. [1] and Zakeri et al. [67]. This technique showed an improvement in the performance of the traditional L-shaped method [63] since it required the solution of the dual problems only to feasibility, as opposed to optimality.

The L-shaped method with inexact cuts was extended to solve stochastic convex programs by Shanbhag et al. [55, 57] and is the method we apply. Following is the definition of an  $\epsilon$ -inexact optimality cut in the context of (StQP).

**Definition 4.3** *Suppose for all  $\omega \in \Omega$ , given an  $\bar{x} \in K_2$ , the subproblem  $(\text{dsub}_\omega(\bar{x}))$  is solved to an optimality tolerance of  $\epsilon$ , with solution given by  $(v^{\omega*}, u^{\omega*})$ . Then an  $\epsilon$ -inexact cut is given by the linear function  $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}$ , such that*

$$\mathcal{H}(x) = G_I^T x + g_I, \quad \forall x \in K_2$$

where  $G_I$  and  $g_I$  are given by

$$G_I := \mathbb{E}_\omega [A_\omega^T v^{\omega*}]$$

$$g_I := \mathbb{E}_\omega \left[ b_\omega^T v^{\omega*} - \frac{1}{2} u^{\omega*T} D_\omega^{-1} u^{\omega*} - \frac{1}{2} v^{\omega*T} B_\omega D_\omega^{-1} B_\omega^T v^{\omega*} + v^{\omega*T} B_\omega D_\omega^{-1} d_\omega + u^{\omega*T} D_\omega^{-1} d_\omega - u^{\omega*T} D_\omega^{-1} B_\omega^T v^{\omega*} \right],$$

respectively, and the  $\epsilon$ -inexact cut satisfies  $\mathcal{H}(\bar{x}) + \epsilon \geq \mathcal{Q}(\bar{x})$ .

We shall use  $(G_I, g_I)$  to denote the inexact cut and  $(G, g)$  to denote the exact ( $\epsilon = 0$ ) cut. In the event that a choice of the first stage variable renders  $(\text{dsub}_\omega)$  unbounded, the algorithm adds a ‘‘feasibility cut’’, defined below.

**Definition 4.4** Given an  $\tilde{x}$  that solves (master), suppose the scenario dual ( $dsub_\omega$ ) is unbounded. If  $(\bar{v}^\omega, \bar{u}^\omega)$  is a direction of unboundedness for ( $dsub_\omega$ ), boundedness of ( $dsub_\omega$ ) is ensured by the following inequality

$$(b_\omega - A_\omega \tilde{x})^T \bar{v}^\omega + (\bar{v}^\omega)^T B_\omega D_\omega^{-1} d_\omega + (\bar{u}^\omega)^T D_\omega^{-1} d_\omega - \bar{u}^\omega D_\omega^{-1} B_\omega^T \bar{v}^\omega \leq 0.$$

A feasibility cut is thus defined as the linear  $\mathbb{R}^n \rightarrow \mathbb{R}$  function  $x \mapsto F_\omega^T x - f_\omega$  where

$$F_\omega := (A_\omega^T \bar{v}^\omega)^T \quad \text{and} \quad f_\omega := b_\omega^T \bar{v}^\omega + (\bar{v}^\omega)^T B_\omega D_\omega^{-1} d_\omega + (\bar{u}^\omega)^T D_\omega^{-1} d_\omega - \bar{u}^\omega D_\omega^{-1} B_\omega^T \bar{v}^\omega.$$

and it satisfies  $F_\omega^T \tilde{x} \geq f_\omega$ .

Let  $I^{\text{opt}}$  and  $I^{\text{fea}}$  be the set of optimality and feasibility cuts present at the current iterate of the algorithm. Note that  $I^{\text{fea}}$  contains elements identified by a tuple  $(j, \omega)$  consisting of an index and the scenario number. At any iteration, the stochastic QP master problem, together with the cuts is then given by

|       |   |
|-------|---|
| m-ils | minimize $\frac{1}{2}x^T P x + c^T x + \theta^I$<br>$x, \theta^I$   |
|       | $Ax = b,$<br>subject to $G_I^{jT} x + \theta^I \geq g_I^j, \quad j \in I^{\text{opt}}$<br>$F_{j,\omega}^T x \geq f_{j,\omega}, \quad j, \omega \in I^{\text{fea}}$<br>$x \geq 0.$ |

At any iteration  $x_k$  ( $dsub_\omega$ ) are solved for all  $\omega \in \Omega$  to  $\epsilon_k$ -inexactness to obtain the cut  $(G_I, g_I)$ . This constraint is added to (m-ils) which is then solved to get  $x_{k+1}$ . If any ( $dsub_\omega$ ) is unbounded, a feasibility cut is added to (m-ils). If

$$(G_I^k)^T x_k + g_i^k > \theta_k^I,$$

then  $\epsilon_{k+1} = \epsilon_k/u$ , and the upper and lower bounds are not updated, implying that the optimality cut is recomputed with a lower  $\epsilon_k$  [55]. The L-shaped method with inexact cuts (Algorithm 2) proceeds as given below [55].

---

**Algorithm 2:** L-shaped method with inexact cuts

---

```

o initialization  $k = 1, U_k^I = \infty, L_k^I = -\infty;$ 
  choose  $\epsilon_1, \tau, u > 1;$ 
  while  $|U_k^I - L_k^I| > \tau$  do
1   Solve (m-ils) to get  $(x_k, \theta_k);$ 
2   Solve ( $dsub_\omega$ ) to tolerance  $\epsilon_k$  to obtain  $(u^{\omega*}, v^{\omega*})$  for all  $\omega \in \Omega;$ 
3   if ( $dsub_\omega$ ) is bounded  $\forall \omega$  then
    | Construct  $(G_I^k, g_I^k)$  as per Definition 4.3;
    else
    | Add feasibility cut and go to step 1;
    end
4   if If  $(G_I^k)^T x_k + g_i^k \leq \theta_k^I$  then
    | Update upper bound  $U_k^I$ , lower bound  $L_k^I$  and add optimality cut  $(G_I^k, g_I^k)$  to (m-ils)
    else
    |  $U_{k+1}^I = U_k^I, L_{k+1}^I = L_k^I$  and  $\epsilon_{k+1} = \epsilon_k/u;$ 
    end
5    $k = k + 1$ 
  end
end

```

---

In the above algorithm,  $u$  is called the *inexactness update*. It controls the rate at which optimality cuts are tightened.  $U_k^I, L_k^I$  are upper and lower bounds, defined as

$$L_k^I \equiv c^T x_k + \frac{1}{2}x_k^T P x_k + \theta_k \quad \text{and} \quad U_k^I \equiv \min\{U_{k-1}^I, c^T x_k + \frac{1}{2}x_k^T P x_k + Q(x_k)\}, \text{ respectively.}$$

Notice that  $\{L_k^I\}$  is a monotonically increasing sequence while  $\{U_k^I\}$  is a monotonically decreasing sequence. It can be shown that as  $k \rightarrow \infty$ ,  $|U_k^I - L_k^I| \rightarrow 0$  and hence  $x_k \rightarrow x^*$ , the solution of (StQP) and  $\theta_k$  approaches  $Q(x^*)$ . The convergence theory for the inexact cut L-shaped method can be found in [55, 57].

## 4.2 Trust-region L-shaped method for stochastic QPs

In [42], a trust-region variant of the L-shaped method is employed for solving stochastic linear programs. Specifically, such a scheme requires appending a trust-region bound to the master problem to regulate the step taken. In the past, trust-region approaches have been explored by Kiwiel [38] (bundle methods) and by Hiriart-Urruty and Lemaréchal [32]. The trust-regions used by Linderoth and Wright (hereafter referred to as LW) differ from these in the way that they use box constraints, keeping the new set of constraints also linear. Furthermore, they assume that their stochastic linear program has complete recourse [42].

As noted in the previous section, assuming relatively complete recourse for (SNLP) does not guarantee us complete recourse for (StQP<sub>k</sub>). Here we present a modified version of their algorithm for stochastic QPs that does not assume complete recourse. Our method has the following distinctions from the method of LW:

- It can address stochastic QPs.
- The convergence of the method is under the assumption of relatively complete recourse (as opposed to complete recourse in the case of LW) and uses feasibility cuts.

