

# Global optimization of robust chance constrained problems

Panos Parpas\*, Berç Rustem\*, Efstratios N. Pistikopoulos†

March 30, 2007

## Abstract

We propose a stochastic algorithm for the global optimization of chance constrained problems. We assume that the probability measure with which the constraints are evaluated is known only through its moments. The algorithm proceeds in two phases. In the first phase the probability distribution is (coarsely) discretized and solved to global optimality using a stochastic algorithm. We only assume that the stochastic algorithm exhibits a weak\* convergence to a probability measure assigning all its mass to the discretized problem. A diffusion process is derived that has this convergence property. In the second phase, the discretization is improved by solving another nonlinear programming problem. It is shown that the algorithm converges to the solution of the original problem. We discuss the numerical performance of the algorithm and its application to process design.

## 1 Introduction

We discuss the global optimization of a model for decision making under uncertainty. Consider the following problem:

$$\begin{aligned} \min_{x \in X} \quad & \gamma \\ \text{s.t} \quad & P(f(x, \omega) > \gamma) \leq \delta. \\ & P(g(x, \omega) > 0) \leq \delta \end{aligned} \tag{1.1}$$

Where  $f : \mathbb{R}^n \times \Omega \rightarrow \mathbb{R}$ , and  $g : \mathbb{R}^n \times \Omega \rightarrow \mathbb{R}^m$ , are assumed to be smooth but not necessarily convex. The random variables are assumed to live in the space  $(\Omega, \mathcal{B}, P)$ . Where  $\Omega$  is some compact subspace of  $\mathbb{R}^l$ . The problem in (1.1) can be interpreted as follows: one tries to compute the optimal strategy  $x$ , so that the probability of the realized optimal value deviating from the optimal one is small. At the same time, the optimal strategy must satisfy the constraints ( $g$ ) of the system with a certain probability. The parameter  $\delta > 0$  is specified by the decision maker. If  $\delta$  is small, then the decision will be feasible for almost all the possible realizations of the uncertainties. However, for a small  $\delta$ , the optimal

---

\*Department of Computing, Imperial College, London SW7 2AZ

†Centre for Process Systems Engineering, Imperial College, London SW7 2AZ

objective function value will increase. Therefore the parameter  $\delta$  is used by the decision maker to select the level of robustness required.

Problems in the form of (1.1) are known as chance constraints. They have a long history in the realm of decision making under uncertainty. They were first proposed in [1]. Despite the fact that this class of problems has been studied for a long time, their solution poses a significant challenge. One of the reasons is that even if  $f$  and  $g$  are linear, the resulting problem may not be convex (see [2] pp 104 for an example).

One of the key aspects of the modeling phase of any decision making problem, is the description of the probability measure  $P$ . If the solution of a Stochastic Programming problem is to have any value then the chosen uncertainty representation must conform to the beliefs of the decision maker about the problem. Making assumptions about the distribution of the random variables, while theoretically convenient, hinders the usefulness of applying optimization methods in decision making. A review of available methods to generate meaningful descriptions of the uncertainties from data can be found in [3]. In this work we will concentrate on a moment matching approach. That is, we assume that the decision maker can not provide an exact description of the distribution but only knows some of its moments. We therefore assume that  $P \in \mathcal{P}$ , and that  $\mathcal{P}$  is defined using moment information. In concrete terms we propose to discuss an algorithm for the optimization of the following system:

$$\begin{aligned} \min_{x \in X, \gamma} \quad & \gamma \\ \text{s.t.} \quad & P(f(x, \omega) > \gamma) \leq \delta. \quad \forall P \in \mathcal{P} \\ & P(g(x, \omega) > 0) \leq \delta. \quad \forall P \in \mathcal{P}. \end{aligned} \tag{P}$$

Different  $\delta$ 's can be used in the formulation above, for clarity of exposition one  $\delta$  will be used. In this paper we discuss the global optimization of the problem in (P). A similar problem formulation was also proposed in [4]. In [4] uncertainty was assumed to be uniform and with a support given by closed intervals. Moreover, in [4] linear problems were considered. The distinctive feature of this paper is that we propose a stochastic global optimization algorithm that is applicable to problems that are not convex. Moreover we discuss more general distributions as well as allowing the support of the distribution to be (more or less) arbitrary. The approach is based on an appropriately defined Stochastic Differential Equation (SDE) discussed in Section 2.3. The basic idea is to select some realizations of the uncertainties and solve a deterministic version of (P). We then formulate another nonlinear programming problem which we use in order to improve the discretization. The algorithm will be explained in detail in Section 2. The proposed model has been motivated by practical applications in chemical engineering. Using a standard case study analyzed by Floudas [5, 6], Androulakis et. al.[7], and Rönnqvist [8] (among others), we discuss initial numerical experience with the proposed algorithm in Section 4. Convergence of the algorithm will be discussed in an appendix.

## 2 A Global Optimization Algorithm

In this section we describe the proposed algorithm for the solution of (P). A convergence analysis will be carried out in the next section. To start the analysis,

suppose that we have an i.i.d sample  $\Omega_0$  from some probability measure from  $\mathcal{P}$ . At the  $k^{\text{th}}$  ( $k \geq 0$ ) iteration, the following approximate problem is solved:

$$\begin{aligned} F(k) = \min_{x \in X, \gamma} \gamma \\ \text{s.t. } f(x, \omega) \geq \gamma \quad \forall \omega \in \Omega_k \\ g(x, \omega) \geq 0 \quad \forall \omega \in \Omega_k. \end{aligned} \tag{D_k}$$

The problem above is solved using a stochastic algorithm that is described in section 2.3. To facilitate the discussion of the development of the algorithm suppose that a solution of  $(D_k)$  is obtained. For simplicity, ignore the fact that the solution is only obtained in a probabilistic sense. We will discuss these aspects of the algorithm in the appendix, the aim of this section is to give an intuitive idea of how the algorithm works. With the aforementioned warnings in mind, let  $(x_k, \gamma_k)$  denote the solution of  $(D_k)$ . The next step is to check whether the current solution is also feasible in the original problem (P). In order to check for feasibility we need to solve the following problem:

$$\begin{aligned} \Phi(x_k, \gamma_k) = \sup_{\mu \in \mathcal{K}} \int 1(\mathcal{F}(k)) d\mu \\ \mu \in \mathcal{P}, \end{aligned} \tag{2.1}$$

where  $\mathcal{K}$  denotes the set of all finite signed measures that are defined on the  $\sigma$ -field  $\mathcal{F}$  of  $\Omega$ .  $1(A)$  denotes an indicator function on the set A. As was mentioned in the introduction, the set  $\mathcal{P}$  is defined by some known moments, and can be defined as follows: suppose a vector of functions  $m(\omega) = [m_1(\omega) \dots m_n(\omega)]$  and a vector of scalars  $\mu = [\mu_1 \dots \mu_n]$  are given, then  $\mathcal{P}$  is given by:

$$\mathcal{P} = \left\{ \rho \in \mathcal{K} \mid \int_{\Omega} m_i(\omega) d\rho = \mu_i, \quad i = 1 \dots w, \int_{\Omega} d\rho = 1 \right\}.$$