Consider a generic stochastic quadratic program (StQP). If we assume that all (sub<sub>ω</sub>) are feasible, solving (StQP) is equivalent to solving the following problem.

|        |   |
|--------|---|
| master | $\begin{aligned} & \underset{x, \theta}{\text{minimize}} && \frac{1}{2}x^T Px + c^T x + \theta \\ & && Ax = b \\ & \text{subject to} && \theta \geq \mathbb{E} \left[ \frac{1}{2} \bar{y}^{\omega T} D_{\omega} \bar{y}^{\omega} + d_{\omega}^T \bar{y}^{\omega} \right] \\ & && x \geq 0. \end{aligned}$ |
|--------|---|

This is an exact-cut analogue of (m-ils) with  $\bar{y}^{\omega}$  as the solution to (sub<sub>ω</sub>). The right hand side of the second constraint forms the recourse function defined by Definition 4.1 and Theorem 4.2. One of our key modifications to LW's method is that in the event that any (dsub<sub>ω</sub>) turns out to be unbounded, and  $\bar{y}^{\omega}$  does not exist, we append the master problem with a feasibility cut as in Definition 4.4. It is hence possible to append the feasibility cut to the constraint  $Ax = b$  in the form of an equality constraint with slack variables to form a new set of constraints  $\bar{A}x - s = \bar{b}$ . In Lemma 4.10, we show that the total number of feasibility cuts that can be added in the algorithm is finite. For simplicity, we assume that all the possible feasibility cuts are added and included within the constraint  $\bar{A}x - s = \bar{b}$ . Before proceeding further, we introduce some notation. Let

$$\widehat{Q}(x) \equiv \frac{1}{2}x^T Px + c^T x + \mathcal{Q}(x),$$

where  $\mathcal{Q}(x)$  is defined in Definition 4.1. We define two kind of iterates – *major* and *minor*, of which the major fall into two *types* to be defined later. Major iterates are denoted with a single subscript as “ $x_k$ ”  $k = 0, 1, 2, \dots$ , regardless of their type. Corresponding to each major iterate there could be a series of minor iterates, denoted by “ $x_{k,l}$ ”  $l = 0, 1, 2, \dots$ . All other entities are subscripted by the pair  $(k, l)$  where  $k$  is the major iterate index and  $l$  is the corresponding minor iterate index. The sets  $I^{\text{opt}}$  and  $I^{\text{fea}}$  represent the set of optimality and feasibility cuts respectively while  $\mathcal{S}$  is the set of all solutions of (StQP). Let  $\Delta_{k,l}$  denote the trust-region imposed at the  $k^{\text{th}}$  major iteration and  $l^{\text{th}}$  minor iterate. The first-stage problem at any iteration, together with the box-shaped trust region is given by (m-tr<sub>k,l</sub>):

|                     |  |
|---------------------|--|
| m-tr <sub>k,l</sub> | $\begin{aligned} & \underset{x, \theta}{\text{minimize}} && \frac{1}{2}x^T Px + c^T x + \theta \\ & && \bar{A}x - s = \bar{b} \\ & \text{subject to} && \theta \geq -G_j^T x + g_j \quad \forall j \in I^{\text{opt}} \\ & && -\Delta_{k,l} e \leq x - x_k \leq \Delta_{k,l} e \\ & && x, s \geq 0. \end{aligned}$ |
|---------------------|--|

An alternative to using the  $\ell_{\infty}$  trust regions would be to use  $\ell_2$  trust regions by adding a constraint to (m-tr<sub>k,l</sub>) like

$$\|x - x_k\|_2^2 \leq \Delta_{k,0}.$$

While such a quadratically constrained QP would not be solvable by standard QP solvers, this is more in line with conventional trust-region subproblems [46]. Another alternative is to impose the trust region implicitly by adding a term  $\rho\|x - x_k\|^2$  as a penalty to the objective of (m-tr<sub>k</sub>). This results in a problem similar to the one in the regularized decomposition method [54]. We leave the question of which trust-region framework may prove advantageous as future research and focus primarily on extending the approach suggested in [42]. Here onwards  $\|\cdot\|$  denotes the  $\ell_\infty$  norm.

Let  $(G_{k,l}, g_{k,l})$  be an optimality cut obtained from exact solution of all subproblems (dsub<sub>ω</sub>). The *model function*, formed by the set of optimality cuts that approximate  $\widehat{Q}(x)$  at iteration  $k, l$ , is denoted as  $m_{k,l}(x)$ . It is defined over  $\mathbb{R}^n$  as follows:

$$\begin{aligned} m_{k,l}(x) &\equiv \frac{1}{2}x^T Px + c^T x + \max_{(k,l) \in I^{\text{opt}}} \{-G_{k,l}^T x + g_{k,l}\} \\ &= \frac{1}{2}x^T Px + c^T x + \inf_{\theta} \{\theta : \theta \geq -G_{k,l}^T x + g_{k,l} \forall (k,l) \in I^{\text{opt}}\}. \end{aligned} \quad (15)$$

Let  $\bar{x}$  be the solution of (m-tr<sub>k-1,l</sub>) or

$$\bar{x} = \arg \min_x \{m_{k-1,l}(x) : \bar{A}x - s = \bar{b}, -\Delta_{k-1,l}e \leq x - x_{k-1} \leq \Delta_{k-1,l}e, x, s \geq 0\}. \quad (16)$$

If the *acceptance test* stated below is satisfied, we set  $x_k$ , the  $k^{\text{th}}$  major iterate of type 1, to be equal to  $\bar{x}$ . If  $\bar{x}$  is not a major iterate, it is taken as *minor iterate*  $x_{k-1,l}$ . Note that a major iterate  $x_k$  is *not* the 0<sup>th</sup> minor iterate  $x_{k,0}$ . In fact,  $x_k$  is a solution to the model function given by  $m_{k-1,r}$  where  $r$  represents the terminating iterate of the previous subsequence. In the next subsection, we elaborate on how the model function gets updated.

#### 4.2.1 Model update procedure

If for any first-stage solution  $x_{k,l}$ , the dual problem (dsub<sub>ω</sub>) is unbounded for some  $\omega \in \Omega$ , we add a feasibility cut to (m-tr<sub>k,l</sub>) and form (m-tr<sub>k,l+1</sub>). The sequence of minor iterates is then taken to have terminated at  $x_{k,l-1}$ . Denote the solution of (m-tr<sub>k,l+1</sub>) as  $x_{k,l}^1$ . If again for  $x_{k,l}^1$  any (dsub<sub>ω</sub>) is unbounded, we add another feasibility cut and generate (m-tr<sub>k,l+2</sub>) and iterate  $x_{k,l}^2$ . We continue to do this until for some  $i$ , all (dsub<sub>ω</sub>) corresponding to  $x_{k,l}^i$  are bounded. This is taken then we set  $x_{k+1} = x_{k,l}^i$ , as the major iterate of type 2.

**Definition 4.5** Let  $x_{k,l}$  be the current iterate and suppose for some  $\omega \in \Omega$  (dsub<sub>ω</sub>) parameterized by  $x_{k,l}$  is unbounded. Let  $x_{k,l}^1, x_{k,l}^2, \dots$  denote the solutions of subsequent problems formed by the addition of feasibility cuts. Then  $x_{k+1} = x_{k,l}^{\bar{i}}$ , where  $\bar{i}$  is the smallest  $i$  such that all (dsub<sub>ω</sub>) corresponding to  $x_{k,l}^i$  are feasible.  $x_{k+1}$  is called a major iterate of type 2.

Since the number of feasibility cuts that can be added during the algorithm is finite (cf. Lemma 4.10), a major iterate of type 2 is a well defined entity. We note that LW have assumed ([42], Assumption 1) that their problem has *complete* recourse and hence do not require feasibility cuts. The following procedure, similar to that in LW [42] with modifications made to accommodate feasibility cuts, is used to obtain  $m_{k,l+1}$  from  $m_{k,l}$ :

1. If any scenario dual (dsub<sub>ω</sub>) is unbounded, then a feasibility cut is added to (m-tr<sub>k,l</sub>) and no cuts are deleted. The model function  $m_{k,l}(x)$  remains unchanged as no optimality cuts are added.
2. If all (dsub<sub>ω</sub>) are bounded,
  - (a) all cuts generated at  $x_k$  are retained.
  - (b) all cuts active at the solution of (m-tr<sub>k,l</sub>) are retained.
  - (c) for a given  $\eta \in [\xi, 1)$ , all cuts generated at previous minor iterates,  $\hat{l}$  during the same major iteration, that satisfy

$$m_{k,l}(x_k) - m_{k,l}(x_{k,l}) > \eta \left[ \widehat{Q}(x_k) - m_{k,i}(x_{k,i}) \right] \quad (17)$$

are retained.

- (d) if a cut does not satisfy (a), (b) or (c) and has been inactive for the past 100 solutions of (m-tr<sub>k,l</sub>), then it is deleted.

(e) all optimality cuts arising from the solutions of (dsub $_{\omega}$ ) at  $x_{k,l}$  are added to  $m_{k,l}(x)$  to get  $m_{k,l+1}(x)$ .

The algorithm needs to be supplied an initial model function  $m_{0,0}$  which satisfies the following assumptions.

**Assumption 4.6** *At the starting point  $x_0$  we have  $m_0(x_0) = \widehat{Q}(x_0)$  where  $m_0(x)$  is a piecewise linear underestimate of  $\widehat{Q}(x)$*

Next, we define the acceptance test for the qualification of  $x_{k,l}$  as a major iterate.

#### 4.2.2 Acceptance test

For some fraction  $\xi \in (0, 1/2)$ , the minor iterate  $x_{k,l}$  is accepted as  $x_{k+1}$  if and only if

$$\widehat{Q}(x_{k,l}) \leq \widehat{Q}(x_k) - \xi (m_{k,l}(x_k) - m_{k,l}(x_{k,l})). \quad (18)$$

The trust-region approach limits the step size and thus allows us to solve (m-tr $_{k,l}$ ) with a limited number of optimality cuts. As a result, cuts that are inconsequential for the solution of  $x_{k,l}$  can be deleted. Which cuts to delete is a matter of careful choice since incorrect deletion can make us re-evaluate cuts that were deleted prematurely. For iterates that do not satisfy the acceptance test, LW provides a procedure to update the trust region which we adopt and describe next.

#### 4.2.3 Updating the trust-region $\Delta$

Let  $\rho$  be defined as

$$\rho := \min(1, \Delta_{k,l}) \left( \frac{\widehat{Q}(x_{k,l}) - \widehat{Q}(x_k)}{\widehat{Q}(x_k) - m_{k,l}(x_{k,l})} \right). \quad (19)$$

Then the update of the trust region is carried out as follows:

(1.) If  $[\rho > 0$  for 3 iterations with same  $\Delta_{k,l}$  and  $\rho \in (1, 3]$  or  $[\rho > 3]$ , then the updated trust region is given by

$$\Delta_{k,l+1} = \frac{1}{\min(\rho, 4)} \Delta_{k,l}.$$

(2.) When the new major iterate  $x_{k+1}$  is obtained, there is a case for increasing the trust region. If

$$\widehat{Q}(x_{k,l}) \leq \widehat{Q}(x_k) - 0.5 \left( \widehat{Q}(x_k) - m_{k,l}(x_{k,l}) \right) \text{ and } \|x_k - x\|_{\infty} = \Delta_{k,l},$$

then  $\Delta_{k+1,0} = \min(\Delta_{\text{hi}}, 2\Delta_{k,l})$  where  $\Delta_{\text{hi}}$  is a pre-specified upper bound on the trust region.

(3.) If (1.) or (2.) are not satisfied, then  $\Delta_{k,l}$  is left unchanged.

Finally, given  $\epsilon_{\text{tol}} > 0$  the algorithm terminates if

$$\widehat{Q}(x_k) - m_{k,l}(x_{k,l}) \leq \epsilon_{\text{tol}} \left( 1 + |\widehat{Q}(x_k)| \right). \quad (20)$$

We now state the complete L-shaped trust-region method (Algorithm 3) for solving stochastic QPs.

---

**Algorithm 3:** Trust-Region L-shaped method

---

```
0 initialization  $k = 1, l = 0$ ;  
  choose  $\xi \in (0, 1/2)$ , initial point  $x_{0,0}$ , initial model  $m_{0,0}$  and initial trust region  $\Delta_{0,0} \in (0, \Delta_{hi}]$ ,  $\epsilon_{tol} < 1$ ;  
  while  $\widehat{Q}(x_k) - m_{k,l}(x_{k,l}) > \epsilon_{tol}(1 + |\widehat{Q}(x_k)|)$  do  
1   | Solve (16) to get  $x_{k,l}$ ;  
2   | Solve  $(\text{dsub}_\omega)$  accurately to obtain solution points  $(u_\omega^*, v_\omega^*)$  for all  $\omega \in \Omega$ ;  
3   | if  $(\text{dsub}_\omega)$  is bounded  $\forall \omega$  then  
   |   | Construct  $(G_{k,l}, g_{k,l})$ ;  
   |   | if feasibility cut was added in the previous iteration then  
   |   |   |  $x_{k+1} = x_{k,l}$ ;  
   |   |   | Obtain new model  $m_{k+1,0}$  as per section 4.2.1;  
   |   |   |  $k = k + 1$ ;  
   |   | end  
   | else  
   |   | Add feasibility cut and go to step 1;  
   | end  
4   | if acceptance test (18) is satisfied then  
   |   | Set  $x_{k+1} = x_{k,l}$ ;  
   |   | Obtain the new model  $m_{k+1,0}$  as per section 4.2.1;  
   |   | Update  $\Delta_{k+1,0}$  as per section 4.2.3;  
5   |   |  $k = k + 1$ ;  
   | else  
   |   | Update the model to get  $m_{k,l+1}$  as per section 4.2.1;  
6   |   | Update  $\Delta_{k,l+1}$  as per section 4.2.3;  
7   |   |  $l = l + 1$ ;  
   | end  
end
```

---

#### 4.2.4 Convergence of the trust-region L-shaped method for stochastic QPs

An outline of the main convergence results follows. We begin by noting some results that will be used in the proofs to follow. By definition of optimality cuts

$$m_{k,l}(x) \leq \widehat{Q}(x), \quad \forall x. \quad (21)$$

Recall from section 4.2.1 that all cuts at  $x_k$  are retained. So we have

$$m_{k,l}(x_k) = \widehat{Q}(x_k), \quad \forall k, l. \quad (22)$$

Recall also that  $m_{k,l}(x)$  is a piecewise linear underestimate of  $\widehat{Q}(x)$ . From (16), since  $x_{k,l}$  minimizes  $m_{k,l}(\cdot)$ ,  $m_{k,l}(x_{k,l}) = \widehat{Q}(x_{k,l})$  if and only if  $x_{k,l} \in \mathcal{S}$ . In fact, since the cut added at  $x_{k,l}$  to form  $m_{k,l+1}(x)$  is exact  $m_{k,l+1}(x_{k,l}) = \widehat{Q}(x_{k,l})$ .

Since this theory follows portions of that in LW closely, we distinguish our results from theirs when the proofs either differ or have been extended significantly. The convergence proof relies on showing that for  $\epsilon_{tol} > 0$  there cannot exist an infinite sequence of iterates that do not satisfy (20). For  $\epsilon_{tol} = 0$  we show that the major iterates approach  $\mathcal{S}$ . The first two results establish that the model function minimum increases with every minor iterate and provide a bound for such an increase. Next, we prove a boundedness result for the trust-region radius as well as the finite termination of the subsequence of minor iterates originating from a major iterate. The convergence theorems will be proved using these results.

**Lemma 4.7** *If a minor iterate  $x_{k,l+1}$  does not satisfy the acceptance test (18), we have*

$$m_{k,l}(x_{k,l}) \leq m_{k,l+1}(x_{k,l+1}).$$

**Proof :** See Lemma 1 in LW [42].

In Lemma 4.8, we provide a lower bound on the difference between the optimal value of the model function and the value of the model function at the last major iterate, namely  $m_{k,l}(x_k) - m_{k,l}(x_{k,l})$ .

**Lemma 4.8** Consider an iterate  $x_{k,l}$  derived at a major iteration  $k$  and minor iterate  $l$  where  $k, l \geq 0$ . Then the following bound on the change in the model function holds:

$$m_{k,l}(x_k) - m_{k,l}(x_{k,l}) = \widehat{Q}(x_k) - m_{k,l}(x_{k,l}) \geq \min\left(\frac{\Delta_{k,l}}{\|x_k - P(x_k)\|}, 1\right) (\widehat{Q}(x_k) - \widehat{Q}^*)$$

where  $P(x_k)$  is the projection of  $x_k$  on  $\mathcal{S}$ .

**Proof :** The equality in the claim follows from (22). We show that the inequality holds. Since  $x_{k,l}$  is the solution of (m-tr $_{k,l}$ ), for all  $x$  that are feasible for (m-tr $_{k,l}$ ), we have

$$m_{k,l}(x_{k,l}) \leq m_{k,l}(x). \quad (23)$$

In particular, this holds for

$$\hat{x} = x_k + \beta_{k,l}(P(x_k) - x_k) \quad \text{where } \beta_{k,l} = \min\left(\frac{\Delta_{k,l}}{\|P(x_k) - x_k\|}, 1\right).$$

To see this, recall that (m-tr $_{k,0}$ ) and (m-tr $_{k,l}$ ) differ only in the optimality cuts. Hence  $x_k$  is feasible for (m-tr $_{k,l}$ ).  $P(x_k) \in \mathcal{S}$ , so it also is feasible for (m-tr $_{k,l}$ ). Since the constraints of (m-tr $_{k,l}$ ) are linear,  $\hat{x}$  is feasible.

Combining with (23), it follows that

$$\begin{aligned} m_{k,l}(x_{k,l}) &\leq m_{k,l}(x_k + \beta_k(P(x_k) - x_k)) \\ &\leq \widehat{Q}((1 - \beta_k)x_k + \beta_k P(x_k)), \\ &\leq (1 - \beta_k)\widehat{Q}(x_k) + \beta_k\widehat{Q}(P(x_k)) = \widehat{Q}(x_k) + \beta_k(\widehat{Q}^* - \widehat{Q}(x_k)), \end{aligned}$$

where the second inequality follows from (21), the third inequality follows since  $\beta_{k,l} \leq 1$  and from convexity and  $\widehat{Q}(P(x_k)) = \widehat{Q}^*$ . Rearranging, we obtain the required result. ■

Next, we prove that the trust-region is bounded away from 0 for all iterates  $x_{k,l} \notin \mathcal{S}$ .

**Lemma 4.9** Suppose  $E_k, F_k$  and  $\beta$  are defined as

$$\begin{aligned} E_k &\equiv \min_{\bar{k}=0,1,\dots,k} \|x_{\bar{k}} - P(x_{\bar{k}})\|_\infty, & F_k &\equiv \min_{\bar{k}=0,1,\dots,k, x_{\bar{k}} \notin \mathcal{S}} \frac{\widehat{Q}(x_{\bar{k}}) - \widehat{Q}^*}{\|x_{\bar{k}} - P(x_{\bar{k}})\|_\infty}, \\ \beta &= \sup \|g\|, \quad g \in \partial\widehat{Q}(x), \quad x \text{ feasible for (m-tr}_{k,l}) \end{aligned}$$

Suppose there exists  $\tilde{k}$  such that  $\min(E_{\tilde{k}}, F_{\tilde{k}}/\beta) < \Delta_{hi}$ . Then for all  $k > \tilde{k}$  and for all  $l$ , the trust-region  $\Delta_{k,l}$  satisfies

$$\Delta_{k,l} \geq (1/4) \min(E_k, F_k/\beta).$$

**Proof :** See Lemma 3 in LW [42]. ■

Lemma 4.10 is a result distinct from that proved by LW. It shows that the algorithm can have only finitely many feasibility cuts.