Finally,  $\mathcal{F}(k)$  is defined as follows:

$$\mathcal{F}(k) = \{ \omega \in \Omega \mid f(x_k, \omega) > \gamma, \quad g(x_k, \omega) > 0 \}. \tag{2.2}$$

If the solution of (2.1) yields an optimal solution that is greater than the threshold  $\delta$  specified by the decision maker, then new realizations are needed in order to ensure that the solution of the next discretized problem to be considered, will satisfy the constraints in (P). The problem in (2.1) is a well known problem in probability theory and statistics. Problems of this type are known as optimal Chebyshev inequalities. They have a long history with contributions made from various fields. A textbook reference on early results is [9]. For a more recent survey of this interesting field see [10]. Before discussing the solution of  $(D_k)$ , we will discuss two alternative methods for the solution of (2.1).

## 2.1 SDP approximations for robust chance constraints

Recently useful results have been obtained by Bertsimas et. al. [11], and Lassere [12]. These results are based on the fact that the problem in (2.1) is a conic optimization problem. Using, suitable approximations of the cone involved in (2.1) one can find good approximate solutions. The result that (2.1) can be viewed as a conic optimization problem can be found in [9]. The fact

that these problems can be formulated as tractable cone problems is due to Bertsimas et. al. [11], and Lasserre [12]. However, in the view of the authors the SDP approach is not appropriate for the class of problems considered in this paper. There are two main problems that need to be addressed when solving moment problems with the SDP approach. Firstly, when the problem is even of modest size, the SDP formulation is prohibitively large even for the best available solvers. More importantly, solving the SDP relaxation will only give us a bound, and asymptotically we are guaranteed that the bound will be tight. However, to solve robust chance constrained problems we also need the realizations of the uncertainties. There is, currently, no known method that is guaranteed to extract the realizations of the random variables in the SDP formulation. It is possible, under some rank assumptions, to use the method described in [13] in order to extract this information. However, after some numerical experimentation we found that, for the problems considered in this paper, the software described in [13] could not yield the necessary realizations. For these reasons, we propose to approach the problem of calculating  $\Phi$  using the nonlinear programming problem described in the next sub-section.

## 2.2 Nonlinear programming formulation

It is a well known result (see for example [9]) that any distribution belonging to the set<sup>1</sup>  $\mathcal{P}$ , can be represented as a discrete distribution with  $w + 1$  realizations. Motivated by this result, we can reformulate (2.1) as follows:

$$\begin{aligned}
\Phi(x_k, \gamma_k, \epsilon) = \max_{\rho, y, \nu, z} & \sum_{i=1}^{w+1} \rho_i \\
\text{s.t.} & \sum_{i=1}^{w+1} \rho_i m_j(y_i) + \sum_{i=1}^{w+1} \nu_i m_j(z_i) = \mu_j \\
& \sum_{i=1}^{w+1} (\rho_i + \nu_i) = 1 \\
& f(x_k, y) - \gamma_k \geq \epsilon \\
& g(x_k, y) \geq \epsilon \\
& z, y \in \Omega, \rho, \nu \geq 0.
\end{aligned} \tag{2.3}$$

For the optimal value of (2.1) given by  $\Phi(x_k, \gamma_k)$ , it can be established that  $\lim_{\epsilon \downarrow 0} \Phi(x_k, \gamma_k, \epsilon) = \Phi(x_k, \gamma_k)$ . The value of  $\epsilon$  can be selected to be a small positive constant. This will guarantee that the realization of  $y$  have their support in (2.2).

In this paper we use the formulation in (2.3) in order to numerically compute the solution of (2.1). The formulation above has the obvious disadvantage that it is a non-convex problem, whereas the SDP formulation is convex. However, the formulation in (2.3) can be used to give the numerical values of the realizations of the uncertainties. Since obtaining these realizations is pivotal to the proposed algorithm the formulation in (2.3) is better suited for this class of problems. The only case where the SDP formulation would be appropriate is when the problem involves linear or quadratic functions. In such a case, the SDP formulation can

---

<sup>1</sup>Under the assumption that  $\mu$  is in the interior of the moment space.

be more attractive. However, if the problem only involves linear or quadratic functions then the formulation given here will only involve linear and quadratic functions. Thus even in this sense both formulations are equally powerful. Given that we are discussing the general non-convex case, the formulation in (2.3) is more appropriate. A further advantage of the formulation above is that it is easier to integrate together with the algorithm used to solve  $(D_k)$  (see section 2.3). Finally, the SDP formulation requires the solution of a possibly infinite sequence of convex problems. The formulation in (2.3) requires the solution of a single non-convex problem.

A potential complication with the formulation in (2.3) is that a solution may be found such that:

$$\sum_{i=1}^{w+1} \rho_i > \delta \text{ but } \rho_i < \delta \forall i.$$

In such a case it will be unclear which realization to add to  $\Omega_k$  so that we ensure that problem in  $(D_k)$  is the correct discretization of the original problem. This point is subtle so we will discuss it briefly.

When  $\delta$  is taken to be zero, the the problem in (P) can be written as:

$$\begin{aligned} \min_{x \in X, \gamma} \quad & \gamma \\ \text{s.t} \quad & P(f(x, \omega) > \gamma) = 0. \quad \forall P \in \mathcal{P} \\ & P(g(x, \omega) > 0) = 0. \quad \forall P \in \mathcal{P}. \end{aligned} \quad (2.4)$$

If one wanted to solve the problem above, then (2.3) could be solved and all the  $y$  realizations would be added to  $\Omega_k$ . If this iterative process is stopped when the optimal objective function value of (2.3) is less than  $\delta$ , then we essentially seek to find a  $\delta$  feasible solution to (2.4). Such a solution may, or may not be optimal (it necessarily is feasible) for (P). Consequently, if all the realizations computed in (2.3) are used we may end up with a bound on the optimal solution. Such problems can be avoided if the following assumption is made about the problem under consideration.

A 2.1. If  $(\rho, y, \nu, z)$  is an optimal solution of (2.3), and moreover:

$$\sum_{i=1}^{w+1} \rho_i > \delta.$$

Then there exists an event  $\varrho > \delta$  and an associated realization  $u$ , such that the following system of equations has a feasible solution:

$$\begin{aligned} \varrho m_j(u) + \sum_{i=1}^{w+1} \nu_i m_j(z_i) &= \mu_j \\ \varrho + \sum_{i=1}^{w+1} \nu_i &= 1 \\ f(x_k, u) - \gamma_k &\geq \epsilon \\ g(x_k, u) &\geq \epsilon \\ z, y \in \Omega, \varrho, \nu &\geq 0.. \end{aligned}$$

In our numerical experiments we always found this assumptions to be satisfied. Using the above assumption the problem in (2.3) can be simplified to:

$$\begin{aligned}
& \max \varrho \\
& \text{s.t. } \varrho m_j(u) + \sum_{i=1}^{w+1} \nu_i m_j(z_i) = \mu_j \\
& \varrho + \sum_{i=1}^{w+1} \nu_i = 1 \\
& f(x_k, u) - \gamma_k \geq \epsilon \\
& g(x_k, u) \geq \epsilon \\
& z, y \in \Omega, \varrho, \nu \geq 0..
\end{aligned} \tag{M_k}$$

If the assumption does not hold then we can not add all the computed realizations to  $\Omega_{k+1}$  as this may lead to an over-conservative solution. One option is to add extra constraints to  $(D_k)$ . For example if  $J$  realizations are found, with associated probabilities  $\rho_j$  such that:

$$\begin{aligned}
& f(x_k, u_j) - \gamma_k \geq \epsilon \\
& g(x_k, u_j) \geq \epsilon \\
& \sum_{j=1}^J \rho_j > \delta \quad \max_j \{\rho_j\} < \delta.
\end{aligned}$$

In addition if Assumption (2.1) does not hold then all the  $J$  realizations can be added to  $\Omega_k$  along with the additional constraints:

$$\begin{aligned}
& \text{If } f(x, u_j) - \gamma \geq \epsilon \text{ and } g(x, u_j) - \gamma \geq \epsilon \text{ then } h_j = 1, \text{ otherwise } h_j = 0. \\
& \sum_j h_j \rho_j \leq \delta.
\end{aligned}$$

Since in our numerical experiments assumption 2.1 was always satisfied we will assume it holds true for the rest of the paper.

### 2.3 Diffusions for constrained global optimization

In this sub-section we describe an algorithm, based on a stochastic differential equation (SDE), that can be used to compute the global optima of general non-linear problems. The algorithm, and its properties will be described in detail in a forthcoming publication[14]. The aim of this section is to give the intuitive idea behind the algorithm. In order to describe the algorithm, consider the following generic problem:

$$\begin{aligned}
& P^* = \min_x f(x) \\
& \text{s.t. } g(x) = 0.
\end{aligned} \tag{2.5}$$

Where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  are twice continuously differentiable. In order to solve (2.5), one could solve the following ordinary differential equation (ODE):

$$dX(t) = -\nabla f(X(t))dt - \nabla g(X(t))^T \lambda(X(t))dt,$$

where the Lagrange multiplier vector,  $\lambda$ , is chosen so that it satisfies:

$$\frac{dg}{dt} = -\nabla g(X(t))[\nabla f(X(t)) + \nabla g(X(t))^T \lambda(X(t))] = -\tau g(X(t)) \quad \tau > 0.$$

Such an algorithm was considered in [15]. These algorithms are related to gradient projection algorithms developed in [16, 17]. Gradient projection methods are not very popular in non-linear programming due to the expensive projection step. However, results obtained in [18] and in [19], show that these algorithms can be competitive with the current state of the art. One disadvantage of this class of algorithms is that they converge to local optima. For the linearly constrained case, Parpas et. al.[20] recently proposed to use a stochastic differential equation (SDE) instead of an ODE. For the unconstrained case the use of SDEs was also advocated in [21, 22, 23, 24]. The advantage of the SDE is that it can help the algorithm escape from local minima and eventually reach the global solution. The SDE used is given by:

$$dX(t) = -\nabla f(X(t)) - \nabla g(X(t))^T \lambda(X(t), t)dt + \sqrt{T(t)}dB(t) \quad (2.6)$$

where:

$$\lambda(x, t) \triangleq [\nabla g(x)\nabla g(x)^T]^{-1}[g(x) + T(t) \Delta g(x) - \nabla g(x)\nabla f(x)] \quad (2.7)$$

$B$  represents an  $n$ -dimensional Brownian motion. The latter process is assumed to live on the probability space  $(\mathbb{R}^n, \mathcal{F}(t), \mathcal{G})$ . The basic idea behind the SDE in (2.6) can be explained as follows: The first term encourages the trajectory to reduce the objective function value, but it does not take into account the constraints of the problem. The second term forces the trajectory to remain inside the feasible set. This property can be shown to hold, for a large enough  $t$ [14]. Finally, the last term helps the trajectory to escape from local minima by adding some noise. The function  $T(t)$  is used to control the level of noise. This function is usually referred to as the annealing schedule. In order for the algorithm to theoretically exhibit convergence to the global solution, the annealing schedule is selected as follows [21, 23, 22, 14]:

$$T(t) = \frac{c}{\log(2+t)}, \quad c > 0.$$

Where  $c$  is some large constant. For the unconstrained case the algorithm has been studied in [21, 22, 23, 24]. It has been shown to theoretically converge, in a probabilistic sense, to the global unconstrained global optimum. For the case where the constraints are linear, an algorithm has been proposed in [20]. Numerical experiments from [20] show that the algorithm can be used for large scale problems (about  $n = 10^3$ ), and the algorithm is also robust. For a more detailed discussion of the performance of the algorithm, we refer the reader to [20].

It can be shown that the diffusion in (2.6) eventually converges to a stochastic process with some distribution, say  $\varrho$  [14]. It can also be shown that  $\varrho$  assigns positive mass only to the global minima of the constrained problem in (2.5). The proof is rather long, but the basic idea is quite simple and will briefly be explained below.