**Lemma 4.10** The subsequence of major iterates of type 2 terminates finitely.

**Proof :** Recall that a feasibility cut is added when one of the dual problem (dsub $_\omega$ ) is unbounded. The region of unboundedness of each (dsub $_\omega$ ) is spanned by the set of extreme rays. Since the dimensionality of (dsub $_\omega$ ) is finite, the number of extreme rays is clearly finite as well. Each feasibility cut corresponds to one and only one extreme ray. Since  $|\Omega| < \infty$ , the total number of feasibility cuts added by Algorithm 3 is finite. It follows that the subsequence of major iterates of type 2 cannot be infinite. ■

The next set of results require specifying the tolerance  $\epsilon_{tol}$  in the algorithm for which we consider two cases:

- (1)  $\epsilon_{tol} = 0$ : Here, we show that the sequence of major iterates either terminates finitely in a point in  $\mathcal{S}$  or converges to a point in  $\mathcal{S}$ .
- (2)  $\epsilon_{tol} > 0$ : In this instance, we show that the algorithm terminates to a point satisfying the termination criterion (20).

Lemma 4.11 proves that the sequence of minor iterates either produces an iterate that satisfies the acceptance test or produces a minor iterate that results in a strict reduction in  $m_{k,l}(x_k) - m_{k,l}(x_{k,l})$ .

**Lemma 4.11** *Let  $\epsilon_{tol} \geq 0$  and  $\xi$  be given where  $\xi$  specifies the relative improvement in the acceptance test in (18). Let  $\eta$  be the parameter used for specifying the deletion of cuts as per (17). For some major iterate  $k$  of type 1 and minor iterate  $l_1$ , suppose  $x_{k,l_1}$  fails the acceptance test given by (18). Then the sequence of minor iterates generates an  $x_{k,l_2}$  that satisfies (18) or there is an index  $l_2$  that satisfies*

$$\widehat{Q}(x_k) - m_{k,l_2}(x_{k,l_2}) \leq \eta \left[ \widehat{Q}(x_k) - m_{k,l_1}(x_{k,l_1}) \right]. \quad (24)$$

**Proof :** We consider two cases:

- (a)  $\epsilon_{tol} = 0$ : See Lemma 4 in LW [42] .
- (b)  $\epsilon_{tol} > 0$ : This result holds by observing that the relation

$$\zeta := \frac{\eta - \xi}{\beta} [\widehat{Q}(x_k) - m_{k,l}(x_{k,l})] > 0,$$

in expression (52) of LW [42] holds even when  $\epsilon_{tol} > 0$ . ■

For the final convergence statement, we first consider the case  $\epsilon_{tol} = 0$ . Combining the implications of Lemma 4.11 with Lemma 4.8 we can show that, if  $x_k \notin \mathcal{S}$ , then it produces a minor iterate  $x_{k,l}$  that satisfies the acceptance test. This is the result of Theorem 4.12.

**Theorem 4.12** *Suppose  $\epsilon_{tol} = 0$ . Then the following hold:*

- (1) *If  $x_k \notin \mathcal{S}$  then there exists an  $l \geq 0$  such that  $x_{k,l}$  satisfies (18)*
- (2) *If  $x_k \in \mathcal{S}$  then either the algorithm terminates finitely or  $\left( \widehat{Q}(x_k) - m_{k,l}(x_{k,l}) \right)$  decreases monotonically to zero.*

**Proof :** See Theorem 1 in LW [42]. ■

Theorem 4.13 shows that the sequence of major iterates approaches a solution.

**Theorem 4.13** *Suppose  $\epsilon_{tol} = 0$ . The sequence of major iterations is either finite terminating at some  $x_k \in \mathcal{S}$  or infinite with  $\|x_k - P(x_k)\|_\infty \rightarrow 0$ .*

**Proof :** See Theorem 2 in LW [42]. ■

We conclude our convergence results with a finite termination result for the case where  $\epsilon_{tol} > 0$ . This result was claimed in LW to be easy to prove and skipped. But since our algorithm differs from LW's to a certain extent we prove this result here for completeness.

**Theorem 4.14** *For  $\epsilon_{tol} > 0$ , the algorithm terminates finitely at a solution satisfying (20)*

**Proof :** Assume in accordance with Lemma 4.10 that all feasibility cuts have been added. From Lemma 4.11, for some minor iterate  $l$  that fails the acceptance test (18), the sequence of minor iterates either yields an iterate that satisfies test (18) or an iterate satisfying (24). Let  $x_{k,l_1}$  be an iterate that does not satisfy test (18). Two cases arise:

- (a) Suppose that the sequence of minor iterates emerging from major iterate  $k$  never satisfies the acceptance test. Then for  $l_2 > l_3 > \dots > l_r$  we have

$$\widehat{Q}(x_k) - m_{k,l_r}(x_{k,l_r}) \leq \eta \left( \widehat{Q}(x_k) - m_{k,l_{r-1}}(x_{k,l_{r-1}}) \right) \leq \dots \leq \eta^{r-1} \left( \widehat{Q}(x_k) - m_{k,l_1}(x_{k,l_1}) \right).$$

This indicates that for sufficiently large, but finite  $r$ , the term  $\widehat{Q}(x_k) - m_{k,l_r}(x_{k,l_r})$  can be reduced enough to satisfy (20), implying the finite termination of the algorithm.

- (b) Suppose the test (18) is satisfied for some  $l_1$ . This results in a new major iterate  $x_{k+1}$ . If for any  $\bar{k} \geq k$  the sequence of minor iterates  $x_{\bar{k},l}, l \geq 0$  behave as in case (a), then convergence follows from the argument in (a). So suppose that for all  $\bar{k} \geq k$  the minor iterate sequence  $x_{\bar{k},l}, l \geq 0$  does not behave as in case (a). This means that for all  $\bar{k} \geq k$  the acceptance test is satisfied for some  $x_{\bar{k},l}, l \geq 0$ . For major iterates that have not terminated we have

$$\widehat{Q}(x_{\bar{k}}) - m_{\bar{k},l}(x_{\bar{k},l}) > \epsilon_{tol}(1 + |\widehat{Q}(x_{\bar{k}})|).$$

Hence every major iterate causes a reduction in  $\widehat{Q}$  of at least  $\xi\epsilon_{tol}$  as seen below

$$\widehat{Q}(x_{\bar{k}}) - \widehat{Q}(x_{\bar{k},l}) \geq \xi \left( \widehat{Q}(x_{\bar{k}}) - m_{\bar{k},l}(x_{\bar{k},l}) \right) \geq \xi\epsilon_{tol}.$$

Since a solution exists to the minimization problem (StQP),  $\widehat{Q}(x_{\bar{k}})$  is bounded below. This further implies that the minimum of (StQP) is achieved in finitely many major iterations and (20) has been satisfied. ■

### 4.3 Recovering the multipliers

We begin by recalling that the multipliers of the original problem (SNLP) are updated using a standard SQP multiplier update (see section 3.3). This update requires the multipliers of the QP approximation, as specified by (StQP). Note that in comparison with (StQP), the master problem has an extra variable ( $\theta$ ), has some new constraints (arising from the optimality/feasibility cuts) while having lost some constraints (these correspond to  $By^\omega = b_\omega - A_\omega x$  which now constrain the feasible region of the second-stage scenario-specific subproblem). Furthermore, the objective of the master problem has been modified. Thus, the multiplier set of the master does not immediately give us the multipliers of (StQP) since these problems, based on the above discussion, vary significantly. In particular the multipliers corresponding to the constraint  $Ax = b$  and  $x \geq 0$  in (StQP) may be different from the multipliers corresponding to these constraints in the master. This section is dedicated to our procedure for recovering the “true multipliers”, i.e. multipliers corresponding to these constraints in (StQP).

We observe below, that a subset of all the multipliers of (StQP) can be obtained directly through the cutting plane method. As a consequence, for the dual problem of (StQP), which has all multipliers as its variables, we know a portion of the optimal solution. Thus the dual problem can be reduced to a problem of smaller dimension, whose solution is precisely the true multipliers we wish to recover. It is primarily this observation that leads to a tractable means of obtaining the required multipliers. Importantly, this dual problem is a quadratic program of the size of the first-stage constraints and can be solved readily.