The diffusion in (2.6) is a Markov process, and under quite general conditions its generator is given by:

$$\mathcal{A}p(x, t) = \sum_i \left( \frac{\partial f(x)}{\partial x_i} + \sum_j \frac{\partial g_j(x)}{\partial x_i} \lambda_j(x, t) \right) \frac{\partial p(x, t)}{\partial x_i} + T(t) \sum_i \frac{\partial^2 p(x, t)}{\partial x_i^2} \quad (2.8)$$

The transition density of (2.6) is then given by the solution of the following PDE:

$$\frac{\partial p(x, t)}{\partial t} = \mathcal{A}^* p(x, t). \quad (2.9)$$

Where  $\mathcal{A}^*$  denotes the adjoint of  $\mathcal{A}$ , and it is given by:

$$\mathcal{A}^* p(x, t) = \sum_i \frac{\partial}{\partial x_i} \left[ \left( \frac{\partial f(x)}{\partial x_i} + \sum_j \frac{\partial g_j(x)}{\partial x_i} \lambda_j(x, t) \right) p(x, t) \right] + T(t) \sum_i \frac{\partial^2 p(x, t)}{\partial x_i^2}. \quad (2.10)$$

The transition density of the diffusion defines the probability of moving at time  $s$  from point  $x_s$ , to some point  $x_\tau \in B$ , at a later time  $\tau > s$ . It is given by:

$$\Pi(x_s, s, B, \tau) = \int_B p(x_s, s, y, \tau) dy.$$

It can be shown that, under quite general conditions, the transition density function also satisfies (2.10). The equation in (2.10) is known as the Fokker–Planck or Kolmogorov’s forward equation. The proof in [14] (as well as the proof for the linear case in [20]) take advantage of this property of diffusions, by establishing that, asymptotically, the solution of (2.10) converges to some function say  $\theta$ . It is then shown that  $\theta$  assigns mass only to the global minima of (2.5). This result is summarized in the following theorem (for a proof see [14]).

**Theorem 2.2.** *Let  $p$  denote the solution of (2.10), so that:*

$$\Pi(x_0, 0, B, t) = \int_B p(x_0, 0, y, t) dy,$$

where  $B$  is any Borel set in  $\mathbb{R}^n$ . Let  $r$  be any bounded and continuous function, then

$$\lim_{t \rightarrow \infty} \int_B r(x) p(x_0, 0, y, t) dy = \int_B r(x) \theta(y) dy,$$

where  $\theta$  is the density function of a probability measure that assigns all its mass to the global minima of (2.5).

The type of convergence established above is known as weak\* convergence. In the next section, we show how to use the algorithm we just described for the solution of (P) using the approximate problems  $(D_k)$  and  $(M_k)$ .

### 3 Description of the algorithm

We finally have the necessary tools to specify the proposed algorithm. As was mentioned in the previous section, the algorithm proceeds in two phases. In the

first phase the discretized problem in  $(D_k)$  is solved. Then using the obtained solution we solve  $(M_k)$ .

Let  $Z(t) = [X(t), \gamma(t)] \in \mathbb{R}^{n+1}$ , then at the  $k^{\text{th}}$  iteration, the following SDE is considered for the solution of  $(D_k)$ :

$$dZ_k(t) = -\nabla \mathcal{H}(Z_k(t), \Omega_k, t)dt + \sqrt{T(t)}dB_1(t), \quad (3.1)$$

where  $\nabla \mathcal{H}$  denotes the gradient of the Lagrangian of  $(D_k)$ , when the Lagrange multipliers are given by (2.7).

Similarly for the problem in  $(M_k)$ , let  $U(t)$  denote the  $2(w+1)$  decision vector. The relevant SDE is given by:

$$dU_k(t) = -\nabla \mathcal{J}(U_k(t), t, x_k, \gamma_k)dt + \sqrt{T(t)}dB_2(t).$$

We assume that the two Brownian motions are independent.

**Step 0:** Let  $T_M > 0$  and  $(x_0, \gamma_0)$  be given. Set  $k=0$ .

**Step 1:** For  $t = kT_M$  to  $T = (k+1)T_M$  estimate:

$$U_k(T) = U_k(t) - \int_t^T \nabla \mathcal{J}(U_k(\tau), \tau, x_k, \gamma_k)d\tau + \int_t^T \sqrt{T(\tau)}dB_1(\tau).$$

**Step 1.A:** Let  $U(k) = E_P[U_k(T)|\mathcal{F}_T]$ . From  $U_k$  let  $\rho(k)$ , and  $u(k)$  denote the probability and realization vector respectively.

**Step 1.B:** If  $k > 0$  and  $\rho(k) < \delta$  then stop,  $x(k)$  is optimal.

**Step 1.C:** Otherwise,  $\Omega_{k+1} = \Omega_k \cup u(k)$ , and set  $k := k+1$ .

**Step 2:** For  $t = (k-1)T_M$  to  $T = kT_M$  estimate:

$$Z_k(T) = Z_k(t) - \int_t^T \nabla H(Z_k(\tau), \Omega_k, \tau)d\tau + \int_t^T \sqrt{T(\tau)}dB_2(\tau).$$

**Step 2a:** Let  $Z(k) = E_P[Z_k(T)|\mathcal{F}_T]$ . From  $Z(k)$  let  $x(k)$ , and  $\gamma(k)$  denote the part of the vector corresponding the  $x$ , and the  $\gamma$  variables respectively. Go to Step 1.

## 4 A nonlinear pooling problem

In this section we discuss the numerical performance of the algorithm on a classical pooling problem. Due to space limitations we do not discuss how the implementation of the algorithm is done. We only point out that for the algorithm to be implementable then the process defined in (2.6) needs to be discretized to:

$$X(t + \delta t) = X(t) - \nabla f(X(t))\delta t - \nabla g(X(t))^T \lambda(X(t), t)\delta t + \sqrt{T(t)}\delta t u.$$

where  $u \sim \mathcal{N}(0, I)$ . We also note that it is not necessary to exactly compute gradients (and the Laplacian) as defined in the previous sections. This problem was first considered in [25], the formulation given here is closer to the one in

Table 1: Uncertainty Specification

Variable	Support	Mean	Variance
$D_Q$ : Demand for Q	[50,100]	75	5
$R_Q$ : Demand for R	[100,200]	150	20
$C_A$ : Costs for A	[5,10]	7.5	1
$C_B$ : Costs for B	[10,20]	15	4
$C_C$ : Costs for C	[12,25]	18	4

[8, 6]. We note that this is an interesting global optimization problem with many applications and it has been considered in the past by many authors (see e.g. [5, 6]). It is well known that this problem is non-convex (see e.g. [5]). The example given here is a modified version of the one in [8]. An illustration is given in Figure 2. There are three input chemicals A,B and C. These chemicals when blended provide two output products Q and R. We assume that demand for Q and R is uncertain but has mean  $(\mu_Q, \mu_R)$  and variance  $(\sigma_R^2, \sigma_Q^2)$ . No distribution assumptions are made on the demands. We also assume that the cost per unit of chemical (raw materials) is also uncertain. The specification of the uncertainties is given in Table 1. We also assume that each product has a fixed profit per unit ( $C_Q = 100$  and  $C_R = 150$ ). Any product that is not sold is stored. Storage costs are fixed and are denoted by  $S_Q (= 5)$  and  $S_R (= 8)$ . The full problem specification is given by:

$$\begin{aligned} \max \quad & \gamma \\ \text{s.t} \quad & P[C_Q \min\{\sqrt{X_Q}, \sqrt{D_Q}\}^2 + C_Q \min\{\sqrt{X_R}, \sqrt{D_R}\}^2 \\ & - C_A X_A - C_B X_B - C_C X_C \\ & - S_Q \max\{X_Q - D_Q, 0\}^2 - S_R \max\{X_R - D_R, 0\}^2 < \gamma] \leq \delta \quad \forall P \quad (4.1) \\ & X_A + X_B = Y_Q + Y_R \quad (4.2) \\ & Y_Q + X_{CQ} = X_Q \\ & Y_R + X_{CR} = X_R \\ & X_{CQ} + X_{CR} = X_C \\ & ZY_Q + 4X_{CQ} \leq 2.5X_Q \\ & ZY_R + 4X_{CR} \leq 1.5X_R \\ & 3X_A + X_B = Z(X_A + X_B) \\ & 1 \leq Z \leq 3. \quad (4.3) \end{aligned}$$

The variables in the optimization problem above are as follows:

- $X_i$  : Amount (in units) of chemical or product  $i$  used,  $i = A, B, C, R, Q$ .
- $Y_i$  : Flow (in units) from the pooling tank to product  $i$ ,  $i = Q, R$ .
- $X_{C_i}$  : Flow (in units) from chemical  $C$  to product  $i$ ,  $i = Q, R$ .
- $Z$  : Sulphur concentration (in percentage) of the pooling tank.

The objective of the problem is to maximize profits. This objective is expressed in constraint (4.1)-(4.3). The first part of the constraint (4.1) attempts to maximize profits, the second part (4.2) takes into consideration the costs of the raw materials. The third part (4.3) represents the costs of storage. The aim is

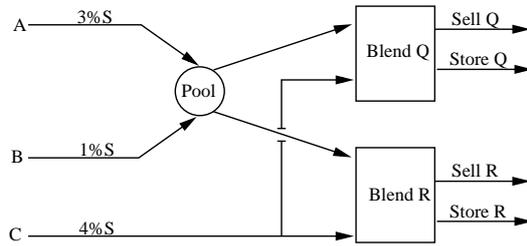


Figure 1: The Haverly Pooling Problem

to calculate the optimal strategy so that the probability of the profit being less than the one calculated is less than  $\delta$  (taken to be 0.05 in our implementation). In Figure (2) we show how the probability of satisfying constraint (4.1)-(4.3) is reduced. In iteration 1, the optimal strategy yields a profit of \$27067.4, but with probability 0.79, the realized profit will be different. The robust strategy yields a profit of \$21708.8, and the probability that profits will be different from this value is 0.048. Results on the optimal strategies are shown in Table 4.

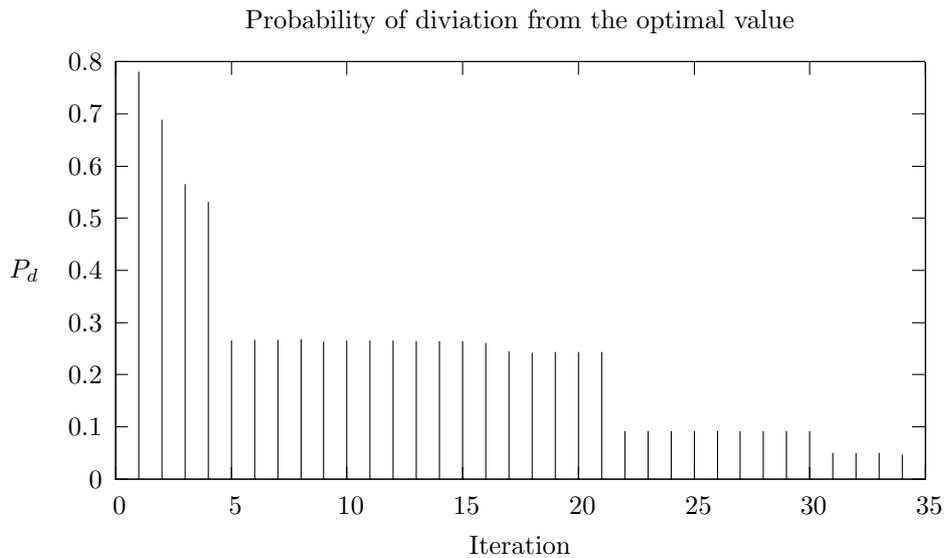


Figure 2:  $P_d$  is the probability of the optimal function value being less than the the calculated, i.e. the probability of constraint (4.1)-(4.3) being violated

## Acknowledgments

The support of EPSRC grant GR/T02560/01 is gratefully acknowledged.

Variable	$k = 1$	Robust Solution
$X_A$	48.74	37.04
$X_B$	146.26	111.2
$X_C$	30	31.81
$X_Q$	75	79.55
$X_R$	150	100.51
$Y_Q$	45	47.75
$Y_R$	150	100.5
$X_{CQ}$	30	31.80
$X_{CR}$	0.002	0.005
$Z$	1.5	1.5
Profit	27067.4	21708.8
Probability	0.21	0.95

Table 2: Comparison of optimal solutions. When  $k = 1$  i.e. in the first iteration of the algorithm the optimal solution obtained is shown in the second column. With this strategy the profit is 27067.4, but with probability 0.79 the profits will change. The robust solution is shown in the last column. The profit has been reduced to 21708.8 but the probability of not achieving this is less than 0.05.

## Appendix

The basic idea behind the convergence analysis is to show that if  $\Omega_k$  is generated as in Step 1.C of the algorithm, then the problem in  $(D_k)$  will eventually yield a solution that is feasible in (P). Of course, any such feasible solution must also be optimal (note that we are making use of assumption 2.1). The second part of the convergence analysis shows that the diffusion process generated for the solution of (3.1) will converge. From Theorem 2.2 we must then have that (3.1) eventually converges to a stochastic process that assigns its mass to the global minima of (P).