To explain this elaborately, let us compare the KKT conditions of the QP approximation (StQP), the master problem and the subproblems. All variables below with an asterisk as superscript pertain to the KKT point of (StQP). The conditions for optimality of (StQP) are given as

$$\begin{aligned} Px^* + c - A^T \lambda^{x*} - \sum_{\omega} A_{\omega}^T \lambda^{\omega*} - \mu^{x*} &= 0 \\ p_{\omega} [D_{\omega} y^{\omega*} + d_{\omega}] - B_{\omega}^T \lambda^{\omega*} - \mu^{\omega*} &= 0 \quad \forall \omega \in \Omega \\ Ax^* &= b \\ B_{\omega} y^{\omega*} &= b_{\omega} - A_{\omega} x^* \quad \forall \omega \in \Omega \\ 0 \leq \mu^{x*} \perp x^* &\geq 0 \\ 0 \leq \mu^{\omega*} \perp y^{\omega*} &\geq 0 \quad \forall \omega \in \Omega. \end{aligned}$$

The solutions of the subproblems ( $dsub_{\omega}$ ) are denoted by a superscript  $s$ , and at a given  $x$  they satisfy their respective KKT conditions

$$\begin{aligned} p_{\omega} [D_{\omega} y^{\omega s} + d_{\omega}] - B_{\omega}^T \lambda^{\omega s} - \mu^{\omega s} &= 0 \\ B_{\omega} y^{\omega s} &= b_{\omega} - A_{\omega} x \\ 0 \leq \mu^{\omega s} \perp y^{\omega s} &\geq 0, \quad \forall \omega \in \Omega. \end{aligned}$$

The master problem solution (for the inexact L-shaped method) is denoted by a superscript  $m$ . Assume for simplicity that there are only optimality cuts. If  $\mathbf{1}$  represents the column of ones, then the KKT point of the master satisfies

$$\begin{aligned} Px^m + c - \sum_j G^j \gamma_j^m - A^T \lambda^{x^m} &= 0 \\ \mathbf{1} - \sum_j \gamma_j^m &= 0 \\ Ax^m &= b \\ 0 \leq \gamma_j^m \perp G^{jT} x^m + \theta^m - g^j &\geq 0 \\ 0 \leq \mu^m \perp x &\geq 0. \end{aligned}$$

At the optimal solution generated by the L-shaped method, we have  $x^m = x^*$  (this follows from the convergence theory of the previous sections). Yet, (StQP) and the master problem differ significantly in terms of the constraints, the objectives and the set of decision variables. Consequently,  $(\lambda^{x^*}, \mu^{x^*})$ , the multipliers of the (StQP), are not necessarily equal to  $(\lambda^{x^m}, \mu^{x^m})$ , the corresponding multipliers of the master. This is further reinforced by observing that the multiplier  $\lambda^{x^m}$  satisfies a different set of equations from  $\lambda^{x^*}$ . Note, however, that since the linear independence constraint qualification holds for (StQP) (in this case requiring that  $B_\omega$  has full row rank for each  $\omega \in \Omega$ ), when  $y^{\omega^s} = y^{\omega^*}$ , it follows that  $\lambda^{\omega^s} = \lambda^{\omega^*}$  and  $\mu^{\omega^s} = \mu^{\omega^*}$ ,  $\forall \omega \in \Omega$ . In effect, the Benders procedure provides us with a subset of the multipliers to (StQP). Indeed, these observations are supported by our numerical experiments. Next, we describe a procedure for recovering the remaining multipliers, namely  $(\lambda^{x^*}, \mu^{x^*})$ .

Let  $z, Q, q, M, h, \lambda$  and  $\mu$  be defined as

$$\begin{aligned} z &:= \begin{pmatrix} x \\ y^1 \\ \vdots \\ y^{|\Omega|} \end{pmatrix} \quad Q := \begin{pmatrix} P & 0 & \dots & \dots & 0 \\ 0 & p_1 D_1 & \ddots & & 0 \\ 0 & 0 & p_2 D_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & p_{|\Omega|} D_{|\Omega|} \end{pmatrix} \quad q := \begin{pmatrix} c^T \\ p_1 d_1 \\ \vdots \\ p_{|\Omega|} d_{|\Omega|} \end{pmatrix} \quad M := \begin{pmatrix} A & 0 & \dots & 0 \\ A_1 & B_1 & \dots & 0 \\ \vdots & & \ddots & 0 \\ A_{|\Omega|} & 0 & \dots & B_{|\Omega|} \end{pmatrix} \\ h &:= \begin{pmatrix} b \\ b_1 \\ \vdots \\ b_{|\Omega|} \end{pmatrix} \quad \lambda := \begin{pmatrix} \lambda^x \\ \lambda^1 \\ \vdots \\ \lambda^{|\Omega|} \end{pmatrix} \quad \mu := \begin{pmatrix} \mu^x \\ \mu^1 \\ \vdots \\ \mu^{|\Omega|} \end{pmatrix}, \text{ respectively.} \end{aligned}$$

Using this, write (StQP) as

|      |   |
|------|---|
| StQP | $\begin{aligned} &\underset{z}{\text{minimize}} && \frac{1}{2} z^T Q z + q^T z \\ &\text{subject to} && Mz = h \quad : \lambda \\ &&& z \geq 0 \quad : \mu \end{aligned}$ |
|------|---|

Let

$$N = \begin{pmatrix} MQ^{-1}M^T & Q^{-1}M^T \\ MQ^{-1} & Q^{-1} \end{pmatrix} \quad \eta = \begin{pmatrix} MQ^{-1}q + h \\ Q^{-1}q \end{pmatrix} \quad \text{and} \quad \nu = \begin{pmatrix} \lambda \\ \mu \end{pmatrix}.$$

The Lagrange dual of (StQP) is

|          |  |
|----------|--|
| StQPdual | $\begin{aligned} &\underset{\lambda, \mu}{\text{maximize}} && -\frac{1}{2} \nu^T N \nu + \eta^T \nu - \frac{1}{2} q^T Q^{-1} q \\ &\text{subject to} && \mu \geq 0, \end{aligned}$ |
|----------|--|

where  $p = \dim(b) + \sum_\omega \dim(b_\omega)$ . The optimal solution of (StQPdual) is  $\lambda = (\lambda^{x^*}, \lambda^{1^*}, \dots, \lambda^{|\Omega|^*})$  and  $\mu = (\mu^{x^*}, \mu^{1^*}, \dots, \mu^{|\Omega|^*})$ . As discussed above the L-shaped method provides us with values of  $\lambda^{\omega^*}, \mu^{\omega^*}$ ,  $\omega \in \Omega$ . The

required multipliers  $(\lambda^{x^*}, \mu^{x^*})$  can be obtained as solutions of a smaller QP, (MULT), formed from (StQPdual) by plugging in the optimal values for variables  $\lambda^\omega, \mu^\omega$ , for all  $\omega \in \Omega$ . Thus multipliers  $\lambda^{x^*}, \mu^{x^*}$  are recovered as solutions of the following problem.

|      |  |
|------|--|
| MULT | $\begin{aligned} & \underset{\lambda^x, \mu^x}{\text{maximize}} && \begin{pmatrix} \lambda^x \\ \mu^x \end{pmatrix}^T \tilde{N} \begin{pmatrix} \lambda^x \\ \mu^x \end{pmatrix} + \tilde{\eta}^T \begin{pmatrix} \lambda^x \\ \mu^x \end{pmatrix} \\ & \text{subject to} && \mu^x \geq 0 \end{aligned}$ |
|------|--|

where

$$\tilde{N} = \begin{pmatrix} AP^{-1}A^T & P^{-1}A^T \\ P^{-1}A^T & P^{-1} \end{pmatrix} \text{ and } \tilde{\eta} = \begin{pmatrix} -AP^{-1} \sum A_\omega^T \lambda^{\omega*} + b + AP^{-1}c - P^{-1} \sum A_\omega \mu^{\omega*} \\ P^{-1}c \end{pmatrix}.$$

It is noteworthy that this procedure of recovering  $\lambda^{x^*}$  and  $\mu^{x^*}$  involves solving only one more QP for every SQP iteration and hence does not significantly impact the overall performance of the algorithm.

## 5 Numerical Results

This section discusses the numerical performance of our algorithm from a variety of standpoints. We begin by examining the behavior of the algorithm with both subproblem solvers on a set of stochastic nonlinear programming test problems. Next, we discuss the impact of changing key parameters associated with the workings of the subproblem solvers. Finally, this is followed by a comparative study between the performances of the inexact cut and the trust-region variants of our proposed algorithm.

### 5.1 Scalability of the algorithm

There is no standard stochastic NLP test problem set available. Chen and Womersley [14] have developed test problems for stochastic quadratic programs. We use this set to generate two sets of test problems (convex and general nonlinear) as follows.

- *Convex problems:* The objective function was taken to be the norm function (of 2<sup>nd</sup> and higher order), parametrized by the scenario number  $\omega$ . The constraints were taken from the Chen and Womersley set for the corresponding number of realizations. This way we ensure that we have the required recourse properties on the subproblems. Specifically, the objective function is given by

$$f(x) = \frac{1}{2}x^T Hx + c^T x + d + (1 + \sum (x(i) - a_i(\omega)))^{1/p}$$

where  $H \succ 0, p = 2, 4, \dots$  and  $a_i(\omega)$  is a function of  $\omega$ .

- *Nonlinear problems:* In this case we used the first stage objective as a function from the Hock–Schittkowski set [33] and the second stage objective as a convex function. The constraints were again taken from the Chen and Womersley set. Following are the two problems we used along with their index in the Hock–Schittkowski set.

$$f(x) = (x(1) - x(2))^2 + (x(3) - 1)^2 + (x(4) - 1)^4 + (x(5) - 1)^6 \tag{Prob.35}$$

$$f(x) = (x(1) - 1)^2 + (x(1) - x(2))^2 + (x(2) - x(3))^2 + (x(3) - x(4))^4 + (x(4) - x(5))^4 \tag{Prob.58}.$$

In the results that follow, “ILS solver” means Algorithm 2 and “TR solver” means Algorithm 3. We first present some representative problems to show the effect of changing  $|\Omega|$  and the comparative behavior of the two algorithms. `cpu` denotes the CPU time in seconds and `maj` denotes the number of outer SQP iterations, `cuts` is the number of optimality cuts added and `kkt` is the KKT residual in  $\ell_\infty$  norm. For every case, the first stage variable  $x$  and the second stage variable  $y^\omega$  had dimension 5. The constraints were linear, with 4 rows for the first stage constraint and 5 rows for the second stage constraints. Thus the number of variables =  $5 \times |\Omega| + 5$ . The number of constraints =  $5 \times |\Omega| + 4$ , in addition to  $5 \times |\Omega| + 5$  nonnegativity constraints. Unless otherwise stated in the tables, all algorithmic parameters (like inexactness update,  $\Delta_{hi}$  etc.) were held constant across problems.