**Theorem 4.1.** *Suppose that assumption (2.1) holds, and that  $\Omega_k$  is generated as in Step 1.C of the algorithm. Moreover assume that the original problem is feasible, and that  $(M_k)$  has a solution, for any  $(x, \gamma)$ . Finally, assume that  $T_M$  is large enough so that the solution obtained in Step 1 is globally optimal (with probability 1) for  $(M_k)$ . Then there exists a  $k_0$ , such that for any  $k \geq k_0$  the solution of:*

$$\begin{aligned}
 F(k) = \min_{x \in X, \gamma} & \quad \gamma \\
 \text{s.t.} & \quad f(x, \omega) \geq \gamma \quad \forall \omega \in \Omega_k \\
 & \quad g(x, \omega) \geq 0 \quad \forall \omega \in \Omega_k,
 \end{aligned} \tag{4.4}$$

*will eventually yield an optimal solution for (P).*

*Proof.* Let us first note that if the solution of (4.4) is feasible for (P) then it must also be optimal. This is because if  $(x_k, \gamma_k)$  is feasible in the original problem, no new discretizations will be generated when solving  $(M_k)$  (or (2.3)). Since (4.4) always yields a lower bound for (P) it follows that such a solution must also be optimal.

We can now suppose that the algorithm does not stop in Step 1.B. We now establish that eventually the solution of (4.4) will be feasible for (P). Clearly, the set be defined by:

$$\mathcal{P} = \{\rho \in \mathcal{K} \mid \int d\rho = 1, \int_{\Omega} m_i(\omega) = \mu_i, i = 1, \dots, w\} \quad (4.5)$$

is convex. It follows from the Krein-Milman Theorem that  $\mathcal{P}$  can be written as the closure of the convex hull consisting of the extreme points of  $\mathcal{P}$ :

$$\mathcal{P} = \text{cl}[\text{co}(\text{Ext } \mathcal{P})].$$

The global solution of  $(M_k)$  consists of vector  $(\varrho(k), \nu(k)) \in \mathbb{R}^{w+2}$  of probabilities, and an associated vector of realizations:  $(u(k), z(k)) \in \mathbb{R}^{w+2}$ . We claim that the finite collection of points  $(u(k), z(k)) \in \mathbb{R}^{w+2}$  is the support of the extremal measure that maximizes (2.1). Suppose that this is not the case, then there must exist some probability measure with support  $(\hat{u}(k), \hat{z}(k)) \in \mathbb{R}^{w+2}$ , and that satisfies the moment conditions in (4.5). Since we are assuming that  $(u(k), z(k))$  does not form the extremal support maximizing (2.1) we must have that  $\varrho(k) < \hat{\varrho}(k)$  contradicting that  $\varrho(k)$  is the globally optimal solution of  $(M_k)$ . We conclude that the solution of  $(M_k)$  will yield the required measure maximizing (2.1).

The algorithm will therefore generate a sequence of probability measures  $Q^k$ . Such a sequence is tight, since we are assuming that all the measures in  $\mathcal{P}$  have compact support. The sequence will therefore, by Prohorov's Theorem, have a limit point  $Q^*$ , say. As a result, there must exist a  $k_0$ , such that:

$$|Q^*(\mathcal{F}(x, \gamma)) - Q^k(\mathcal{F}(x, \gamma))| < \epsilon \quad \forall k > k_0. \quad (4.6)$$

Let  $(x(k), \gamma(k))$  denote an optimal solution for  $(D_k)$ . Then by using steps 1.A and 1.C of the algorithm we must also have that:

$$Q^{k+1}(\mathcal{F}(x(k), \gamma(k))) < Q^k(\mathcal{F}(x(k), \gamma(k))).$$

Therefore, there must exist an  $r > 0$  such that: Which implies that

$$Q^{k+r}(\mathcal{F}(x(k), \gamma(k))) < \delta.$$

We conclude that there must exist a  $\hat{k}_0$ , such that

$$|Q^*(\mathcal{F}(x(k), \gamma(k))) - \delta| < \epsilon.$$

□

Our next task is to show that the SDE associated with  $(D_k)$  will eventually converge. It is known from Theorem (2.2) that once the transition density function converges, it will converge to density that assigns positive probability only to the global minima of  $(D_k)$ . However, Theorem 2.2 establishes this property as  $t$  grows large. It basically says that if  $T_M$  (see Step 2 of the algorithm) is allowed to be large enough, then  $Z_k(t)$  will converge to  $Z_k$ . The latter process will have a density function that assigns positive probability to the global minima of  $(D_k)$ . Our main result (Theorem 4.2) will show that as  $k$  tends to

infinity  $Z_k$  also converges. From Theorem 4.1 we must then have that once  $Z_k$  converges, then we will have a solution to our original problem.

The SDE under consideration is given by:

$$Z(kT_M) = Z((k-1)T_M) - \int_{(k-1)T_M}^{kT_M} \nabla \mathcal{H}(Z(\tau), \tau, \Omega_k) d\tau + \int_{(k-1)T_M}^{kT_M} \sqrt{T(\tau)} dB_2(\tau). \quad (4.7)$$

Where  $\nabla \mathcal{H}$  is the gradient of the Lagrangian of  $(D_k)$ , when the Lagrange multipliers are given by (2.7). Associated with the SDE in (4.7) is the following initial value problem,

$$\begin{aligned} \frac{\partial p}{\partial t} &= \sum_i \frac{\partial}{\partial x_i} \left( \nabla \mathcal{H}(x, t; \Omega_k)_i p(x, t) \right) + T(t) \frac{1}{2} \frac{\partial^2 p}{\partial x_i^2} \\ p(x, 0) &= h(x). \end{aligned} \quad (4.8)$$

Were  $h$  is some continuously differentiable function. Due to the way the noise is introduced into the system the associated forward equation satisfies a uniform parabolicity condition. Moreover under our assumptions the coefficients of (4.8) are regular. Under these assumptions the solution to the initial value problem is unique, and is given by the integral equation:

$$p(t, x) = \int \Gamma(t, x, y) h(y) dy.$$

The kernel  $\Gamma$  is called the fundamental solution. The function  $p$  that satisfies the preceding equation is called a generalized solution. Using the generalized solution we define a family of operators  $\mathcal{O}_t$  as follows

$$\mathcal{O}_0 h(x) = h(x), \quad \mathcal{O}_t h(x) = \int \Gamma(t, x, y) h(y) dy,$$

where  $t \in [(k-1)T_M, kT_M]$ . Using this operator we establish the desired convergence of the process as  $k$  grows large.

**Theorem 4.2.** *Suppose that the Lagrangian of  $(D_k)$  is three times continuously differentiable and has compact support.  $T_M$  in Steps 1 and 2 of the algorithm is sufficiently large for Theorem 2.2 to hold. Suppose that  $T(t)$  is defined as follows:*

$$T(t) = \frac{c}{\ln(3+t)}.$$

*The there exists a  $\hat{c}$  such that for  $c > \hat{c}$ , and as  $k \rightarrow \infty$  the operator given by  $\mathcal{O}_t h(x)$ , with  $t \in [(k-1)T_M, kT_M]$ , will converge to  $\mathcal{O}^*$ .*

*Proof.* Consider the following SDEs:

$$dZ(s) = -\nabla \mathcal{H}(Z(s), s; \Omega_k) ds + \sqrt{T(s)} dB_2(s) \quad (4.9)$$

$$Z((k-1)T_M) = z \quad s \in [(k-1)T_M, kT_M].$$

$$dZ(s) = \sqrt{T(s)} dB_2(s) \quad (4.10)$$

$$Z((k-1)T_M) = z \quad s \in [(k-1)T_M, kT_M].$$

Let  $R$ , and  $Q$ , be the transition probability measures, associated with (4.9), and (4.10), respectively. Then, by Girsanov's theorem, the Radon-Nikodym derivative of  $Q$  w.r.t  $R$ , is given by:

$$\frac{dQ}{dR} Z(\cdot) = \exp \left\{ \int_{(k-1)T_M}^{kT_M} \left\langle \frac{\nabla \mathcal{H}(Z(s), s; \Omega_k)}{\sqrt{T(s)}}, dB_2(s) \right\rangle - \int_{(k-1)T_M}^{kT_M} \frac{\|\nabla \mathcal{H}(Z(s), s; \Omega_k)\|^2}{2T(s)} ds \right\}. \quad (4.11)$$

From (4.9) we have:

$$dB_2(s) = \frac{dZ(s) + \nabla \mathcal{H}(Z(s), s; \Omega_k) ds}{\sqrt{T(s)}}.$$

Consequently, (4.11) can be written as follows:

$$\begin{aligned} \frac{dQ}{dR} Z(\cdot) = \exp \left\{ \int_{(k-1)T_M}^{kT_M} \left\langle \frac{\nabla \mathcal{H}(Z(s), s; \Omega_k)}{T(s)}, dZ(s) \right\rangle \right. \\ \left. + \int_{(k-1)T_M}^{kT_M} \left\langle \frac{\nabla \mathcal{H}(Z(s), s; \Omega_k)}{2T(s)}, \nabla \mathcal{H}(Z(s), s; \Omega_k) \right\rangle ds \right\}. \end{aligned}$$

Under our assumptions  $Z(s; \Omega_k)$  (with  $s$  in  $[(k-1)T_M, kT_M]$ ) will remain in a compact set. Using this fact the following bound can be derived:

$$\begin{aligned} & \int_{(k-1)T_M}^{kT_M} \left\langle \frac{\nabla \mathcal{H}(Z(s), s; \Omega_k)}{2T(s)}, \nabla \mathcal{H}(Z(s), s; \Omega_k) \right\rangle ds \\ & \leq \frac{c_1}{T(kT_M)}, \end{aligned}$$

where to derive the last inequality we used the monotonicity property of  $T(t)$ .

By Itô's Lemma:

$$d[\mathcal{H}(Z(s), s; \Omega_k)] = \langle \nabla \mathcal{H}(Z(s), s; \Omega_k), dZ(s) \rangle + \frac{\partial \mathcal{H}}{\partial t} ds + \frac{1}{2} \left[ T(s) \sum_i \frac{\partial^2 \mathcal{H}}{\partial x_i^2} \right] ds.$$

Therefore,

$$\left\langle \nabla \mathcal{H}(Z(s), s; \Omega_k), \frac{dZ(s)}{T(s)} \right\rangle = \frac{1}{T(s)} d[\mathcal{H}(Z(s), s; \Omega_k)] - \frac{1}{T(s)} \frac{\partial \mathcal{H}}{\partial t} ds - \frac{1}{2} \left[ \sum_i \frac{\partial^2 \mathcal{H}}{\partial x_i^2} \right] ds \quad (4.12)$$

Integrating by parts, we can calculate the following bound for the first term in the right-hand-side of (4.12):

$$\begin{aligned} & \left| \int_{(k-1)T_M}^{kT_M} \frac{1}{T(s)} d[\mathcal{H}(Z(s), s; \Omega_k)] \right| \\ & = \left| \frac{1}{T(s)} \mathcal{H}(Z(s), s; \Omega_k) \Big|_{(k-1)T_M}^{kT_M} - \int_{(k-1)T_M}^{kT_M} \mathcal{H}(Z(s), s; \Omega_k) d \left[ \frac{1}{T(s)} \right] \right| \leq \frac{c_2}{T(kT_M)}, \end{aligned}$$

where we used the fact that  $Z$  remains in a compact set. Using the same argument, bounds for the rest of the terms in (4.12) can be found:

$$\left| \int_{(k-1)T_M}^{kT_M} \frac{1}{T(s)} \frac{\partial \mathcal{H}}{\partial t} ds \right| \leq \frac{c_3}{T(kT_M)}.$$

$$\left| \int_{(k-1)T_M}^{kT_M} \frac{1}{2} \left( \sum_i \frac{\partial^2 \mathcal{H}}{\partial x_i^2} \right) ds \right| \leq c_4$$

Therefore, there must exist a constant  $c_5$ , such that:

$$\frac{dQ}{dR} Z(\cdot) \leq \exp \left\{ \frac{c_5}{T(kT_M)} \right\}.$$

Under (4.10),  $Z(kT_M)$  is normally distributed. Indeed, its mean and covariance matrix are given by:

$$E[Z(kT_M)] = E[Z((k-1)T_M)] + E \left[ \int_{(k-1)T_M}^{kT_M} \left\langle \sqrt{2T(s)}, dB(s) \right\rangle \right] = z.$$

$$\begin{aligned} & \text{Cov}(Z_i(kT_M)Z_j(kT_M)) \\ &= E[Z_i(kT_M)Z_j(kT_M)] - z_i z_j \\ &= E \left[ \left( z_i + \int_{(k-1)T_M}^{kT_M} \sqrt{T(s)} dB_2^i(s) \right) \left( z_j + \int_{(k-1)T_M}^{kT_M} \sqrt{T(s)} dB_2^j(s) \right) \right] - z_i z_j \\ &= \int_{(k-1)T_M}^{kT_M} T(s) \delta_{ij} ds. \end{aligned}$$

By  $\Lambda$  we denote the covariance matrix, whose  $ij^{\text{th}}$  entry is given by the preceding equation. Let  $\epsilon$  be an arbitrary positive constant.

$$R[|Z(kT_M) - \mathcal{W}| \leq \epsilon] \geq \exp \left\{ -\frac{c_5}{T(kT_M)} \right\} \int_{|u - \mathcal{W}| \leq \epsilon} \frac{\exp \left( -\frac{1}{2} (u - z)^T \Lambda^{-1} (u - z) \right)}{(2\pi)^{\frac{n}{2}} (\det \Lambda)^{\frac{1}{2}}} du.$$