The convex problems were solved using exact Hessians, while the nonlinear problems used our update rule. In most cases, the KKT residual desired to satisfy (11) is about  $10^{-5}$ . The reader may observe that for several instances the KKT residual obtained has been greater than this number. This is because the algorithm has been terminated because of minimal progress. A detailed assessment of this phenomenon will follow later.

Table 1: Computational effort for the ILS solver – with convex costs ( $p = 2, a_i(\omega) = \omega/|\Omega| \ \forall i, u = 2$ )

| Prob | $ \Omega $ | cpu <sup>ils</sup> | maj <sup>ils</sup> | cuts <sup>ils</sup> | qps <sup>ils</sup> | kkt <sup>ils</sup> |
|------|------------|--------------------|--------------------|---------------------|--------------------|--------------------|
| 1    | 729        | 2432               | 6                  | 348                 | 254046             | 2.2E-04            |
| 2    | 1000       | 1446               | 5                  | 150                 | 150155             | 6.0E-05            |
| 3    | 1225       | 3793               | 4                  | 321                 | 393229             | 1.0E-06            |
| 4    | 1600       | 1293               | 3                  | 83                  | 132886             | 2.7E-05            |
| 5    | 2187       | 3643               | 5                  | 174                 | 380717             | 1.9E-04            |
| 6    | 6561       | 14967              | 5                  | 217                 | 1423959            | 3.8E-04            |
| 7    | 16384      | 74734              | 5                  | 430                 | 7045555            | 9.6E-05            |
| 8    | 32768      | 63300              | 4                  | 189                 | 6193345            | 1.1E-04            |
| 9    | 59049      | 218836             | 4                  | 354                 | 20903704           | 2.1E-05            |
| 10   | 65536      | 64599              | 3                  | 87                  | 5701722            | 3.0E-06            |

Table 2: Computational effort for TR solver – with convex costs ( $p = 2, a_i(\omega) = \omega/|\Omega| \ \forall i, \Delta_{hi} = 10$ )

| Prob | $ \Omega $ | cpu <sup>tr</sup> | maj <sup>tr</sup> | cuts <sup>tr</sup> | qps <sup>tr</sup> | kkt <sup>tr</sup> |
|------|------------|-------------------|-------------------|--------------------|-------------------|-------------------|
| 1    | 729        | 2676              | 6                 | 377                | 275216            | 1.0E-08           |
| 2    | 1000       | 1697              | 8                 | 172                | 172180            | 1.5E-04           |
| 3    | 1225       | 13868             | 12                | 1165               | 1428302           | 2.4E-05           |
| 4    | 1600       | 1026              | 3                 | 65                 | 104068            | 2.4E-04           |
| 5    | 2187       | 4368              | 4                 | 208                | 455108            | 1.6E-04           |
| 6    | 6561       | 11141             | 6                 | 174                | 1141794           | 1.0E-06           |
| 7    | 16384      | 35145             | 3                 | 223                | 3653858           | 7.4E-05           |
| 8    | 32768      | 27764             | 3                 | 81                 | 2654292           | 1.1E-04           |
| 9    | 59049      | 285652            | 6                 | 482                | 28462106          | 4.8E-05           |
| 10   | 65536      | 51568             | 3                 | 70                 | 4587593           | 1.3E-04           |

Table 3: Computational effort for the ILS solver – nonlinear costs (Prob. 35,  $u = 2$ )

| Prob | $ \Omega $ | cpu <sup>ils</sup> | maj <sup>ils</sup> | cuts <sup>ils</sup> | qps <sup>ils</sup> | kkt <sup>ils</sup> |
|------|------------|--------------------|--------------------|---------------------|--------------------|--------------------|
| 1    | 729        | 7336               | 23                 | 1010                | 737323             | 9.4E-03            |
| 2    | 1000       | 3842               | 24                 | 385                 | 385409             | 2.4E-04            |
| 3    | 1225       | 5266               | 13                 | 430                 | 527193             | 4.6E-05            |
| 4    | 1600       | 5435               | 29                 | 337                 | 539566             | 9.2E-04            |
| 5    | 2187       | 10844              | 26                 | 501                 | 1096214            | 5.2E-04            |

Table 4: Computational effort for the TR solver – nonlinear costs (Prob. 35,  $\Delta_{hi} = 10$ )

| Prob | $ \Omega $ | cpu <sup>tr</sup> | maj <sup>tr</sup> | cuts <sup>tr</sup> | qps <sup>tr</sup> | kkt <sup>tr</sup> |
|------|------------|-------------------|-------------------|--------------------|-------------------|-------------------|
| 1    | 729        | 11254             | 31                | 1589               | 1159970           | 5.0E-06           |
| 2    | 1000       | 4706              | 27                | 474                | 474474            | 3.0E-03           |
| 3    | 1225       | 5549              | 23                | 460                | 563960            | 1.5E-05           |
| 4    | 1600       | 12607             | 50                | 791                | 1266391           | 1.3E-04           |
| 5    | 2187       | 16843             | 39                | 787                | 1721956           | 1.3E-04           |

It is notable that both TR and ILS algorithms show good scaling with  $|\Omega|$ . We have noted some problems to show have taken far less effort than expected which we can regard as being exceptions from the broad linearity seen otherwise. From our tables it can be seen that for convex problem

$$\max \left\{ \frac{\text{qp}_1/\text{qp}_2}{|\Omega_1|/|\Omega_2|} \right\} \approx \begin{cases} 2, & \text{ILS} \\ 3, & \text{TR.} \end{cases}$$

Expectedly the performance on nonlinear problems is worse, though still linear. We also observed that for convex problems, the both solvers showed little change in the number of major SQP iterations. For nonlinear problems, we found no such trend. An important insight obtained from the results was that TR solver provided better KKT points (lower residuals) than the ILS method, but the ILS method converged faster.

## 5.2 Modifying the TR update

An interesting trend emerges for the trust-region method when the maximum trust region  $\Delta_{hi}$  is varied. Recall that at any iteration we must have  $\Delta_{k,l} \leq \Delta_{hi}$ . Let us denote  $\Delta_\infty = \max_{k,l} \{\Delta_{k,l}\}$  as the largest trust region used by the algorithm during an implementation with  $\Delta_{hi} = \infty$ . We carried out tests for a fixed convex problem by increasing  $\Delta_{hi}$  from 5 to 500. The results are tabulated in table 5.

It was seen that an increase in  $\Delta_{hi}$  lead to greater number of cuts and longer CPU times. This behaviour is expected. In the implementation of the trust region method, we begin with a trust region of size  $10^{-3}$ . Such a small trust region often leaves the initial point infeasible. In such a situation we increase the trust region gradually until a feasible point is obtained. This trust region is  $\Delta_{0,0}$ . After the first iterate the trust region is increased only when a new major iterate is found and the model update rule for this case is satisfied. If  $\Delta_{hi}$  is large, the model update rule permits a large  $\Delta_{k,l}$  leaving the algorithm with a large region to be approximated by cuts, leading to more cuts. After a certain value an increase in  $\Delta_{hi}$  does not make any difference to the performance since we get  $\max_{k,l} \{\Delta_{k,l}\} = \Delta_\infty \ll \Delta_{hi}$ .

Table 5: Variation of computational effort and accuracy with  $\Delta_{hi}$  for TR method

| Prob | $ \Omega $ | $\Delta_{hi}$ | cpu  | maj | cuts | qps    | $\Delta_\infty$ | kkt     |
|------|------------|---------------|------|-----|------|--------|-----------------|---------|
| 1    | 729        | 5             | 3157 | 6   | 240  | 175200 | 5               | 1.0E-08 |
| 2    | 729        | 10            | 2676 | 6   | 377  | 275216 | 10              | 1.0E-08 |
| 3    | 729        | 15            | 5947 | 8   | 456  | 332880 | 15              | 2.0E-06 |
| 4    | 729        | 20            | 9056 | 10  | 685  | 500050 | 16.38           | 1.0E-06 |
| 5    | 730        | 30            | 9013 | 10  | 685  | 500735 | 16.38           | 1.0E-06 |
| 6    | 729        | 50            | 8971 | 10  | 685  | 500050 | 16.38           | 1.0E-06 |
| 7    | 729        | 500           | 9097 | 10  | 685  | 500050 | 16.38           | 1.0E-06 |

## 5.3 Comparing ILS and TR

In this section, we present two types of performance profiles [21] which are plotted in Fig. 1:

1. with the performance metric as the number of QPs solved, shown in Fig. 1(a)
2. with the performance metric as the KKT residual (note that the  $x$ -axis is plotted in logarithmic scale), shown in Fig. 1(b).

Our termination criteria for the algorithm were very strict, with  $\tau_P = \tau_D = 10^{-6}$  in criteria (11), while the termination criteria for the inner stochastic QPs were identical for both methods:  $\tau = 10^{-10}$  for the ILS method and  $\epsilon_{tol} = 10^{-10}$  for TR.