It is easy to see that there exists a  $c_6$ , such that:

$$R[|Z(kT_M) - \mathcal{W}| \leq \epsilon] \geq \exp \left\{ -\frac{c_6}{kT_M} \right\}.$$

Therefore,

$$\begin{aligned} \delta_k &= \inf_{x,y} p((k-1)T_M, x, kT_M, y) \\ &= \inf_{x,y} \lim_{\epsilon \rightarrow 0} \frac{1}{(2\epsilon)^n} R(|Z(kT_M) - \mathcal{W}| \leq \epsilon) \\ &\geq \exp \left\{ -\frac{c_7}{T(kT_M)} \right\}. \end{aligned}$$

Finally, if we choose  $c \geq c_7$  then  $T(t)$  satisfies:

$$\exp \left\{ -\frac{c_7}{T(kT_M)} \right\} \geq \frac{1}{3 + kT_M}.$$

Therefore

$$\lim_{k \rightarrow \infty} [\mathcal{O}_{kT_M} h(x), 0]^- = 0.$$

The result now follows from Theorem 7.4.1 in [26].  $\square$

## References

- [1] A. Charnes and W. W. Cooper, "Chance-constrained programming," *Management Sci.*, vol. 6, pp. 73–79, 1959/1960.
- [2] J. Birge and F. Louveaux, *Introduction to stochastic programming*. Springer Series in Operations Research, New York: Springer-Verlag, 1997.
- [3] J. Dupačová, G. Consigli, and S. Wallace, "Scenarios for multistage stochastic programs," *Ann. Oper. Res.*, vol. 100, pp. 25–53 (2001), 2000.
- [4] E. Erdoğan and G. Iyengar, "Ambiguous chance constrained problems and robust optimization," *Math. Program.*, vol. 107, pp. 37–61, 2006.
- [5] C. Floudas, *Deterministic global optimization*, vol. 37 of *Nonconvex Optimization and its Applications*. Dordrecht: Kluwer Academic Publishers, 2000. Theory, methods and applications.
- [6] C. Floudas, P. Pardalos, C. Adjiman, W. Esposito, Z. Gümüs, S. Harding, J. Klepeis, C. Meyer, and C. Schweiger, *Handbook of test problems in local and global optimization*, vol. 33 of *Nonconvex Optimization and its Applications*. Dordrecht: Kluwer Academic Publishers, 1999.
- [7] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, "αBB: a global optimization method for general constrained nonconvex problems," *J. Global Optim.*, vol. 7, no. 4, pp. 337–363, 1995.
- [8] M. Rönnqvist, "Modelling techniques for nonlinear pooling problems," in *From local to global optimization (Rimfors, 1997)*, vol. 53 of *Nonconvex Optim. Appl.*, pp. 85–103, Dordrecht: Kluwer Acad. Publ., 2001.
- [9] S. Karlin and W. Studden, *Chebyshev systems: With applications in analysis and statistics*. Pure and Applied Mathematics, Vol. XV, Interscience Publishers John Wiley & Sons, New York-London-Sydney, 1966.
- [10] J. E. Smith, "Generalized Chebychev inequalities: theory and applications in decision analysis," *Oper. Res.*, vol. 43, no. 5, pp. 807–825, 1995.
- [11] D. Bertsimas and I. Popescu, "Optimal inequalities in probability theory: a convex optimization approach," *SIAM J. Optim.*, vol. 15, no. 3, 2005.
- [12] J. Lasserre, "Bounds on measures satisfying moment conditions," *Ann. Appl. Probab.*, vol. 12, no. 3, pp. 1114–1137, 2002.

- [13] D. Henrion and J. Lasserre, “Detecting global optimality and extracting solutions in GloptiPoly,” in *Positive polynomials in control*, vol. 312 of *Lecture Notes in Control and Inform. Sci.*, pp. 293–310, Berlin: Springer, 2005.
- [14] P. Parpas, “Constrained global optimization and stochastic differential equations.” In preparation, 2007.
- [15] Y. Evtushenko and V. Zhadan, “Stable barrier-projection and barrier-newton methods in nonlinear programming,” *Optimization Methods and Software*, vol. 3, 1994.
- [16] J. B. Rosen, “The gradient projection method for nonlinear programming. II. Nonlinear constraints,” *J. Soc. Indust. Appl. Math.*, vol. 9, pp. 514–532, 1961.
- [17] D. Luenberger, “The gradient projection method along geodesics,” *Management Sci.*, vol. 18, pp. 620–631, 1972.
- [18] E. G. Birgin and J. M. Martínez, “Local convergence of an inexact-restoration method and numerical experiments,” *J. Optim. Theory Appl.*, vol. 127, no. 2, pp. 229–247, 2005.
- [19] J. Martínez and E. Pilotta, “Inexact restoration methods for nonlinear programming: advances and perspectives,” in *Optimization and control with applications*, vol. 96 of *Appl. Optim.*, pp. 271–291, New York: Springer, 2005.
- [20] P. Parpas, B. Rustem, and E. Pistikopoulos, “Linearly constrained global optimization and stochastic differential equations,” *J. Global Optim.*, vol. 36, no. 2, pp. 191–217, 2006.
- [21] T. Chiang, C. Hwang, and S. Sheu, “Diffusion for global optimization in  $\mathbf{R}^n$ ,” *SIAM J. Control Optim.*, vol. 25, no. 3, pp. 737–753, 1987.
- [22] B. Gidas, “The Langevin equation as a global minimization algorithm,” in *Disordered systems and biological organization (Les Houches, 1985)*, vol. 20 of *NATO Adv. Sci. Inst. Ser. F Comput. Systems Sci.*, pp. 321–326, Berlin: Springer, 1986.
- [23] S. Geman and C. Hwang, “Diffusions for global optimization,” *SIAM J. Control Optim.*, vol. 24, no. 5, pp. 1031–1043, 1986.
- [24] F. Aluffi-Pentini, V. Parisi, and F. Zirilli, “Global optimization and stochastic differential equations,” *J. Optim. Theory Appl.*, vol. 47, no. 1, pp. 1–16, 1985.
- [25] C. A. Haverly, “Studies of the behaviour recursion for the pooling problem,” *SIGMAP Bulletin*, vol. 25, pp. 19–28, 1978.
- [26] A. Lasota and M. Mackey, *Chaos, fractals, and noise*, vol. 97 of *Applied Mathematical Sciences*. New York: Springer-Verlag, second ed., 1994. Stochastic aspects of dynamics.