It is clear that while the two methods are comparable purely on the basis of the number of QPs solved, the TR algorithm finds KKT points with lower residuals. Some explanations can be offered for this observed behaviour. The ‘‘acceptance test’’ in the TR method (18) is essentially a sufficient descent criterion. Theorem 4.12 shows that the convergence of

$$\widehat{Q}(x_k) - m_{k,l}(x_{k,l})$$

is monotone. The ILS method on the other hand offers no monotonicity guarantee. This is the crucial difference between the methods. While iterates of the ILS method might *hover around* the optimal point, the TR method proceeds steadily towards it, often leading to a more accurate KKT point of (SNLP). In the practical implementation of the algorithm we found that the Benders-SQP method may terminate due to the step length  $\alpha$  obtained from the linesearch becoming arbitrarily small. As a consequence, the algorithm cannot make any further progress. Our understanding is that the latter becomes a possibility when the stochastic QPs are not solved to desired accuracy. As a result the multipliers obtained by solving problem (MULT) are not compatible with the primal solution, and what results is *not* a KKT point of the stochastic QP. This error propagates and eventually the algorithm ends up at a point from where it can neither make any progress, nor satisfy criterion (11). In such a situation we declare the algorithm to have terminated. Indeed, we have observed that tightening the termination criteria for the stochastic QPs results in better (lower) KKT residuals for (SNLP).

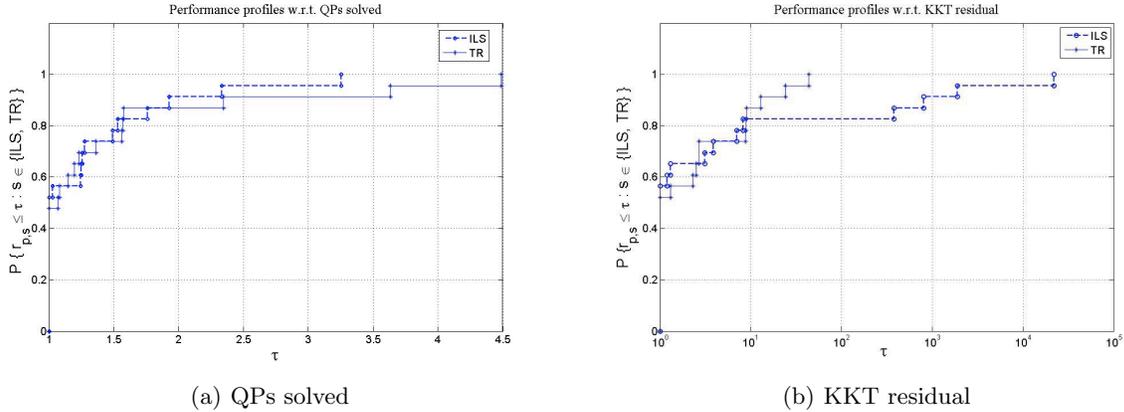


Figure 1: Performance profiles

## 5.4 Comparison with alternate schemes

An important question that remains is how our framework compares to other methods. Such comparisons generally require a set of standard testproblem sets, such as those for nonlinear programming [33], mathematical programming with equilibrium constraints (MPECs) [39] and others. Further, one requires access to the relevant solvers. For instance in the context of solving NLPs and MPECs, this necessitates access to solvers such as SNOPT [25], KNITRO [13] and others. However, stochastic nonconvex nonlinear programming is at a nascent stage where neither off-the-shelf solvers nor testproblem sets exist.

Suppose [14] is used as a starting point; this reference presents a problem generator for two-stage stochastic quadratic programs and provides a foundation for generating stochastic nonlinear convex and nonconvex programs (as we have done in section 5.1). In [14], the authors compare several schemes for solving the stochastic QPs, two of them being projection schemes and splitting methods. Unfortunately, all of these methods can generally be employed for solving convex quadratic, or at best, general convex programs. The nonconvexity cannot be addressed systematically and therefore, a direct comparison is rather challenging to make. It should be said that these schemes could be employed for solving QP subproblems instead of our Benders approach. Note that our interest was in a more general set of stochastic programs (NLPs) and such a comparison would be more relevant in a paper dedicated to scalable schemes for stochastic QPs.

Yet, if we did indeed follow such an avenue, we expect that the schemes in [14] will bear a strong similarity in terms of the scalability of effort (linear) with  $|\Omega|$ . But, the applicability of these schemes is not always straightforward and may not always prove advantageous as we explain further in the next two points:

1. For instance, splitting methods, as suggested in [14] are applied to a primal-dual form that arises from a pure (as opposed to mixed) complementarity formulation of the QP. Convergence of splitting methods [15], particularly in asymmetric settings, is not immediate even when considering the complementarity formulation of the underlying strictly convex QP. Instead, one needs a two-level scheme that requires a regularization. Convergence of the overall scheme can then be obtained by Tikhonov schemes. In addition, a splitting method requires solving a set of scenario-specific monotone linear complementarity problems (LCPs), in contrast with solving sparse stochastic convex QPs (as is the case with our dual decomposition scheme).
2. In [14], the authors discuss a parallel projection method for stochastic QPs. Projection methods [23], as is well known, often require strong convexity (or a strong monotonicity property in the context of variational inequalities) and Lipschitzian properties to ensure a contraction. The resulting steplength in constant steplength schemes are restricted by a function of the monotonicity and Lipschitz constants. Consider for instance if a mapping is close to monotone ( i.e. monotonicity constant is close to zero), then the resulting steplengths are small, implying that the convergence properties are poor. Extensions of such schemes to parallel settings generally require solving the projection problem efficiently. For instance, if the set being projected upon is  $\mathbb{R}_+^n$ , then the projection is easily computed; otherwise, it needs the solution of a quadratic program. Note that in our setting, we do impose strict (not strong) convexity (see our related work in [57])

that relaxes this assumption) but the crucial difference is that the convergence properties do not hinge on either the monotonicity or Lipschitz constant. Furthermore, when using active-set or interior-point schemes for solving QPs, the local convergence behavior in the vicinity of the solution of the QP is superlinear or quadratic.

Yet, in many settings, these schemes are worth considering and their examination and implementation remain an integral part of our future research. We see our current work as providing one of the first data points for conducting such a comparison and plan to make our code/testproblem sets available for future comparisons.

## 6 Summary and future directions

This paper has made two contributions to stochastic nonlinear programming. The first is a characterization of the set of almost-sure feasible solutions of (SNLP) for general measure spaces and convex objective functions with bounded level sets. We showed that if the “W-condition” holds,  $x$  is almost sure feasible for (SNLP) if and only if (RNLP( $x; \omega$ )) is feasible for all  $\omega \in \tilde{\mathcal{Y}}$ . We derived such W-conditions for problems with linear and nonlinear constraints. Using such conditions, sufficient conditions for complete and relatively-complete recourse were presented in this setting. We were also able to derive interesting insights in the implementation of L-shaped methods where we showed that if the W-condition holds feasibility cuts in L-shaped method need be added only for scenarios  $\omega \in \tilde{\mathcal{Y}}$ .

The second is a novel technique of solving stochastic nonlinear programs. The suggested method displays *global convergence* and *superlinear local convergence properties*, while numerical results have shown that it possesses *scalability* and is consequently capable of solving problems of a few hundred thousand variables. Our scheme, while inspired by SQP methods, has three crucial differences: (1) a sparse quasi-Newton update that takes advantage of the structure of the problem, (2) the use of two variants of cutting-plane methods for effectively solving the stochastic QP approximations, and finally (3) a means of obtaining multiplier estimates by solving a single sparse QP.

We consider this work the first step in the development of algorithms for solving large-scale multiperiod nonlinear stochastic optimization problems. Many questions however still persist. For instance, do immediate extensions of the proposed framework to the multiperiod domain exist? Our view is that Benders framework has provided much by way of solving multistage LPs [8, 7]. Through QP extensions to such approaches, we believe that large-scale multistage stochastic NLPs could be solvable by Benders-SQP techniques.

## References

- [1] K. T. AU, J. L. HIGLE, AND S. SEN, *Inexact subgradient methods with applications in stochastic programming.*, Math. Program., 63 (1994), pp. 65–82.
- [2] J-P. AUBIN AND H. FRANKOWSKA, *Set-valued Analysis*, Springer, 1990.
- [3] E. M. L. BEALE, *On minimizing a convex function subject to linear inequalities*, Journal of the Royal Statistical Society, 17B (1955), pp. 173–184.
- [4] J. F. BENDERS, *Partitioning procedures for solving mixed-variables programming problems*, Numer. Math., 4 (1962), pp. 238–252.
- [5] A. BERKELAAR, C. DERT, B. OLDENKAMP, AND S. ZHANG, *A primal-dual decomposition-based interior point approach to two-stage stochastic linear programming*, Oper. Res., 50 (2002), pp. 904–915.
- [6] D. BERTSEKAS, *Nonlinear Programming: 2nd Edition*, Athena Scientific, Belmont, MA., 1999.
- [7] J.R. BIRGE, *Decomposition and partitioning methods for multi-stage stochastic linear programs*, Operations Research, 33 (1985), pp. 989–1007.
- [8] J. R. BIRGE AND F. LOUVEAUX, *Introduction to Stochastic Programming: Springer Series in Operations Research*, Springer, 1997.

- [9] J. BLOMVALL, *A multistage stochastic programming algorithm suitable for parallel computing*, Parallel Comput., 29 (2003), pp. 431–445. Parallel computing in numerical optimization (Naples, 2001).
- [10] J. BLOMVALL AND P. O. LINDBERG, *A Riccati-based primal interior point solver for multistage stochastic programming*, EJOR, 143 (2002), pp. 452–461.
- [11] P. BOGGS AND J. TOLLE, *Sequential quadratic programming*, Acta Numerica, 4 (1995), pp. 1–50.
- [12] P. BOGGS, J. TOLLE, AND P. WANG, *On the local convergence of Quasi-Newton methods for constrained optimization*, SIAM Journal on Control and Optimization, 20 (1982), pp. 161–171.
- [13] R.H. BYRD, M.E. HRIBAR, AND J. NOCEDAL, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optim., 9 (1999), pp. 877–900 (electronic). Dedicated to John E. Dennis, Jr., on his 60th birthday.
- [14] X. CHEN AND R. S. WOMERSLEY, *Random test problems and parallel methods for quadratic programs and quadratic stochastic programs*, Optim. Methods Softw., 13 (2000), pp. 275–306.
- [15] R. W. COTTLE, J-S. PANG, AND R. E. STONE, *The Linear Complementarity Problem*, Academic Press, Inc., Boston, MA, 1992.
- [16] G. B. DANTZIG, *Linear programming under uncertainty*, Management Sci., 1 (1955), pp. 197–206.
- [17] G. B. DANTZIG AND P.W. GLYNN, *Parallel processors for planning under uncertainty*, Ann. Oper. Res., 22 (1989), pp. 1–21.
- [18] G. B. DANTZIG AND G. INFANGER, *Large scale stochastic linear programs: Importance sampling and benders decomposition*, tech. report, 1991.
- [19] G. DENG AND M.C. FERRIS, *Variable-number sample-path optimization*, Math. Program., 117 (2009), pp. 81–109.
- [20] J. E. DENNIS JR. AND J. J. MORÉ, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.
- [21] E. D. DOLAN AND J. J. MORÉ, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.
- [22] W. S. DORN, *Duality in quadratic programming*, Quart. Appl. Math., 18 (1960/1961), pp. 155–162.
- [23] F. FACCHINEI AND J-S. PANG, *Finite Dimensional Variational Inequalities and Complementarity Problems: Vols I and II*, Springer-Verlag, NY, Inc., 2003.
- [24] R. FLETCHER AND S. LEYFFER, *User manual for filterSQP*, May 21 1998.
- [25] P. E. GILL, W. MURRAY, AND M. A. SAUNDERS, *SNOPT: an SQP algorithm for large-scale constrained optimization*, SIAM Rev., 47 (2005), pp. 99–131 (electronic).
- [26] P. E. GILL, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT, *User's guide for NPSOL (version 4.0): A Fortran package for nonlinear programming*, Technical Report SOL 86-2, Department of Operations Research, Stanford University, Stanford, CA, USA, jan 1986.
- [27] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization*, Academic Press, Boston, MA, USA, 1981.
- [28] M. J. GOLDSMITH, *Sequential Quadratic Programming Methods based on Indefinite Hessian Approximations*, PhD thesis, Stanford University, 1999.
- [29] S. P. HAN, *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Programming, 11 (1976), pp. 263–282.

- [30] ———, *A globally convergent method for nonlinear programming*, J. Optim. Theory Applic., 22 (1977), pp. 297–309.
- [31] J. HIGLE AND S. SEN, *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*, Kluwer Academic Publishers, Boston, MA., 1996.
- [32] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Convex Analysis and Minimization Algorithms*, vol. 1 and 2, Springer, Berlin, 1993.
- [33] W. HOCK AND K. SCHITTKOWSKI, *Test Examples for Nonlinear Programming Codes*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1981.
- [34] G. HOEK, *Asymptotic properties of reduction methods applying linearly equality constrained reduced problems*, in Algorithms for Constrained Minimization of Smooth Nonlinear Functions, 1982, pp. 162–189.
- [35] T. HOMEM-DE-MELLO, *Variable-sample methods for stochastic optimization*, ACM Trans. Model. Comput. Simul., 13 (2003), pp. 108–133.
- [36] G. INFANGER, *Monte Carlo (importance) sampling within a Benders decomposition algorithms for stochastic linear programs*, Ann. Oper. Res., 39 (1992), pp. 41–67.
- [37] ———, *Planning Under Uncertainty*, Boyd and Fraser Publishing Co., 1994.
- [38] K. C. KIWIEL, *Proximity control in bundle methods for convex nondifferentiable minimization*, Mathematical Programming, 46 (1990), pp. 105–122.
- [39] S. LEYFFER, *macMPEC: AMPL collection of MPECs*, tech. report, University of Dundee, 2000.
- [40] G-H. LIN, X. CHEN, AND M. FUKUSHIMA, *Smoothing implicit programming approaches for stochastic mathematical programs with linear complementarity constraints*, Technical Report, 2003-006, Department of Applied Mathematics and Physics, Kyoto University, (2003).
- [41] J. LINDEROTH, A. SHAPIRO, AND S. WRIGHT, *The empirical behavior of sampling methods for stochastic programming*, Ann. Oper. Res., 142 (2006), pp. 215–241.
- [42] J. LINDEROTH AND S. WRIGHT, *Decomposition algorithms for stochastic programming on a computational grid*, Comput. Optim. Appl., 24 (2003), pp. 207–250. Stochastic programming.
- [43] X. LIU AND G. ZHAO, *A decomposition method based on SQP for a class of multistage stochastic nonlinear programs*, SIAM J. on Optimization, 14 (2003), pp. 200–222.
- [44] A. LUCIA, *An explicit quasi-newton update for sparse optimization calculations*, Mathematics of Computation, 40 (1983), pp. 317–322.
- [45] W. MURRAY AND F. J. PRIETO, *A sequential quadratic programming algorithm using an incomplete solution of the subproblem*, SIAM J. Optim., 5 (1995), pp. 590–640.
- [46] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.
- [47] M. J. D. POWELL, *Algorithms for nonlinear constraints that use Lagrangian functions*, Math. Prog., 14 (1978).
- [48] ———, *The convergence of variable metric methods for nonlinearly constrained optimization problems*, in Nonlinear Programming 3, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, NY, 1978, pp. 27–63.
- [49] ———, *A fast algorithm for nonlinearly constrained optimization calculations*, in Lecture notes in Mathematics, Numerical Analysis, Dundee 1977, G. A. Watson, ed., Springer-Verlag, 1978, pp. 144–157.

- [50] S.M. ROBINSON, *Analysis of sample-path optimization*, Mathematics of Operations Research, 21 (1996), pp. 513–528.
- [51] S.M. ROBINSON, *A quadratically-convergent algorithm for general nonlinear programming problems*, Mathematical Programming, 3 (1972), pp. 145–156.
- [52] R.T. ROCKAFELLAR AND R.J.-B WETS, *Scenarios and policy aggregation in optimization under uncertainty*, Math. Operations Res. 16 (1991) 1-29, 16 (1991), pp. 1–29.
- [53] R. T. ROCKAFELLAR AND R. J.-B. WETS, *Stochastic convex programming: Kuhn-Tucker conditions*, J. Math. Econom., 2 (1975), pp. 349–370.
- [54] A. RUSZCZYNSKI, *Decomposition methods*, in Handbook in Operations Research and Management Science, vol. 10, Elsevier Science, Amsterdam, 2003, pp. 141–212.
- [55] U. V. SHANBHAG, *Decomposition and Sampling Methods for Stochastic Equilibrium Problems*, PhD thesis, Department of Management Science and Engineering (Operations Research), Stanford University, 2006.
- [56] U. V. SHANBHAG, G. INFANGER, AND P. W. GLYNN, *A complementarity framework for forward contracting under uncertainty*, Under second revision at Operations Research, (2007).
- [57] ———, *On the solution of stochastic convex programs via sampling and decomposition*, Under review, (2008).
- [58] D. F. SHANNO, *On variable metric methods for sparse Hessians*, Math. Comp., 34 (1980), pp. 499–514.
- [59] A. SHAPIRO, *Monte carlo sampling methods*, in Handbook in Operations Research and Management Science, vol. 10, Elsevier Science, Amsterdam, 2003, pp. 353–426.
- [60] A. SHAPIRO AND T. HOMEM DE MELLO, *A simulation-based approach to two-stage stochastic programming with recourse*, Mathematical Programming: Series A, 81 (1998), pp. 301–325.
- [61] J. STOER AND R. A. TAPIA, *On the characterization of  $q$ -superlinear convergence of quasi-Newton methods for constrained optimization*, Mathematics of Computation, 49 (1987), pp. 581–584.
- [62] PH. L. TOINT, *On sparse and symmetric matrix updating subject to a linear equation*, Math. of Computation, 31 (1977), pp. 954–961.
- [63] R. M. VAN SLYKE AND R. WETS, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM J. Appl. Math., 17 (1969), pp. 638–663.
- [64] D.W. WALKUP AND R.J.-B. WETS, *Continuity of some convex-cone-valued mappings*, Proc. Amer. Math. Soc., 18 (1967), pp. 229–235.
- [65] D. W. WALKUP AND R. J.-B. WETS, *Stochastic programs with recourse*, SIAM Journal of Applied Mathematics, 15 (1967), pp. 1299–1314.
- [66] R. B. WILSON, *A Simplicial Algorithm for Concave Programming*, PhD thesis, Harvard University, Cambridge, MA., 1963.
- [67] G. ZAKERI, A. B. PHILPOTT, AND D. M. RYAN, *Inexact cuts in Benders decomposition*, SIAM J. Optim., 10 (2000), pp. 643–657.
- [68] G. ZHAO, *A log-barrier method with Benders decomposition for solving two-stage stochastic linear programs*, Mathematical Programming, 90 (2001), pp. 507–536.
- [69] ———, *A Lagrangian dual method with self-concordant barriers for multi-stage stochastic convex programming*, Math. Program., 102 (2005), pp. 1–24.